



Universidad Carlos III Madrid

Ingeniería Técnica Telecomunicaciones:
Sonido e imagen

Proyecto Fin de Carrera

Integración de servicios de voz y Google
Maps

Autor: Manuel Zuil Oliva

Tutor: Israel González Carrasco

Tabla de contenidos

Tabla de contenidos	1
Índice de figuras	7
Indice de tablas	11
1. Introducción	19
2. Definiciones y acrónimos	20
2.1 Definiciones	20
2.2. Acrónimos	20
3. Estudio de Viabilidad del Sistema	22
3.1 Establecimiento del alcance del sistema.....	22
3.1.1 Estudio de la solicitud.....	22
3.1.2 Identificación del alcance del sistema.....	23
3.1.3 Identificación de los interesados en el sistema (stakeholders)	24
3.2 Estudio de la situación actual.....	25
3.3 Definición de los requisitos del sistema	26
3.3.1 Obtención de los requisitos.....	26
3.3.1.1. Requisitos de capacidad.....	28
3.3.1.2 Requisitos de restricción	36
3.3.2. Definición de los requisitos del sistema	39
3.4. Estudio de alternativas de solución	39
3.5 Valoración de las alternativas	40
3.6 Selección de la solución	40
4. Gestión del Proyecto	41
4.1. Organización del proyecto	42
4.2 Planificación	50

4.3 Estimación de costes	57
5. ANÁLISIS DEL SISTEMA	59
5.1. Introducción	59
5.2. Definición del sistema	59
5.2.1 Bases de datos.....	59
5.2.2. Google Maps	60
5.2.3. TTS.....	60
5.2.4. Localización	60
5.2.5. Gestión de archivos.....	60
5.3. Identificación de subsistemas de análisis.....	61
5.3.1. Determinación de los subsistemas de análisis	61
5.4. Establecimiento de requisitos	61
5.4.1. Obtención de requisitos	61
5.4.1.1 Requisitos funcionales.....	63
5.4.1.2 Requisitos de interfaz.....	70
5.4.1.3 Requisitos de operación.....	71
5.4.1.4 Requisitos de documentación	74
5.4.1.5 Requisitos de calidad.....	74
5.4.2. Especificación de los casos de uso	75
5.4.2.1. Actores del sistema	75
5.4.2.2. Listado de casos de uso.....	75
5.5 Análisis de los casos de uso.....	84
5.5.1 Identificación de clases asociadas a un caso de uso	84
5.5.2 Descripción de la interacción de objetos	86
5.6 Análisis de clases	87

5.6.1 Identificación de responsabilidades y atributos	89
5.6.2 Identificación de asociaciones.....	95
5.6.3 Identificación de generalizaciones	100
5.7 Definición de las interfaces de usuario	100
5.7.1. Identificación de perfiles y diálogos.....	101
5.7.2. Especificación de formatos individuales de la interfaz de la pantalla.	101
5.7.2.1. Descripción de pantallas que conforman la aplicación.....	102
5.8 Análisis de consistencia y especificación de requisitos.....	117
5.8.1 Verificación y análisis de consistencia entre modelos	117
5.8.1.1 Matrices de Requisitos de usuario de capacidad-Casos de uso.....	118
5.8.1.2 Matrices de Requisitos software-Casos de uso.....	119
5.8.2 Validación de los modelos.....	119
6. Diseño del Sistema	120
6.1. Alcance	120
6.2 Definición de la arquitectura.....	121
6.2.1 Definición de los niveles de la arquitectura	121
6.2.2. Identificación de requisitos de diseño y construcción.	121
6.2.3 Especificación de estándares y normas de diseño y construcción.....	121
6.2.3.1 Nombres de Ficheros	122
6.2.3.2 Idioma.....	122
6.2.3.3 Cabeceras de las clases	122
6.2.3.4 Visibilidad de las clases	122
6.2.3.5 Identificadores de las clases.....	122
6.2.3.6 Constantes.....	123
6.2.3.7 Variables.....	123

6.2.3.8 Funciones y procedimientos	123
6.2.4 Identificación de los subsistemas de la aplicación	124
6.2.5. Especificación del entorno tecnológico	124
6.2.5.1. Hardware.....	124
6.2.5.2 Software	124
6.2.6. Especificación de requisitos de seguridad y operación.....	124
6.2.7 Estudio de la seguridad requerida en el proceso de diseño del	124
sistema	124
6.2.8 Análisis de los riesgos del entorno tecnológico	125
6.3 Diseño de la arquitectura de soporte	125
6.3.1 Diseño de subsistemas de soporte.....	125
6.3.2 Diseño de casos de uso reales.....	126
6.3.3 Identificación de clases asociadas a un caso de uso	126
6.3.2 Diseño de la realización de los casos de uso	127
6.4 Diseño de clases	128
6.4.1 Identificación de clases de diseño.....	129
6.4.2 Identificación de atributos y métodos de clases de diseño	131
6.5 Almacenamiento de datos	143
6.5.1. Modelado de bases de datos	145
6.5.1.1. Base de datos de usuarios.....	145
6.5.1.2. Base de datos de favoritos	146
6.5.1.3. Base de datos de notificaciones.....	146
6.5.1.4. Relación entre bases	146
6.5.2. Diseño lógico de datos	147
6.5.2.1. Modelo entidad/relación	147

6.5.2.2 Modelo relacional	148
6.5.3. Diseño físico de datos	149
6.5.3.1. Modelo de tablas.....	149
6.5.3.2 Descripción de tablas	150
6.5.3.2.1 Tabla USUARIOS	150
6.5.3.2.2. Tabla FAVORITOS	150
6.5.3.2.2. Tabla NOTIFICACIONES	150
6.6 Verificación y aceptación de la arquitectura del sistema	152
6.6.1 Análisis de consistencia de las especificaciones de diseño.....	152
6.6.1.1 Trazabilidad casos de uso – clases de diseño.....	153
6.7 Especificación Técnica del Plan de Pruebas	154
7. Plan de Validación y Verificación del Software	155
7.1. Definición de pruebas, entornos y metodología.....	155
7.1.1. Pruebas unitarias.....	156
7.1.2. Pruebas de integración y aceptación	156
7.1.3 Matriz de trazabilidad Pruebas Aceptación/Requisitos de	170
Capacidad	170
7.2 Definición de las Pruebas Unitarias.....	171
7.2.1 Definición de las pruebas de implantación	171
7.2.2 Definición de las pruebas de Integración	171
8. Conclusión	173
8.1. Conclusiones del proyecto	173
8.2. Ampliaciones	173
8.2.1. Ampliaciones con necesidad de capital	174
8.2.2. Ampliaciones sin necesidad de capital.....	174

9. Manual de usuario	176
9.1. Acceso a la aplicación.....	176
9.1.1 Acceder a la aplicación	176
9.1.2. Menú inicial.....	177
9.1.3 Base de datos de usuarios.....	179
9.1.4. Creación y edición de usuarios.....	182
9.1.5. Autenticación de usuarios.....	183
9.1.6. Menú principal	184
9.1.7. Favoritos.....	187
9.1.8 Seguimiento de ruta.....	189
9.1.8. Modo Libre	190
9.1.9 Resumen de ruta libre.....	194
10. Bibliografía	196
Anexo A. Diferencia unión punto a punto y algoritmo polilíneas codificadas	197
Anexo B. Cálculo de distancias entre puntos. Navegación por estima inversa	199
Anexo C. Posición del mapa en ruta. Concepto de bearing	200

Índice de figuras

Figura 1. Ciclo de vida.....	42
Figura 2. RBS.....	43
Figura 3. Proceso WBS - Estudio de viabilidad	44
Figura 4. Proceso WBS - Gestión del proyecto.....	44
Figura 5. Proceso WBS – Creación BBDD	45
Figura 6. Proceso WBS – Google Maps	46
Figura 7. Proceso WBS – TTS	47
Figura 8. Proceso WBS – Localización	48
Figura 9. Proceso WBS – Gestión archivos.....	49
Figura 10. Proceso WBS – Manual del usuario.....	50
Figura 11. Diagrama de Gantt (General)	50
Figura 12. Diagrama de Gantt (Estudio de Viabilidad)	51
Figura 13. Diagrama de Gantt (Gestión del Proyecto)	52
Figura 14. Diagrama de Gantt (Creación de BBDD).....	53
Figura 15. Diagrama de Gantt (Google Maps).....	54
Figura 16. Diagrama de Gantt (TTS)	54
Figura 17. Diagrama de Gantt (Localización)	55
Figura 18. Diagrama de Gantt (Gestión archivos)	56
Figura 19. Diagrama de Gantt (Manual de Usuario)	56
Figura 20. Diagrama de Gantt (Entrega)	57
Figura 21. Diagrama de secuencia Caso de Uso CU-003	86
Figura 22. Diagrama de secuencia Caso de Uso CU-009	87
Figura 23. Diagrama conceptual de clases (Parte 1)	88

Figura 24. Diagrama conceptual de clases (Parte 2) 89

Figura 25. Interfaz principal dispositivo 103

Figura 26. Interfaz Portada..... 104

Figura 27. Interfaz MenuInicial 105

Figura 28. Interfaz Usuarios 106

Figura 29. Interfaz UsuariosFormulario..... 107

Figura 30. Interfaz Login..... 108

Figura 31. Interfaz MainActivity 109

Figura 32. Interfaz Favoritos 110

Figura 33. Interfaz MostrarMapa 111

Figura 34. Interfaz ModoLibre (modelo 1) 112

Figura 35. Interfaz ModoLibre (modelo 2) 113

Figura 36. Interfaz ModoLibre (modelo 3) 114

Figura 37. Interfaz ModoLibre (modelo 4) 115

Figura 38. Interfaz FinDeLaRuta 116

Figura 39. Diagrama de secuencia Caso de Uso CU-003 128

Figura 40. Diagrama de secuencia Caso de Uso CU-009 128

Figura 41. Diagrama de clases de diseño (Parte 1) 129

Figura 42. Diagrama de clases de diseño (Parte 2) 130

Figura 43. Captura de base de datos FAVORITOS 143

Figura 44. Captura de documento de radares 144

Figura 45. Captura de ruta generada libremente por el usuario 145

Figura 46. Modelo entidad/relación 148

Figura 47. Modelo relacional..... 148

Figura 48. Modelo de tablas..... 149

Figura 49. Botón inicio aplicación	176
Figura 50. Menu inicial	177
Figura 51. Botón de acceso a base de datos de usuarios.....	178
Figura 52. Botón de acceso a la aplicación.....	178
Figura 53. Botón actualizar radares	178
Figura 54. Botón acceso servicios ubicación	179
Figura 55. Interfaz lista de usuarios	180
Figura 56. Botón de creación de usuario	180
Figura 57. Opciones click largo usuario.....	181
Figura 58. Confirmación borrar usuarios	181
Figura 59. Edición y creación de usuarios	182
Figura 60. Botón editar usuarios	182
Figura 61. Botón crear usuario.....	183
Figura 62. Interfaz de autenticación de usuarios.....	183
Figura 63. Botón acceso aplicación	183
Figura 64. Interfaz menú principal	184
Figura 65. Botones de modo de ruta.....	185
Figura 66. Botón micrófono	185
Figura 67. Botón de acceso a favoritos	186
Figura 68. Interfaz Favoritos	187
Figura 69. Opciones ruta favoritos.....	188
Figura 70. Botón nuevo favorito	188
Figura 71. Interfaz seguimiento de ruta.....	189
Figura 72. Interfaz Modo Libre.....	190
Figura 73. Botón cargar rutas.....	191

Figura 74. Listado de rutas libres almacenadas	191
Figura 75. Modo Pelea contra ti mismo	192
Figura 76. Botón empezar	192
Figura 77. Etiqueta tiempo y botón parar	193
Figura 78. Cuadro Ruta Libre.....	193
Figura 79. Interfaz resumen de ruta.....	194
Figura 80. Botón guardar ruta.....	195
Figura 81. Cuadro central guardar ruta.....	195
Figura 82. Botón volver	195
Figura 83. Ejemplo algoritmo polilíneas codificadas (valor)	197
Figura 84. Ejemplo de codificación de serie	198
Figura 85. Cálculo de distancia mediante estimación inversa	199
Figura 86. Variación polo norte magnético 1831-2007	200
Figura 87. Cálculo de bearing	201

Índice de tablas

Tabla 1. Requisito de usuario RU-C-001.....	28
Tabla 2. Requisito de usuario RU-C-002.....	28
Tabla 3. Requisito de usuario RU-C-003.....	29
Tabla 4. Requisito de usuario RU-C-004.....	29
Tabla 5. Requisito de usuario RU-C-005.....	30
Tabla 6. Requisito de usuario RU-C-006.....	30
Tabla 7. Requisito de usuario RU-C-007.....	31
Tabla 8. Requisito de usuario RU-C-008.....	31
Tabla 9. Requisito de usuario RU-C-009.....	32
Tabla 10. Requisito de usuario RU-C-010.....	32
Tabla 11. Requisito de usuario RU-C-011.....	33
Tabla 12. Requisito de usuario RU-C-012.....	33
Tabla 13. Requisito de usuario RU-C-013.....	34
Tabla 14. Requisito de usuario RU-C-014.....	34
Tabla 15. Requisito de usuario RU-C-015.....	35
Tabla 16. Requisito de usuario RU-C-016.....	35
Tabla 17. Requisito de restricción RU-R-001.....	36
Tabla 18. Requisito de restricción RU-R-002.....	36
Tabla 19. Requisito de restricción RU-R-003.....	37
Tabla 20. Requisito de restricción RU-R-004.....	37
Tabla 21. Requisito de restricción RU-R-005.....	38
Tabla 22. Requisito de restricción RU-R-006.....	38
Tabla 23. Requisito de restricción RU-R-007.....	39
Tabla 24. Coste Licencias.....	57

Tabla 25. Coste recursos humanos	58
Tabla 26. Coste total del proyecto	58
Tabla 27. Requisito de funcionalidad RS-F-001.....	63
Tabla 28. Requisito de funcionalidad RS-F-002.....	63
Tabla 29. Requisito de funcionalidad RS-F-003.....	64
Tabla 30. Requisito de funcionalidad RS-F-004.....	64
Tabla 31. Requisito de funcionalidad RS-F-005.....	65
Tabla 32. Requisito de funcionalidad RS-F-006.....	65
Tabla 33. Requisito de funcionalidad RS-F-007.....	66
Tabla 34. Requisito de funcionalidad RS-F-008.....	66
Tabla 35. Requisito de funcionalidad RS-F-009.....	67
Tabla 36. Requisito de funcionalidad RS-F-010.....	67
Tabla 37. Requisito de funcionalidad RS-F-011.....	68
Tabla 38. Requisito de funcionalidad RS-F-012.....	68
Tabla 39. Requisito de funcionalidad RS-F-013.....	69
Tabla 40. Requisito de funcionalidad RS-F-014.....	69
Tabla 41. Requisito de funcionalidad RS-F-015.....	70
Tabla 42. Requisito software RS-I-001	70
Tabla 43. Requisito software RS-O-001.....	71
Tabla 44. Requisito software RS-O-002.....	71
Tabla 45. Requisito software RS-O-003.....	72
Tabla 46. Requisito software RS-O-004.....	72
Tabla 47. Requisito software RS-O-005.....	73
Tabla 48. Requisito software RS-O-006.....	73
Tabla 49. Requisito software RS-D-001.....	74

Tabla 50. Requisito calidad RS-C-001	74
Tabla 51. Casos de uso	76
Tabla 52. Caso de uso CU-001	77
Tabla 53. Caso de uso CU-002	77
Tabla 54. Caso de uso CU-003	78
Tabla 55. Caso de uso CU-004	78
Tabla 56. Caso de uso CU-005	79
Tabla 57. Caso de uso CU-006	79
Tabla 58. Caso de uso CU-007	80
Tabla 59. Caso de uso CU-008	80
Tabla 60. Caso de uso CU-009	81
Tabla 61. Caso de uso CU-010	81
Tabla 62. Caso de uso CU-011	82
Tabla 63. Caso de uso CU-012	82
Tabla 64. Caso de uso CU-013	83
Tabla 65. Caso de uso CU-014	83
Tabla 66. Caso de uso CU-015	84
Tabla 67. Clases asociadas a Casos de uso	85
Tabla 68. Clase AndroidXMLParsingActivity	90
Tabla 69. Clase CalcularDistanciaPuntos	90
Tabla 70. Clase Favoritos	90
Tabla 71. Clase FavoritosCursorAdapter	90
Tabla 72. Clase FavoritosDbAdapter	91
Tabla 73. Clase FavoritosDbHelper	91
Tabla 74. Clase FavoritosDescargaArchivo	91

Tabla 75. Clase FinDeLaRuta	91
Tabla 76. Clase GPSTracker	92
Tabla 77. Clase Login	92
Tabla 78. Clase Mail.....	92
Tabla 79. Clase MainActivity	92
Tabla 80. Clase MenuInicial.....	93
Tabla 81. Clase ModoLibre	93
Tabla 82. Clase MostrarMapa	93
Tabla 83. Clase Portada.....	93
Tabla 84. Clase RutasHelper.....	94
Tabla 85. Clase Usuarios.....	94
Tabla 86. Clase UsuariosCursorAdapter	94
Tabla 87. Clase UsuariosDbAdapter	94
Tabla 88. Clase UsuariosDbHelper	95
Tabla 89. Clase UsuariosFormulario.....	95
Tabla 90. Clase XMLParser	95
Tabla 91. Asociación clases Portada y MenuInicial	95
Tabla 92. Asociación clases GPSTracker y MenuInicial	96
Tabla 93. Asociación clases MenuInicial y Usuarios.....	96
Tabla 94. Asociación clases Usuarios y UsuariosFormulario.....	96
Tabla 95. Asociación clases Usuarios y UsuariosDbHelper	96
Tabla 96. Asociación clases UsuariosDbHelper, UsuariosDbAdapter y UsuariosCursorAdapter	97
Tabla 97. Asociación clases UsuariosFormulario y Mail.....	97
Tabla 98. Asociación clases MenuInicial y Login	97
Tabla 99. Asociación clases Login y MainActivity.....	97

Tabla 100. Asociación clases MainActivity y GPSTracker	97
Tabla 101. Asociación clases MainActivity y ModoLibre.....	98
Tabla 102. Asociación clases ModoLibre y FinDeLaRuta.....	98
Tabla 103. Asociación clases ModoLibre y GPSTracker	98
Tabla 104. Asociación clases MainActivity y MostrarMapa	98
Tabla 105. Asociación clases MostrarMapa y RutasHelper	99
Tabla 106. Asociación clases MostrarMapa, XMLParser y AndroidXMLParsingActivity	99
Tabla 107. Asociación clases MostrarMapa y CalcularDistanciaPuntos	99
Tabla 108. Asociación clases MainActivity y Favoritos	99
Tabla 109. Asociación clases Favoritos y FavoritosDescargaArchivo	100
Tabla 110. Asociación clases FavoritosDbHelper, FavoritosDbAdapter y FavoritosCursorAdapter	100
Tabla 111. Matriz de trazabilidad Requisitos de Usuario - Casos de Uso	118
Tabla 112. Matriz de trazabilidad Requisitos Software - Casos de Uso	119
Tabla 113. Clases de diseño asociadas a Casos de uso	127
Tabla 114. Atributos, funciones y procedimientos clase AndroidXMLParsingActivity	131
Tabla 115. Atributos, funciones y procedimientos clase CalcularDistanciaPuntos	131
Tabla 116. Atributos, funciones y procedimientos clase Favoritos	132
Tabla 117. Atributos, funciones y procedimientos clase FavoritosCursorAdapter.....	132
Tabla 118. Atributos, funciones y procedimientos clase FavoritosDbAdapter	133
Tabla 119Atributos, funciones y procedimientos clase FavoritosDbHelper	133
Tabla 120Atributos, funciones y procedimientos clase FavoritosDescargaArchivo	134
Tabla 121. Atributos, funciones y procedimientos clase FinDeLaRuta	134
Tabla 122. Atributos, funciones y procedimientos clase GPSTracker	135
Tabla 123. Atributos, funciones y procedimientos clase Login	135
Tabla 124. Atributos, funciones y procedimientos clase Mail	136

Tabla 125. Atributos, funciones y procedimientos clase MainActivity	136
Tabla 126. Atributos, funciones y procedimientos clase MenuInicial	137
Tabla 127. Atributos, funciones y procedimientos clase ModoLibre.....	137
Tabla 128. Atributos, funciones y procedimientos clase MostrarMapa	138
Tabla 129. Atributos, funciones y procedimientos clase Portada.....	138
Tabla 130. Atributos, funciones y procedimientos clase RutasHelper	139
Tabla 131. Atributos, funciones y procedimientos clase Usuarios	139
Tabla 132. Atributos, funciones y procedimientos clase UsuariosCursorAdapter.....	140
Tabla 133. Atributos, funciones y procedimientos clase UsuariosDbAdapter.....	140
Tabla 134. Atributos, funciones y procedimientos clase UsuariosDbHelper	141
Tabla 135. Atributos, funciones y procedimientos clase UsuariosFormulario.....	141
Tabla 136. Atributos, funciones y procedimientos clase XMLParser	142
Tabla 137. Tabla USUARIOS	150
Tabla 138. Tabla FAVORITOS.....	150
Tabla 139. Tabla NOTIFICACIONES.....	151
Tabla 140. Matriz de trazabilidad Clases de diseño - Casos de Uso.....	153
Tabla 141. Prueba PR-A-001.....	157
Tabla 142. Prueba PR-A-002.....	157
Tabla 143. Prueba PR-A-003.....	157
Tabla 144. Prueba PR-A-004.....	158
Tabla 145. Prueba PR-A-005.....	158
Tabla 146. Prueba PR-A-006.....	158
Tabla 147. Prueba PR-A-007.....	159
Tabla 148. Prueba PR-A-008.....	159
Tabla 149. Prueba PR-A-009.....	159

Tabla 150. Prueba PR-A-010.....	160
Tabla 151. Prueba PR-A-011.....	160
Tabla 152. Prueba PR-A-012.....	160
Tabla 153. Prueba PR-A-013.....	161
Tabla 154. Prueba PR-A-014.....	161
Tabla 155. Prueba PR-A-015.....	161
Tabla 156. . Prueba PR-A-016.....	162
Tabla 157. Prueba PR-A-017.....	162
Tabla 158. Prueba PR-A-018.....	162
Tabla 159. Prueba PR-A-019.....	163
Tabla 160. Prueba PR-A-020.....	163
Tabla 161. Prueba PR-A-021.....	163
Tabla 162. Prueba PR-A-022.....	164
Tabla 163. Prueba PR-A-023.....	164
Tabla 164. Prueba PR-A-024.....	164
Tabla 165. Prueba PR-A-025.....	165
Tabla 166. Prueba PR-A-026.....	165
Tabla 167. Prueba PR-A-027.....	165
Tabla 168. Prueba PR-A-028.....	166
Tabla 169. Prueba PR-A-029.....	166
Tabla 170. Prueba PR-A-030.....	166
Tabla 171. Prueba PR-A-031.....	167
Tabla 172. Prueba PR-A-032.....	167
Tabla 173. Prueba PR-A-033.....	167
Tabla 174. Prueba PR-A-034.....	168

Tabla 175. Prueba PR-A-035.....	168
Tabla 176. Prueba PR-A-036.....	168
Tabla 177. Prueba PR-A-037.....	169
Tabla 178. Prueba PR-A-038.....	169
Tabla 179. Prueba PR-A-039.....	169
Tabla 180. Prueba PR-A-040.....	170
Tabla 181. Matriz de trazabilidad Pruebas Aceptación – Requisitos de capacidad.....	171
Tabla 182. Prueba PR-I-001.....	172
Tabla 183. Prueba PR-I-002.....	172

1. Introducción

El presente proyecto surge por el auge de los dispositivos móviles frente a los dispositivos de navegación debido a su mayor presencia en nuestra sociedad.

La intención del mismo es integrar en una aplicación varias de las posibilidades que nos dan los smartphones en el panorama actual. La intención es poder manejar mediante la voz humana distintas aplicaciones integradas en el dispositivo y emular el comportamiento de Google Maps, añadiendo algunas funcionalidades no disponibles de serie como sería el aviso de radares, el manejo de una serie de rutas favoritas o la posibilidad de crear tus propias rutas para repetirlas posteriormente.

El cliente medio cada vez demanda más aplicaciones con múltiples funciones para poder tener un dispositivo menos sobrecargado de procesos en todo momento.

En este caso, la intención es mezclar funcionalidades de aplicaciones de navegación como Google Maps o Garmin; junto a la función de un avisador de radares y además agregar parte de las funcionalidades que nos permiten realizar las aplicaciones destinadas al entrenamiento personal.

En definitiva, lo que se ha buscado es realizar una correcta integración de variados servicios y unificarlos en una sola herramienta.

2. Definiciones y acrónimos

2.1 Definiciones

- **API:** Siglas en inglés de “Application Programming Interface”. Es el conjunto de funciones y procedimientos que ofrece una biblioteca para ser utilizada por otro software como capa de abstracción.

- **Array:** Tipo de dato estructurado que permite almacenar un conjunto de datos homogéneos, es decir, todos del mismo tipo y relacionados.

- **DOM:** Siglas en inglés de “Document Object Model”. Es una API que proporciona un conjunto estándar de objetos para representar documentos HTML y XML. Sirve para acceder, añadir y cambiar dinámicamente contenido.

- **GUI:** Siglas en inglés de “Graphical User Interface”. Es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador

- **IDE:** Siglas en inglés de “Integrated Development Environment”. Es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

- **POI:** Siglas en inglés de “Point Of Interest”. Se usa para denominar aquellos puntos geográficos con una importancia determinada. En nuestra aplicación, consideraremos un punto de interés tanto a los radares fijos como a los puntos de paso que nos de Google Maps para realizar una ruta.

- **Stakeholder:** Quienes pueden afectar o son afectados por las actividades de una empresa

- **TTS:** Siglas en inglés de “Text To Speech”. Conversión de lenguaje escrito a voz.

- **XML:** Siglas en inglés de “eXtensible Markup Language”. Lenguaje de texto para el almacenamiento de datos de forma legible. Valido para bases de datos, editores de texto y hojas de cálculo entre otras utilidades.

2.2. Acrónimos

- CU: Caso de uso

- RS: Requisito Software

- RU-C: Requisito de Usuario de Capacidad
- RU-R: Requisito de Usuario de Restricción

3. Estudio de Viabilidad del Sistema

El objetivo del Estudio de Viabilidad del Sistema es realizar un análisis detallado de las necesidades del cliente, con el fin de proponer una solución en un plazo establecido. Este estudio debe tener en cuenta restricciones de diversa índole: económicas, técnicas, legales y operativas.

Basándose en las necesidades del cliente, se detallará el alcance del sistema, es decir, se estudiará el alcance de la necesidad planteada por el cliente identificando los primeros requisitos, estructuras implicadas, suposiciones y restricciones así como los stakeholders, el equipo de trabajo y la planificación a seguir por éste.

Una vez definido el alcance del sistema se realizará un estudio de la situación actual de los sistemas similares existentes en la actualidad. El objetivo de esta tarea consiste en identificar los recursos de información existentes, posibles problemas y mejoras.

A continuación se realizará una definición de los requisitos que debe cumplir el sistema, de manera que se obtengan un conjunto de necesidades detalladas, no ambiguas, y completas, que sirva de base para las siguientes etapas del ciclo de vida del proyecto. Dichos requisitos se describirán en un lenguaje sencillo con el propósito de que el cliente los comprenda y pueda validarlos sin problemas.

Finalmente se expondrá la solución que he elegido para el desarrollo del trabajo.

3.1 Establecimiento del alcance del sistema

El objetivo de esta actividad es realizar una primera aproximación al estudio de los requisitos, identificar las unidades organizativas afectadas y analizar las suposiciones y restricciones del sistema. También se identificarán a las personas que deben participar en el estudio de viabilidad, así como sus perfiles, especificando sus tareas y responsabilidades dentro del proceso.

3.1.1 Estudio de la solicitud

En la solicitud del cliente se especifica la necesidad de desarrollar una aplicación capaz de analizar el tratamiento de comandos orales, traduciendo los mismos para su uso en la aplicación, la creación de rutas para viajes y el almacenamiento de las mismas para mayor velocidad de creación.

Con el desarrollo de este proyecto se pretende desarrollar una aplicación que cumpla las siguientes características:

- Interaccionar con el usuario a través del dispositivo.
- Generar rutas de viaje.

- Controlar errores en la introducción de parámetros de entrada por parte del usuario.
- Almacenamiento de diversos datos de usuarios y rutas en bases de datos.
- Uso offline de rutas previamente programadas.
- Crear rutas desde cero para poder repetirlas cuando se decida.
- Extremar la funcionalidad. El cliente busca algo sencillo, rápido e intuitivo.

Tras el estudio de la solicitud del cliente, el equipo de desarrollo del proyecto ha determinado que el cliente propone un proyecto perfectamente viable.

La persona encargada de llevar a cabo este proyecto es Israel González Carrasco, como director del mismo, y Manuel Zuil Oliva, como desarrollador.

3.1.2 Identificación del alcance del sistema

El presente proyecto comprende una aplicación a desarrollar que permita la conversión de voz a texto de diversas órdenes orales que realice el usuario. Por otro lado, el tratamiento de ficheros en tres formatos: formato XML, formato SQLite y texto plano, dependiendo del modo de uso.

En este apartado se pretenden identificar los requisitos de cada una de las partes, identificando las principales funcionalidades que se deberán desarrollar.

En primer lugar, sobre tratamiento de ficheros debe proporcionar al usuario la funcionalidad principal de generar los ficheros necesarios para el posterior manejo de la aplicación. Estos ficheros se generan de manera automática al acceder a determinados puntos de la aplicación, atendiendo a una serie de parámetros prefijados por el programador, tales como:

- Base de datos de usuarios: ID, usuario, clave, dirección e-mail
- Base de datos de favoritos: ID, usuario, punto de inicio y final, modo de ruta.
- Ruta libre: Texto plano donde se almacena latitud y longitud, separadas por la cadena “;”.
- Ruta prefijada: Archivo “.xml” descargado desde Internet con una serie de parámetros que explicaremos a posteriori.

Merece la pena destacar que la aplicación debe ser capaz de tratar los ficheros de datos originales descargados desde Google Maps, tratando todas las posibles excepciones que estos nos dan (rutas inviables, puntos de inicio o llegada inexistentes).

Otro bloque de funcionalidades, es emplear la capacidad del dispositivo para conocer su localización para poder desarrollar múltiples tareas, como por ejemplo, seguir una ruta

generada a través de la interpretación de los ficheros de Google Maps, o bien, crear rutas de entrenamiento con total y absoluta libertad para luego poder competir contra ti mismo para así mejorar tus tiempos.

El sistema, tendrá la capacidad de ser manejado en gran parte con la voz humana y de recibir instrucciones a través de la misma para así poder centrarse en el seguimiento de una ruta sin tener que volver los ojos hacia el dispositivo.

3.1.3 Identificación de los interesados en el sistema (stakeholders)

En este punto se lleva a cabo una identificación de todas aquellas personas o entidades interesadas de alguna forma en el proyecto. Se describen a continuación:

- Cliente: Es la persona que realiza la solicitud de desarrollo del sistema, en este caso, el cliente es Israel González Carrasco, profesor de la Universidad Carlos III de Madrid.
- Usuarios gestores: Son aquellas personas que harán uso de la aplicación.
- Tutor y coordinador del proyecto: Israel González Carrasco
- Autor del Proyecto: Manuel Zuil Oliva. Es la persona encargada de la realización del proyecto, siendo además el máximo responsable del mismo y de todos los productos generados durante su desarrollo.

3.2 Estudio de la situación actual

La situación actual es el estado en el que se encuentran los sistemas de navegación existentes en el momento en el que se inicia su estudio.

En la actualidad, las aplicaciones de GPS para Android es un terreno bastante explorado y en el cual podemos confirmar que este proyecto lo que realiza es un acercamiento a las mismas, tratando de incluir nuevas funcionalidades e integrando lo que creemos más útil de las ya existentes.

Por otro lado, las aplicaciones para realizar rutas propias ya sea a pie o en vehículo, tampoco incorporan todas las opciones de un GPS estándar.

Haciendo un pequeño repaso, se van a presentar las opciones más completas del mercado actual:

- **Google Maps:** El indiscutible líder del sector debido principalmente a dos motivos, el primero es su gratuidad debido a que es una herramienta nativa de Google y viene implementada de serie en gran parte de los dispositivos. Por otro lado, por este mismo motivo, dispone de una ingente cantidad de información gracias al feedback recibido por los usuarios gracias al cual puede informar del tráfico en tiempo real.

- **Waze:** Este GPS tiene una peor infraestructura inicial que Google Maps pero lo compensa con sus atributos sociales. Se permite a los propios usuarios introducir puntos con accidentes, controles o radares; para así crear una red de colaboración ciudadana que mantenga la seguridad.

- **Tomtom:** El primero en llegar a este sector sigue teniendo mapas bien actualizados y fiables con caminos situados en lugares bastante alejados de cualquier lugar poblado. Su gran contra es el precio.

Una vez realizado el estudio de las técnicas y herramientas existentes en la actualidad, se puede concluir que existen varios sistemas que nos proporcionan funcionalidades de nuestra aplicación por separado, pero no existe ninguno que aglutine todas, por lo cual lo que se pretende alcanzar es una integración total de dichas funcionalidades.

Mientras que la mayoría de sistemas y aplicaciones existentes se han desarrollado para la resolución de un problema concreto, con el presente proyecto se pretende desarrollar la arquitectura para la resolución de problemas de cualquier dominio sobre el que se tengan datos suficientes.

3.3 Definición de los requisitos del sistema

Una vez realizado el estudio de la solicitud del cliente se realiza la extracción de requisitos con el fin de cumplir con las funcionalidades debe proporcionar la aplicación.

Esta extracción de requisitos se realiza también con el objetivo de orientar al cliente, permitiendo al equipo de desarrollo obtener una lista de requisitos detallada, completa y sin ambigüedades. Estos requisitos proporcionarán una visión general de la aplicación, sin ahondar en aspectos técnicos, estableciendo las principales funcionalidades y restricciones, sirviendo de base a posteriores procesos del ciclo de vida.

3.3.1 Obtención de los requisitos

En esta tarea se realiza la obtención detallada de requisitos de usuario mediante sesiones de trabajo con el cliente. La lista de requisitos obtenida no es definitiva y podrá ser modificada durante el proyecto, añadiendo requisitos no contemplados, o bien eliminando o modificando requisitos incorrectos, de forma que se terminen cubriendo todas las funcionalidades expuestas por el cliente. La definición de requisitos recoge lo que quiere el cliente y lo que necesita, englobando los requisitos obtenidos del usuario en dos grandes categorías:

- **Requisitos de capacidad:** Representan lo que necesitan los usuarios para resolver un problema o lograr un objetivo.
- **Requisitos de restricción:** Son las restricciones impuestas por los usuarios sobre cómo se debe resolver el problema o cómo se debe alcanzar el objetivo.

Cada requisito de usuario debe incluir en su definición una serie de atributos, los cuales proporcionen toda la información necesaria para su seguimiento posterior y su clasificación. Estos atributos se describen a continuación:

- **Identificador:** Cada requisito de usuario debe estar identificado de forma única. Este identificador tendrá el siguiente formato: RU-C-*nnn* o RU-R-*nnn*, donde:

- RU: Indica que se trata de un requisito de usuario.
- C: Indica que se trata de un requisito de usuario de capacidad.
- R: Indica que se trata de un requisito de usuario de restricción.
- *nnn*: Tomará valores numéricos dentro del rango 000-999.

- **Prioridad:** Se asignará una prioridad a cada requisito con el fin de poder realizar una planificación correcta durante fases posteriores. Su clasificación puede tomar los valores: alta, media y baja.

- **Necesidad:** Los requisitos clasificados como esenciales para el usuario no pueden ser eliminados, mientras que los demás requisitos estarán siempre sujetos a modificación en el caso de que exista una causa que lo justifique.

Por lo tanto, se clasificarán descendientemente de acuerdo a su necesidad de la siguiente forma: esencial, deseable y opcional.

- **Claridad:** Identifica la falta o existencia de ambigüedad de un requisito, esto es, si puede ser interpretado de varias formas dependiendo del contexto. Los requisitos serán clasificados de acuerdo a su claridad de forma descendente de la siguiente forma: alta, media y baja.

- **Fuente:** Identifica el origen del requisito, que puede estar en el usuario, una fuente externa como un documento, o el propio equipo de desarrollo del proyecto.

- **Estabilidad:** Algunos requisitos pueden no estar sujetos a cambios durante el proyecto debido a su naturaleza, mientras que otros pueden estar sujetos a determinados cambios por el desarrollo de la etapa de diseño o los requisitos software. Los requisitos poco estables deberán ser observados con mayor cuidado durante el desarrollo del proyecto al ser susceptibles a cambios. Cada requisito se clasificará en: estable o inestable.

- **Verificabilidad:** Indica si el cumplimiento de un requisito en el sistema puede ser susceptible de comprobación, esto es, si se puede verificar que el requisito se ha incorporado en el diseño y que en el sistema se puede verificar su cumplimiento. La verificabilidad de cada requisito se clasificará en: alta, media y baja.

3.3.1.1. Requisitos de capacidad

A continuación se presentan los requisitos de capacidad, los cuales representan lo que necesita el usuario para resolver un problema o lograr un objetivo.

Identificador	RU-C-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Generar bases de datos en formato SQLite para almacenar datos según las pautas marcadas en [9]		

Tabla 1. Requisito de usuario RU-C-001

Identificador	RU-C-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Documento
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Inestable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Lectura y captura de datos de archivos con extensión .xml según las pautas marcadas en [4]		

Tabla 2. Requisito de usuario RU-C-002

Identificador	RU-C-003		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Usuario
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Inestable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Actualización de la posición del usuario en tiempo real según las pautas marcadas en [8]		

Tabla 3. Requisito de usuario RU-C-003

Identificador	RU-C-004		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Documento/Usuario
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Inestable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Trazo de rutas sobre mapas a partir de instrucciones de un documento descargado previamente o sobre la posición del usuario según las pautas marcadas en [3]		

Tabla 4. Requisito de usuario RU-C-004

Identificador	RU-C-005		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Comprensión y traducción de voz a texto y tratado del mismo según las pautas marcadas en [5]		

Tabla 5. Requisito de usuario RU-C-005

Identificador	RU-C-006		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Crear ubicación de entrada de ficheros descargados desde Internet o creados por el usuario según las pautas marcadas en [1]		

Tabla 6. Requisito de usuario RU-C-006

Identificador	RU-C-007		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Usuario
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Crear ficheros con ubicaciones automáticamente a elección del usuario según las pautas marcadas en [1]		

Tabla 7. Requisito de usuario RU-C-007

Identificador	RU-C-008		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Documento
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Inestable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Añadir rutas a la lista de favoritos con los parámetros de configuración seleccionados según las pautas marcadas en [9]		

Tabla 8. Requisito de usuario RU-C-008

Identificador	RU-C-009		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Eliminar rutas a la lista de favoritos limpiando toda la información sobre la misma almacenada en el sistema según las pautas marcadas en [9]		

Tabla 9, Requisito de usuario RU-C-009

Identificador	RU-C-010		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Creación, lectura e interpretación de ficheros en formato de texto plano (.txt).		

Tabla 10. Requisito de usuario RU-C-010

Identificador	RU-C-011		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Dibujar rutas a través de los siguientes formatos: <ul style="list-style-type: none"> - Algoritmo de polilíneas codificadas según las pautas marcadas en [3] - Unión punto a punto de coordenadas según las pautas marcadas en [1] 		

Tabla 11. Requisito de usuario RU-C-011

Identificador	RU-C-012		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Usuario
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Gestionar la creación, edición y borrado de usuarios según las pautas marcadas en [9]		

Tabla 12. Requisito de usuario RU-C-012

Identificador	RU-C-013		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Cerrar procesos internos abiertos durante la aplicación para la gestión de la memoria del dispositivo.		

Tabla 13. Requisito de usuario RU-C-013

Identificador	RU-C-014		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Usuario
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Activar el posicionamiento del dispositivo a elección del usuario según las pautas marcadas en [8]		

Tabla 14, Requisito de usuario RU-C-014

Identificador	RU-C-015		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Generar informes de rutas diseñadas por el usuario para conocer sus parámetros de velocidad.		

Tabla 15. Requisito de usuario RU-C-015

Identificador	RU-C-016		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Actualización de ficheros con nuevos datos en caso de cumplir determinadas condiciones según las pautas marcadas en [2]		

Tabla 16. Requisito de usuario RU-C-016

3.3.1.2 Requisitos de restricción

Son las restricciones impuestas acerca de cómo se debe resolver el problema o como se debe alcanzar el objetivo.

Identificador	RU-R-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Generar archivos en formato de texto plano y con extensión “.txt” según las pautas marcadas en [2]		

Tabla 17. Requisito de restricción RU-R-001

Identificador	RU-R-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Disponer de una interfaz sencilla e intuitiva que facilite el uso de la aplicación.		

Tabla 18. Requisito de restricción RU-R-002

Identificador	RU-R-003		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Mostrar un aviso durante la realización de trabajos de descarga e interpretación de archivos.		

Tabla 19. Requisito de restricción RU-R-003

Identificador	RU-R-004		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Almacenar usuarios y rutas en formato SQLite según las pautas marcadas en [9]		

Tabla 20. Requisito de restricción RU-R-004

Identificador	RU-R-005		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Desarrollar la aplicación en Eclipse Juno 4.2		

Tabla 21. Requisito de restricción RU-R-005

Identificador	RU-R-006		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Uso de los servicios de Google Maps para la navegación según las pautas marcadas en [7]		

Tabla 22. Requisito de restricción RU-R-006

Identificador	RU-R-007		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Uso de la versión Android 4.4.2		

Tabla 23. Requisito de restricción RU-R-007

3.3.2. Definición de los requisitos del sistema

Asimismo, esta toma de requisitos se realiza para el cliente con el fin de orientarle y obtener una lista detallada de los requisitos, completa y sin ambigüedades. Estos requisitos presentarán una visión general de la aplicación, sin ahondar en aspectos técnicos, estableciendo las principales funcionalidades y restricciones, sirviendo de base a posteriores procesos del ciclo de vida.

3.4. Estudio de alternativas de solución

Este proyecto pretende desarrollar una aplicación que sirva de alternativa a Google Maps mediante las herramientas de desarrollo de su API. La solución ha sido implementada y modificada con el paso del tiempo de desarrollo tanto por el cliente como por el desarrollador para tratar de buscar más fortalezas al trabajo frente a una gran empresa como Google.

Dado que la aplicación está basada en la integración de diversas funcionalidades para Android ya implementadas y modificadas para nuestro fin.

En cuanto al entorno de desarrollo se ha optado por ADT Bundle para Windows, implementado en Eclipse Juno al ser un entorno ya conocido en el estudio de algunas asignaturas durante la diplomatura y la adaptación a grado.

3.5 Valoración de las alternativas

No aplica porque la solución viene dada.

3.6 Selección de la solución

No aplica porque la solución viene dada.

4. Gestión del Proyecto

En este apartado se pretende realizar una planificación, seguimiento y control de las actividades y de los recursos humanos y materiales que participarán en el proyecto. Abarcando la vida del sistema desde la definición hasta la finalización de su uso.

De forma genérica, se puede decir que un ciclo de vida de un software se divide en cuatro etapas:

- 1. Análisis:** Determinación de los requisitos del sistema.
- 2. Diseño:** A partir del modelo de análisis se deducen las estructuras de datos, la estructura en la que descompone el sistema y la interfaz de usuario.
- 3. Implementación:** Construye el sistema. La salida de esta fase es código ejecutable.
- 4. Mantenimiento:** Se comprueba que se cumplen criterios de corrección y calidad. Después de la entrega se asegura que el sistema siga funcionando y adaptándose a nuevos requisitos.

Las formas de organizar y estructurar la secuencia de ejecución de las tareas en las diferentes fases de cada uno de los métodos pueden dar lugar a un tipo de ciclo de vida diferente. Los principales ciclos de vida son:

- 1. En cascada o modelo tradicional:** diferentes etapas procesadas de un modo lineal.
- 2. Prototipado:** El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles al usuario. El prototipo es evaluado por el cliente y el usuario y utilizado para refinar los requerimientos del software a ser desarrollado.
- 3. En espiral:** Se produce una cadena continua de productos, los cuales están disponibles para la exanimación y evaluación por parte del cliente. La reevaluación después de cada fase permite cambios en las percepciones de los usuarios, avances tecnológicos o perspectivas financieras.
- 4. Orientados a objetos:** La reevaluación después de cada fase permite cambios en las percepciones de los usuarios, avances tecnológicos o perspectivas financieras.

Entre los diferentes tipos de ciclos de vida, se ha optado por seguir un ciclo de vida prototipado, por los siguientes motivos:

- Permite evaluar distintas posibilidades de desarrollo, ya que en los prototipos se puede fácilmente añadir o eliminar funcionalidades.
- Permite al tutor o cliente realizar un seguimiento periódico del proyecto para verificar si se han comprendido bien los requisitos y si se está desarrollando la aplicación de acuerdo con éstos.
- El prototipo es un documento vivo de buen funcionamiento del producto final.
- Experimenta sobre los aspectos del sistema que representan mayor complejidad.

- El usuario reacciona mucho mejor ante el prototipo, sobre el que puede experimentar, que no sobre una especificación escrita.
- Se incrementa la calidad del producto final, ya que el prototipo permite trabajar.

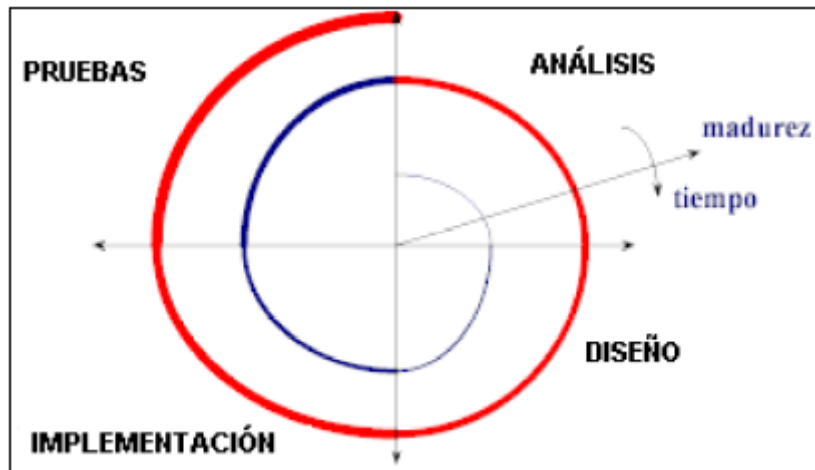


Figura 1. Ciclo de vida

4.1. Organización del proyecto

A la hora de llevar a cabo cualquier proyecto se hace necesaria la tarea de realizar una planificación detallando el tiempo estimado necesario para cumplir las distintas actividades y tareas que forman parte del proyecto, así como los recursos asignados a cada una de ellas. Por ello, se ha desarrollado una planificación de trabajo en función del ciclo de vida seleccionado anteriormente, ciclo de vida por prototipos.

Se ha decidido realizar tres iteraciones completas, tras cada una de las cuales el desarrollador se reúne con el tutor para corregir o matizar distintos aspectos de la aplicación.

En lo que se refiere a los recursos humanos que intervienen en el desarrollo del proyecto, a continuación se muestran las personas y roles que participan en el proyecto. Las funciones de las que se encarga cada uno de ellos son:

- Jefe de Proyecto: Se encarga de la gestión del proyecto, su organización, planificación y supervisión a lo largo de todo el desarrollo del mismo.
- Analista de Sistemas: Se encarga de obtener y redactar los requisitos de usuario, además de modelar los procesos y tareas a codificar.
 - Diseñador: Su tarea es el diseño de las interfaces y la arquitectura del sistema.
 - Programador: Se encarga de la codificación del sistema, así como de las pruebas necesarias del mismo.

A continuación se muestra el RBS (Resource Breakdown Structure) del proyecto, una representación de forma jerárquica de los recursos tanto humanos como materiales necesarios para el desarrollo del mismo. Esta técnica de organización de proyectos software tiene por objeto representar la organización humana del proyecto, su estructura, responsabilidades, etc., así como la estructura de recursos tecnológicos y materiales.

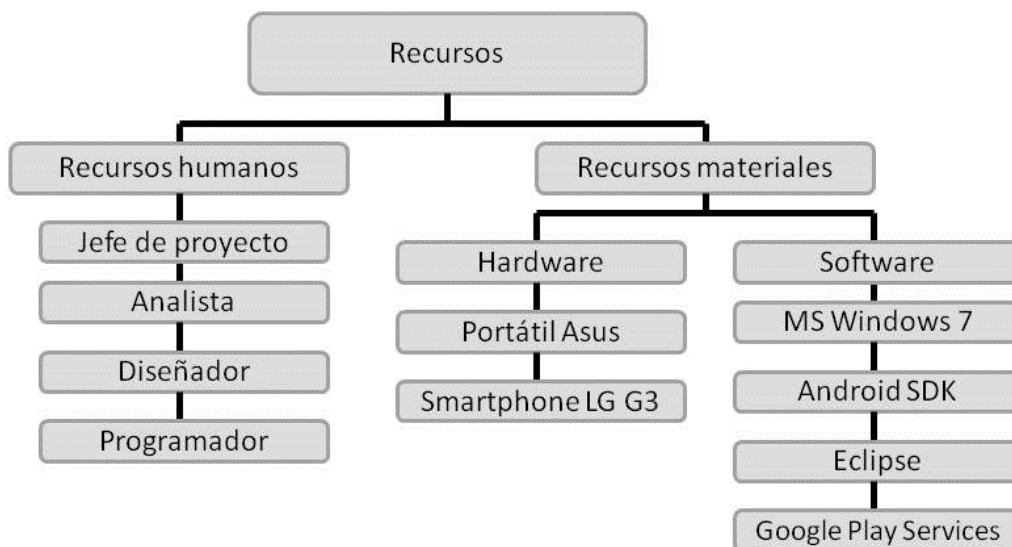


Figura 2. RBS

El WBS (Work Breakdown Structure) es una herramienta simple y práctica para definir el alcance de un proyecto. Es la base para poder saber qué se requiere hacer, para después poder realizar la planificación del proyecto. Se trata de identificar y definir las tareas que se van a realizar en el proyecto.

En el WBS se puede observar la descomposición de los procesos en subprocesos, y éstos a su vez en tareas, agrupadas por funcionalidad.

La base para en la que está basada el WBS es el modelo de proceso software IEEE 1074.

A continuación se muestra la división de las distintas tareas del proyecto mediante el WBS:

El primer proceso que se debe llevar a cabo es el estudio de viabilidad del proyecto, mediante el cual se analiza la solicitud del cliente, la situación actual y si el desarrollo del proyecto es viable. Toda la información se recoge en el documento de Estudio de Viabilidad.

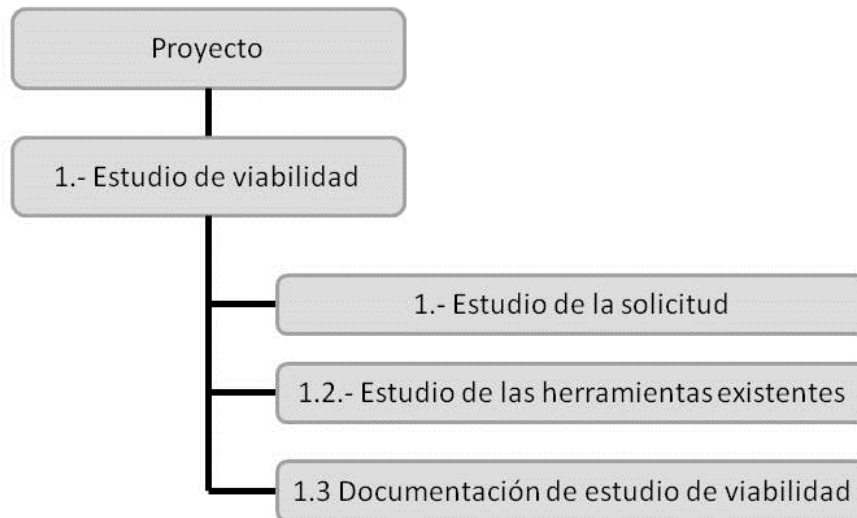


Figura 3. Proceso WBS - Estudio de viabilidad

Una vez que el desarrollo del proyecto se ha considerado viable, el siguiente proceso a realizar es la gestión del proyecto. En este proceso se debe seleccionar el modelo de ciclo de vida más adecuado para el proyecto, y se realiza tanto la planificación a seguir durante el desarrollo del proyecto como una estimación del coste de desarrollo.

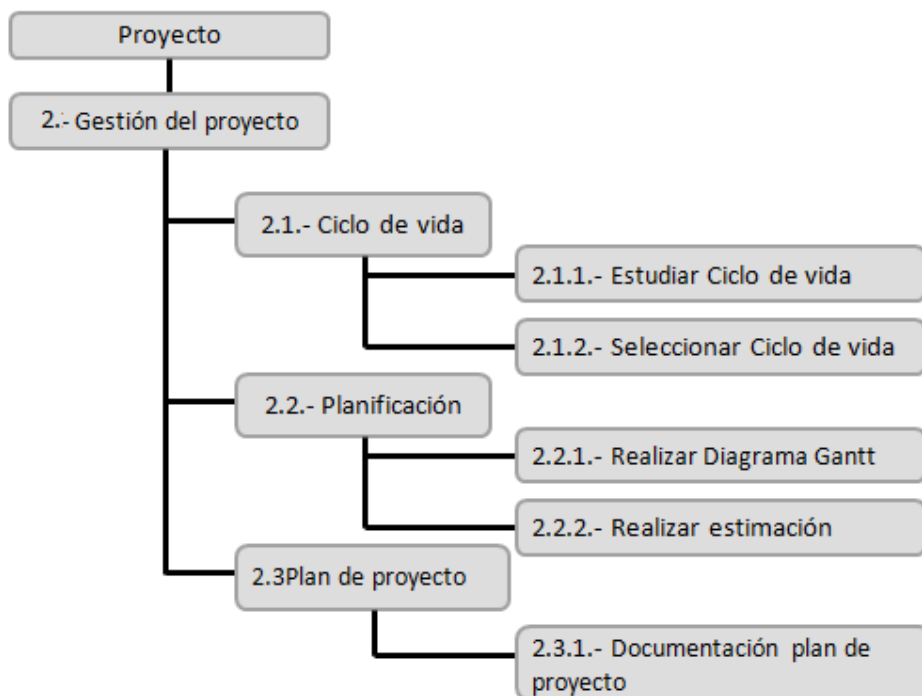


Figura 4. Proceso WBS - Gestión del proyecto

A continuación se presentan las diversas funcionalidades realizadas en el modelo de ciclo de vida elegido para el desarrollo del sistema. En cada funcionalidad se deben llevar a cabo las tareas propias del análisis, tales como la recopilación de especificaciones y requisitos, o la selección de las herramientas de desarrollo más apropiadas. Es imprescindible la realización de estas tareas antes del diseño, para asegurar que en la fase de diseño se logre crear una interfaz simple, intuitiva y funcional para el cliente.

La fase de implementación consiste principalmente en desarrollar e integrar el código referente a dicha funcionalidad con el resto de las existentes en el proyecto. Por último, este proceso consta de un subproceso imprescindible como es el de Pruebas, en el que las tareas a realizar son la definición y ejecución de pruebas para la verificación y validación del software desarrollado.

Tras la finalización de cada funcionalidad, se plantea una reunión para ver que se puede mejorar e implementar hasta llegar a un resultado completo.

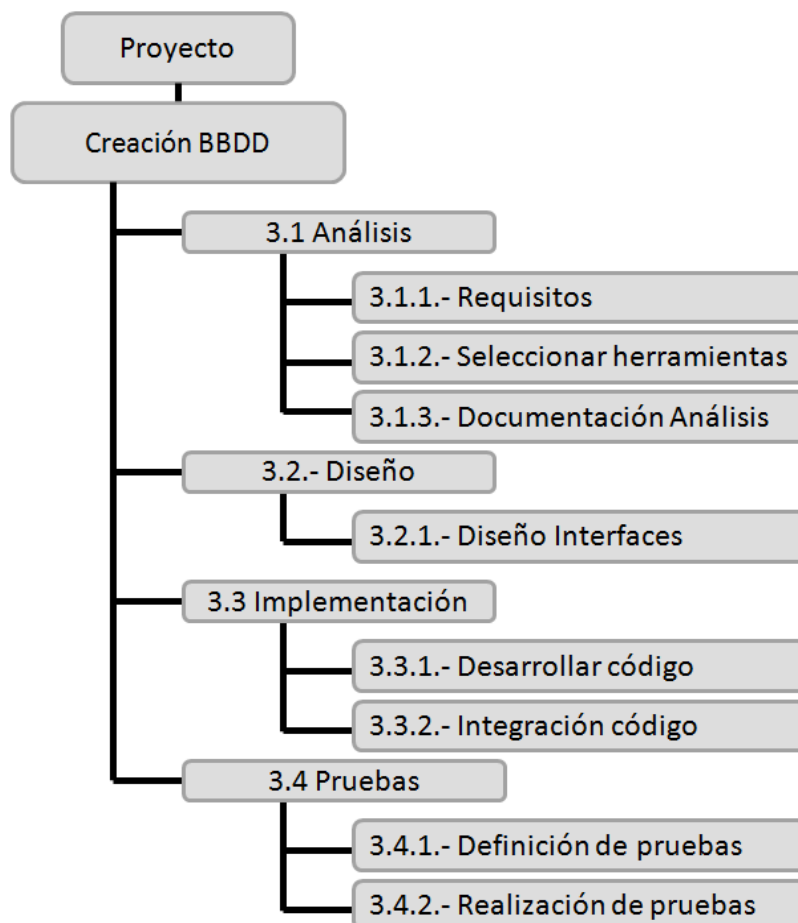


Figura 5. Proceso WBS – Creación BBDD

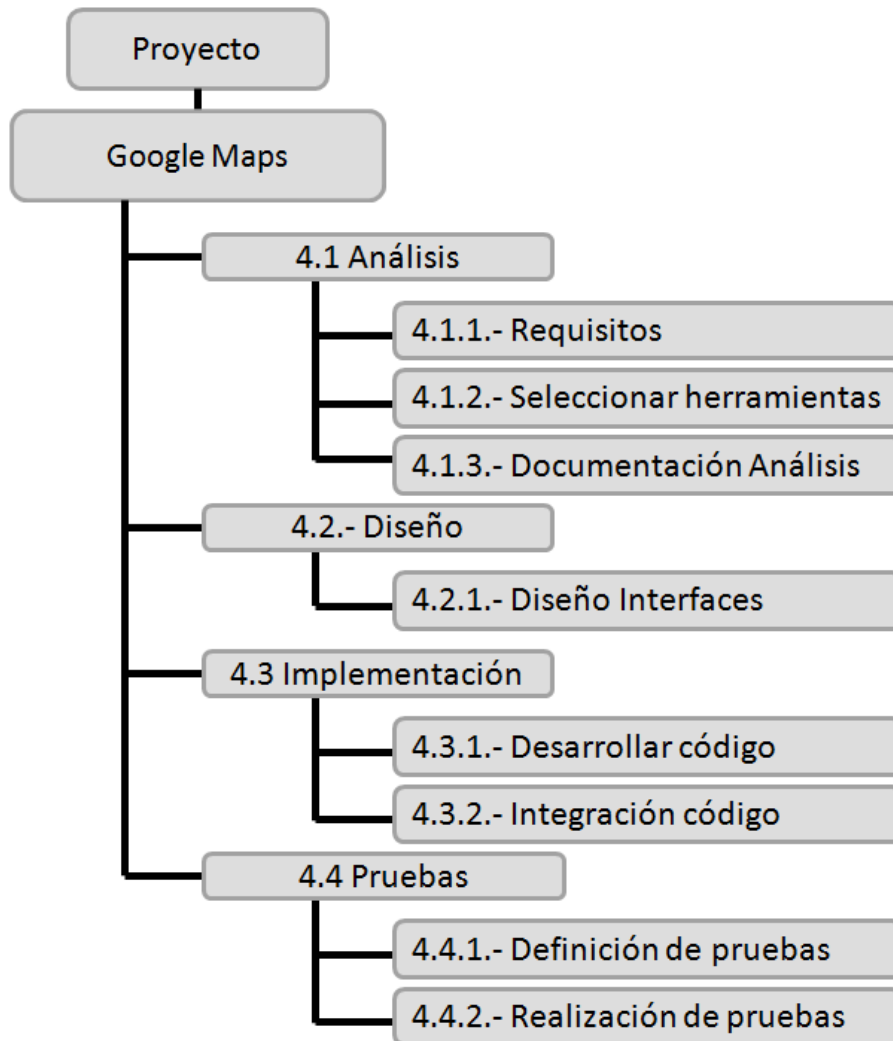


Figura 6. Proceso WBS – Google Maps

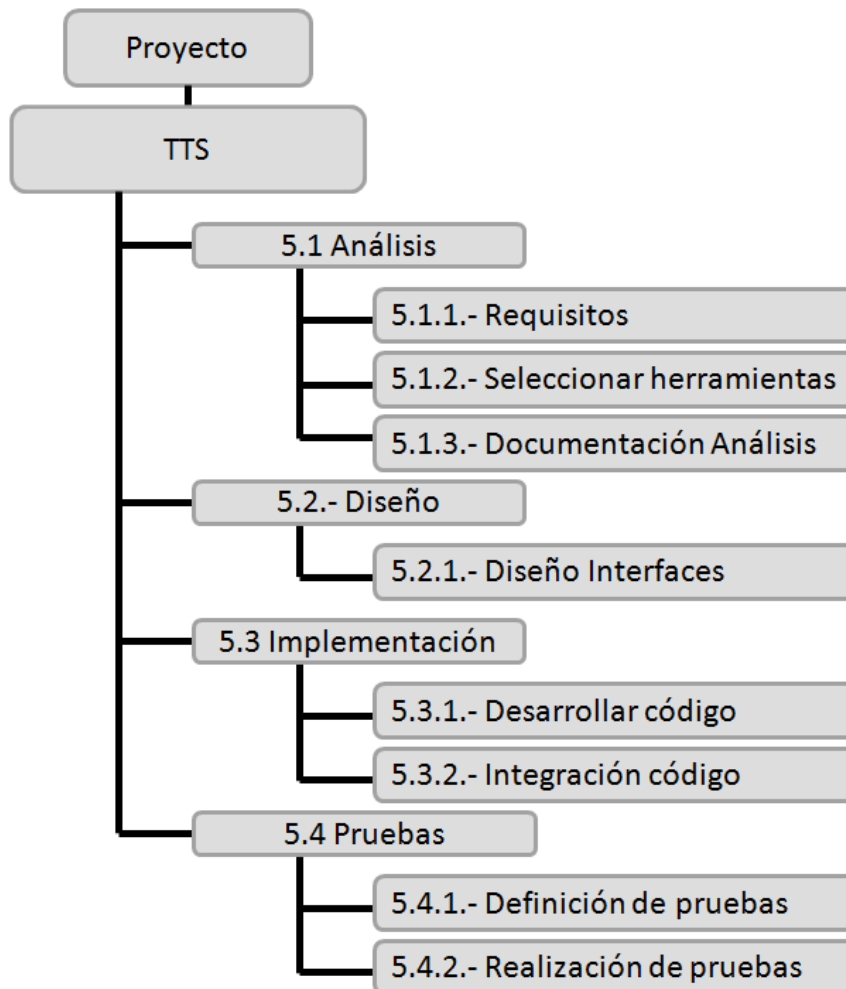


Figura 7. Proceso WBS – TTS

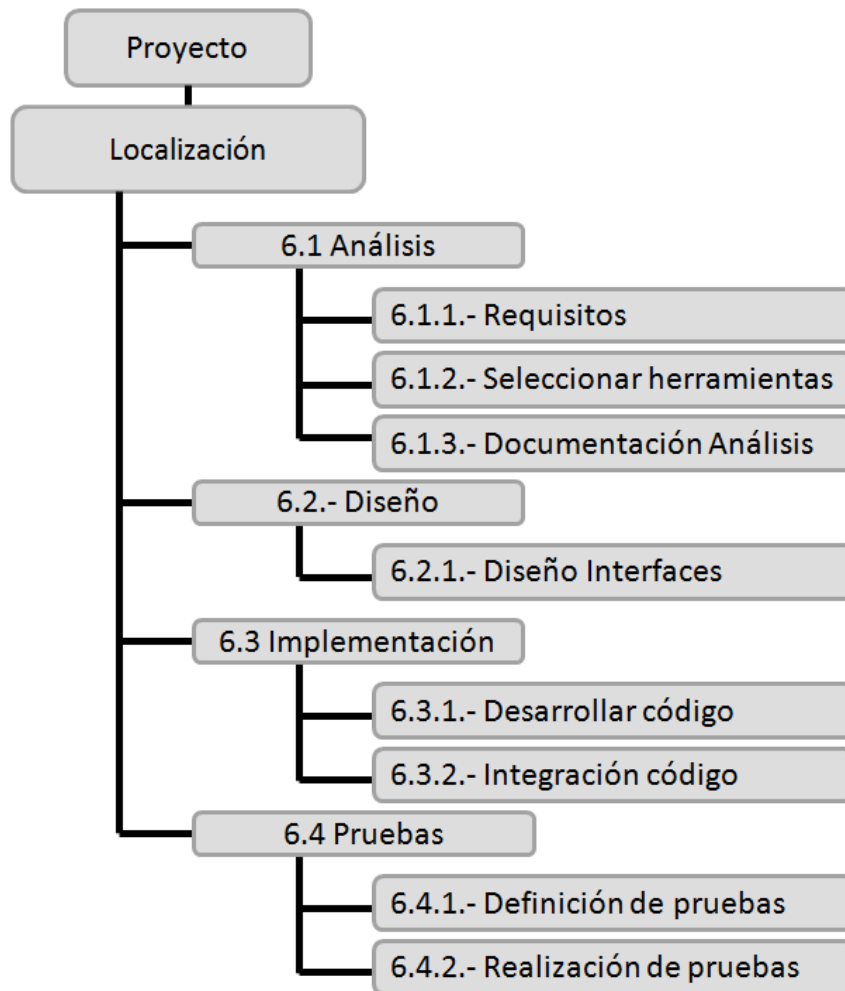


Figura 8. Proceso WBS – Localización

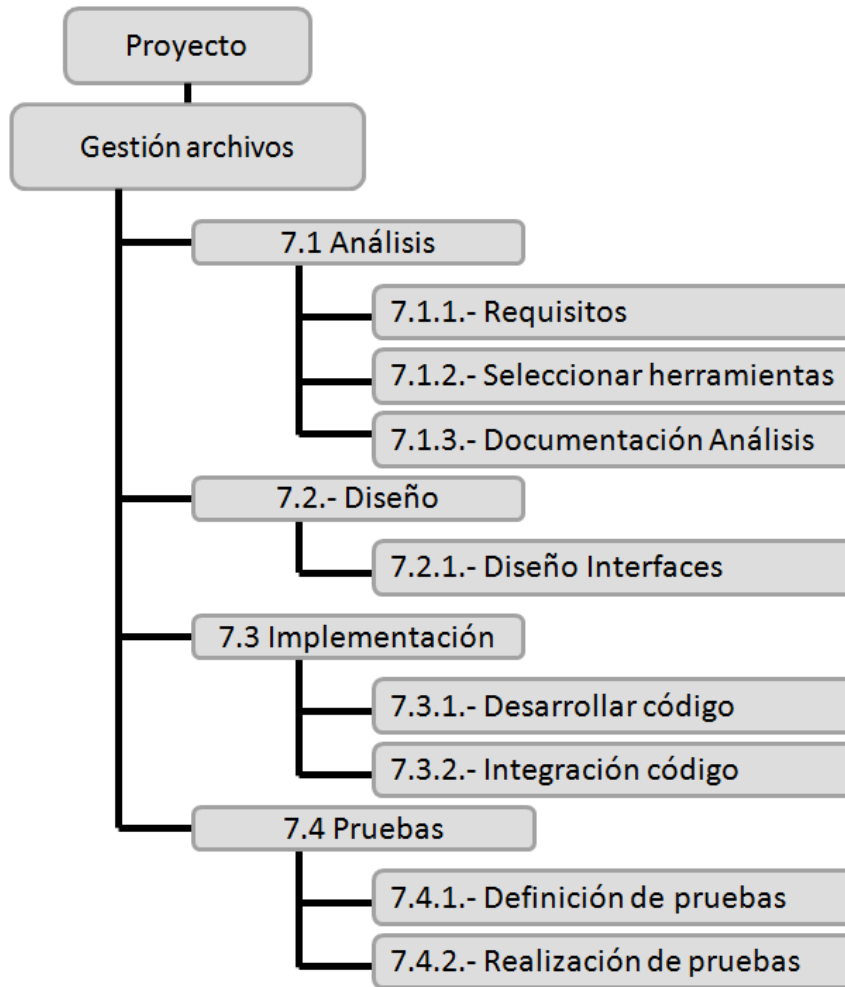


Figura 9. Proceso WBS – Gestión archivos

Cuando la última funcionalidad desarrollada es aprobada por el cliente se pasa al siguiente proceso. En este proceso se realiza el manual de usuario para que éste pueda utilizar todas las funcionalidades disponibles en el sistema con la mayor facilidad.

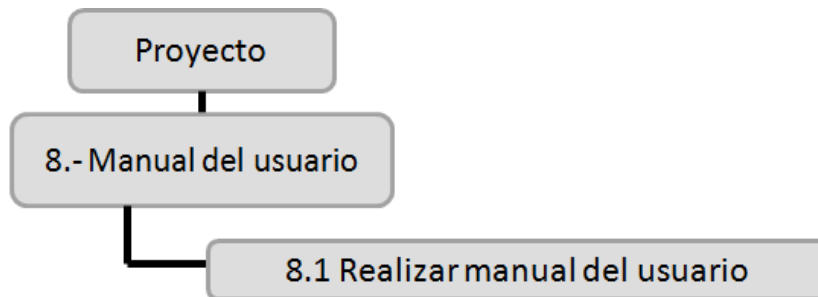


Figura 10. Proceso WBS – Manual del usuario

4.2 Planificación

La duración total del proyecto es de 190 días laborables, comenzando su desarrollo el día 12 de Mayo de 2014 y finalizando 30 de Enero de 2015, cuando el sistema se encuentre finalizado y documentado para su entrega al cliente.

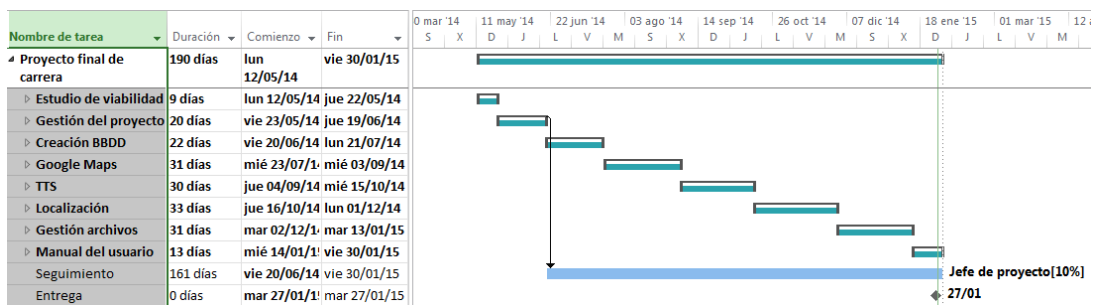


Figura 11. Diagrama de Gantt (General)

El punto de partida del proyecto es el estudio de viabilidad del mismo. Para ello, el jefe de proyecto debe realizar un estudio de la solicitud realizada por el cliente, de la situación actual y elaborar un documento que contenga dicha información. Asimismo, este documento debe indicar si la solicitud realizada es factible de ser realizada o no.

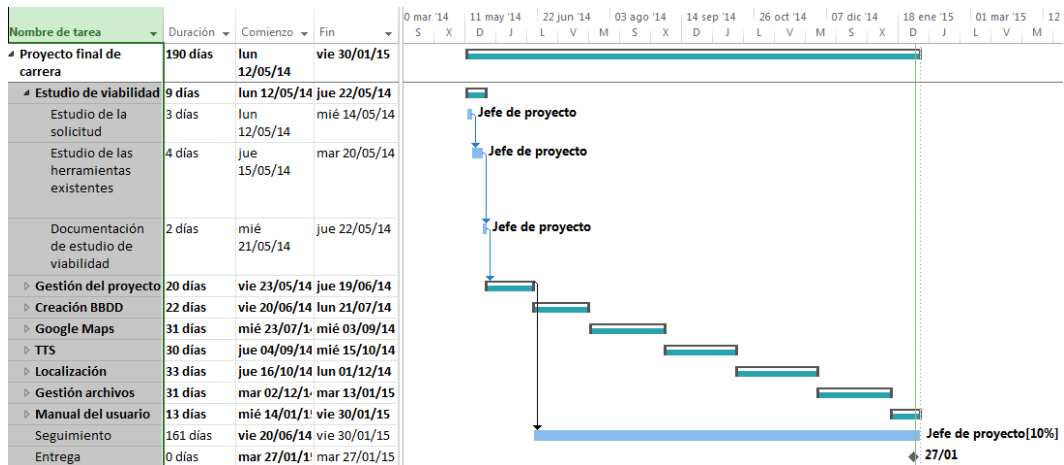


Figura 12. Diagrama de Gantt (Estudio de Viabilidad)

Una vez que el proyecto se ha considerado viable, comienza la fase de gestión del proyecto. De esta fase también se encarga el jefe de proyecto, y en ella debe realizar un estudio sobre los ciclos de vida existentes, seleccionar el más apropiado para el desarrollo del presente proyecto, documentando claramente las razones y motivos de su elección.

A continuación, se realiza el diagrama Gantt en el cual figurarán todas las tareas y actividades necesarias para realizar el desarrollo de la aplicación, así como la duración estimada de cada una de ellas y los recursos necesarios que se utilizarán en cada una de las tareas. En este diagrama las tareas ya aparecen como finalizadas, puesto que ya se ha concluido el proyecto. Por último, el jefe de proyecto ha de realizar el correspondiente documento de Gestión de Proyecto, en el que se incluyan toda la planificación y estimaciones realizadas y comienza el seguimiento de éste, el seguimiento se llevará a cabo en paralelo a todas las actividades a realizar hasta la entrega final de la aplicación.

En la siguiente figura se muestran las tareas que conforman la Gestión de Proyecto:

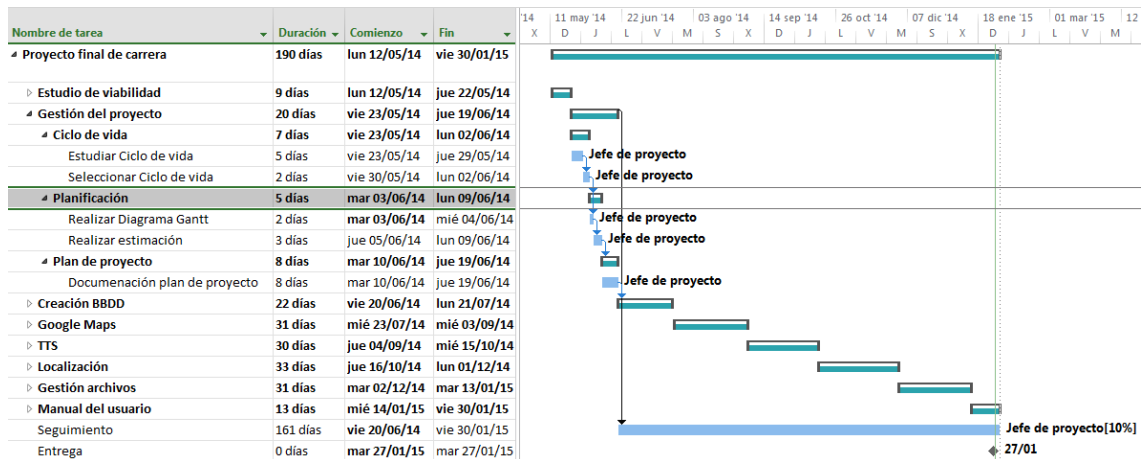


Figura 13. Diagrama de Gantt (Gestión del Proyecto)

Cuando todas las tareas de gestión del proyecto han sido realizadas comienza desarrollo de la primera funcionalidad del sistema, las bases de datos para usuarios y favoritos. En primer lugar el analista debe recopilar los requisitos del usuario para obtener los requisitos software y seleccionar las herramientas necesarias para satisfacerlos. Es necesario que antes crear una funcionalidad, el analista trabaje en conjunto con el cliente o tutor para lograr una correcta identificación de los requerimientos a satisfacer.

El diseñador lee la documentación generada por el analista y diseña la arquitectura del sistema, definiendo para ello tanto las interfaces como los procesos implementar.

A continuación el programador debe implementar el sistema de acuerdo al diseño realizado para satisfacer los requisitos del usuario lo más fielmente posible al diseño realizado.

La primera funcionalidad se concluye con una batería de pruebas, para verificar su correcto funcionamiento y la evaluación por parte del cliente o tutor. Es responsabilidad del cliente trabajar con el prototipo y evaluar sus características y operación. La experiencia del sistema, bajo condiciones reales, permite obtener la familiaridad indispensable para determinar los cambios o mejoras que sean necesarios, así como las características inadecuadas.

Las pruebas serán redactadas por el diseñador y ejecutadas por el programador para comprobar el correcto funcionamiento de la aplicación.



Figura 14. Diagrama de Gantt (Creación de BBDD)

El analista comienza a estudiar los requisitos necesarios para el segundo bloque de funcionalidades, Google Maps tras recibir el visto bueno al primer bloque. Los cambios al primer bloque son planificados con el cliente antes de llevarlos a cabo, aunque es el analista el responsable de tales modificaciones.

Como en el primer bloque de funcionalidades, el analista documenta los nuevos requisitos y selecciona las herramientas necesarias. A continuación el diseñador lee la documentación aportada por el analista y diseña las nuevas interfaces y funcionalidades, las cuales el programador codifica de acuerdo a las especificaciones.

A partir de esta funcionalidad, comienza a darse un importante paso, la integración entre bloques de funcionalidades para que estas funcionalidades se combinen para conseguir el resultado solicitado por el cliente.

Se realizan las pruebas para verificar que todo funciona correctamente y se le muestra al cliente para que evalúe este segundo bloque.

Este mismo bucle se realizará para las funcionalidades tercera y cuarta (TTS y localización respectivamente), cuyas gráficas se muestran a continuación.

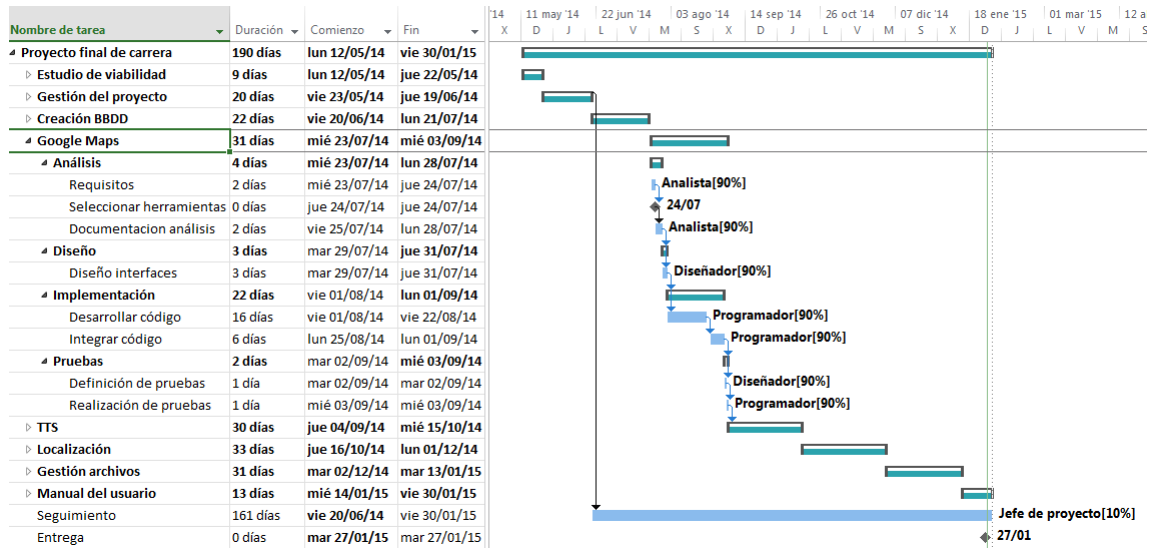


Figura 15. Diagrama de Gantt (Google Maps)

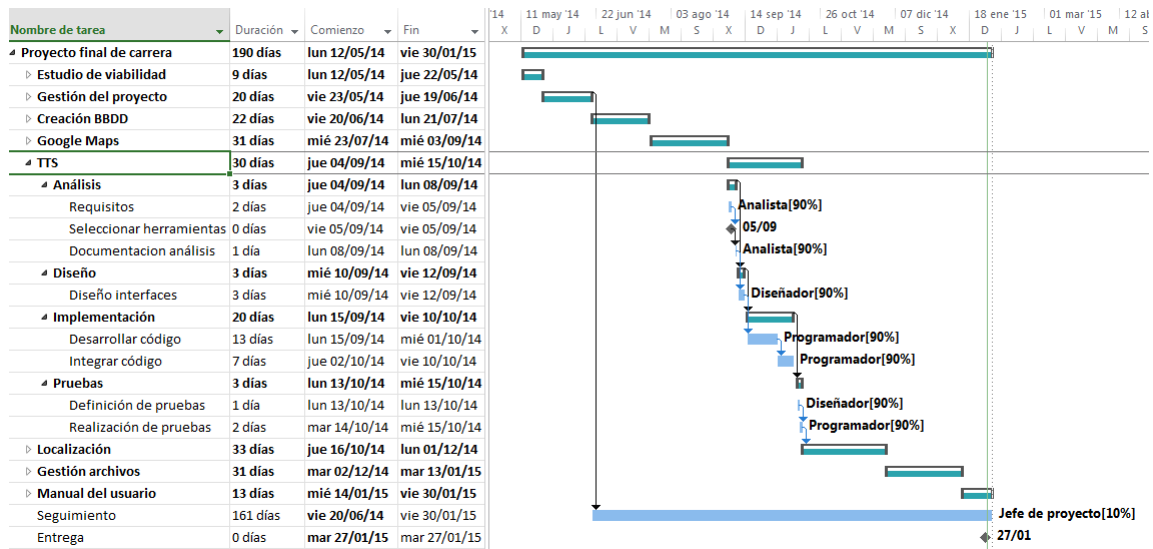


Figura 16. Diagrama de Gantt (TTS)

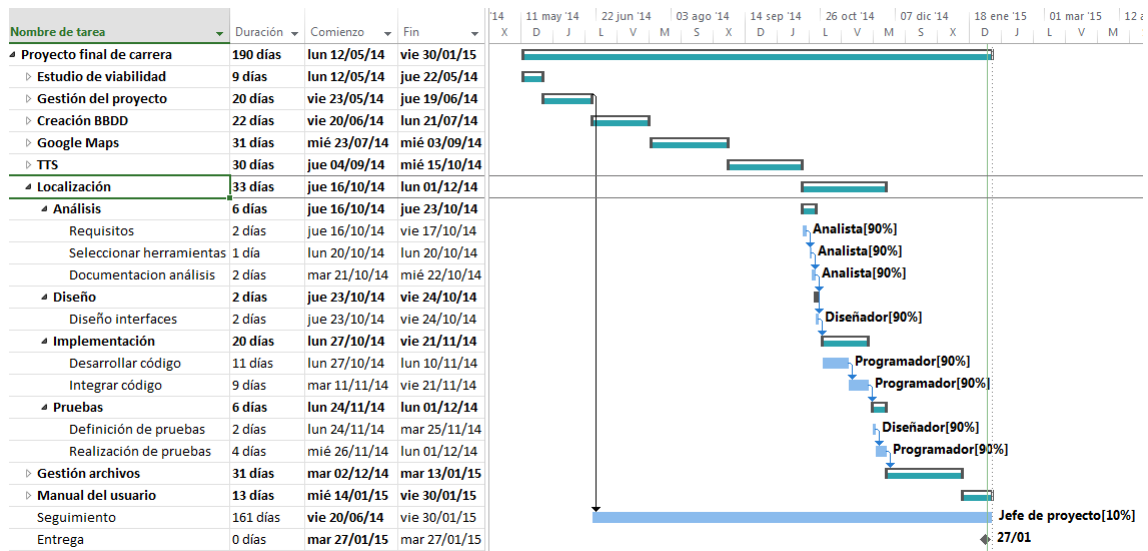


Figura 17. Diagrama de Gantt (Localización)

Finalmente se comienza con la quinta y última funcionalidad (Gestión de archivos) en el que se intentarán satisfacer todos los requisitos expuestos por el cliente a lo largo del proyecto. El analista recoge todos los requisitos y comentarios del usuario y elabora la documentación definitiva, si es necesario utilizar una nueva herramienta de desarrollo la buscará y seleccionará. El diseñador plantea las interfaces y funcionalidades que sean necesarias y el programador implementa la aplicación definitiva.

Una vez realizado esto, se realiza la integración definitiva de todas las funcionalidades, creando así la aplicación requerida al completo.

La fase de pruebas es crucial para determinar que los requerimientos han sido satisfechos y que el sistema se comporta como se esperaba. El prototipo debe pasar las pruebas de funcionalidad para validar su correcto funcionamiento y la evaluación por parte del cliente, siendo éste el que certifique la aceptación del sistema que utiliza.

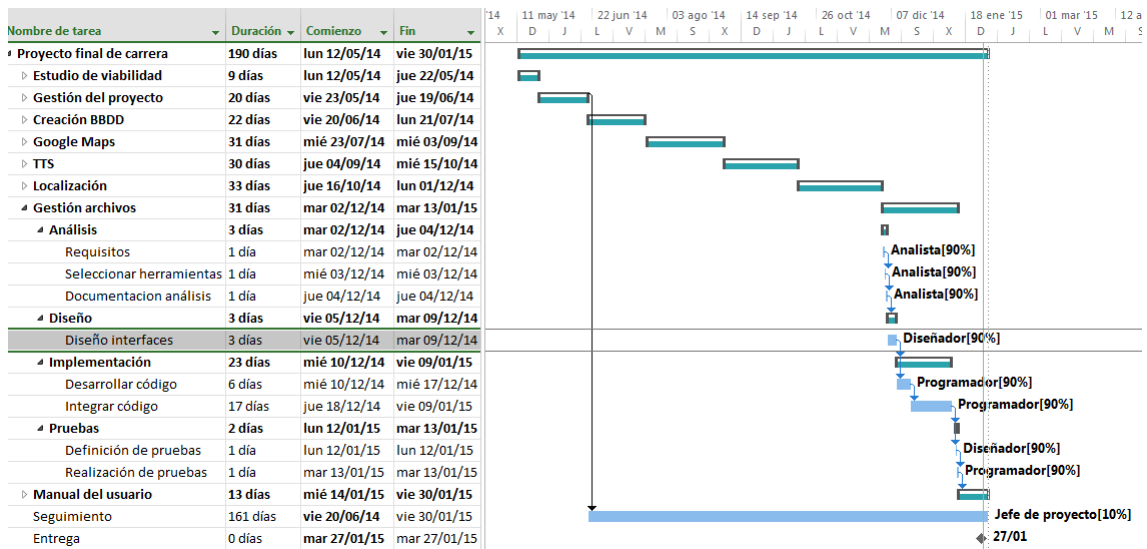


Figura 18. Diagrama de Gantt (Gestión archivos)

Finalmente se genera un manual de usuario indicando todos los aspectos de la aplicación, tanto sus funciones como la manera de ejecutarlas o seleccionarlas, para que el usuario final u otra persona sea capaz de utilizar la nueva aplicación sin problemas, y no tener que recurrir a los desarrolladores.

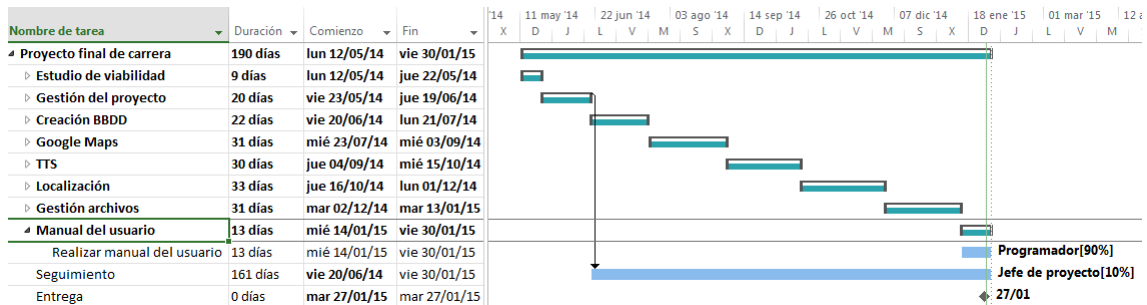


Figura 19. Diagrama de Gantt (Manual de Usuario)

El último hito del proyecto es la entrega de la documentación y el software desarrollado al cliente. Tras este paso se da por concluido el proyecto el día 30 de Enero de 2015.

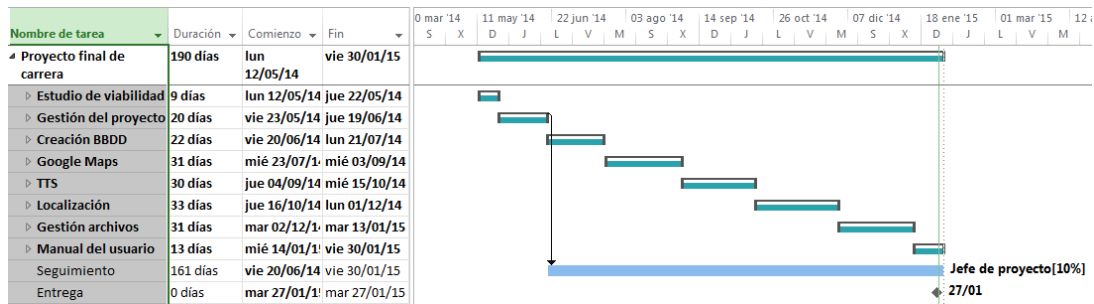


Figura 20. Diagrama de Gantt (Entrega)

4.3 Estimación de costes

Para calcular el coste estimado del proyecto se han tenido en cuenta tanto los costes materiales como los costes humanos del mismo.

En primer lugar se calcula el coste asociado a las herramientas utilizadas para el desarrollo del proyecto:

Herramienta	Coste
Microsoft Windows 7 Home Premium	99,99
Microsoft Office 2007 Professional	255,35
Eclipse Juno 4.2	Licencia gratuita
Android SDK	Licencia gratuita
Total (IVA incluido)	335,34 €

Tabla 24. Coste Licencias

En cuanto al coste de recursos materiales se ha contabilizado, además, la cantidad de 950€ en cuanto a equipamiento informático (Smartphone LG G3 con un coste de 400 euros y un ordenador portátil Asus con coste 550 euros).

Para el cálculo de los costes asociados a los recursos humanos se ha tomado una jornada laboral de 8 horas, distribuidas en horario de 9:00 a 14:00 y 16:00 a 19:00. El proyecto ha sido desarrollado por un solo ingeniero, el cual ha adoptado los distintos roles anteriormente descritos. La siguiente tabla muestra una relación entre las funciones que se han adoptado, el tiempo invertido en cada una de ellas y su coste:

Recurso	Horas	Precio/Hora	Total (euros)
Jefe de proyecto	360,8	45 €	16236
Analista	136,8	35 €	4788
Diseñador	129,6	35 €	4536
Programador	878,4	25 €	21960
Total	1505,6		47520 €

Tabla 25. Coste recursos humanos

Analizando la tabla anterior se puede observar que el coste total resultante de las horas dedicadas por el desarrollador asciende a 47.520€.

Por lo tanto, el coste total de desarrollo del proyecto es la suma de los costes materiales, más los costes humanos:

Recurso	Coste
Coste recursos materiales	1285,34
Coste recursos humanos	47520
Total	48805,34
Total (IVA Incluido)	54.053 €

Tabla 26. Coste total del proyecto

El coste total de desarrollo del proyecto asciende a la cantidad de 54.053 € (cincuenta y cuatro mil cincuenta y tres euros).

Este coste está asociado solo a recursos humanos y recursos materiales, debido a que la aplicación no tiene un fin comercial no se han realizado estudios de cara a las posibles ganancias, márgenes de riesgo y costes indirectos derivados del mantenimiento y actualización de la aplicación de cara al futuro.

5. ANÁLISIS DEL SISTEMA

5.1. Introducción

El objetivo del Análisis del Sistema es la obtención de una colección completa de requisitos del sistema. Esta obtención se realiza a partir de los requisitos obtenidos en el anterior documento: Estudio de Viabilidad del Sistema (EVS).

Además, este documento será la base de la siguiente fase, el Diseño del Sistema. Durante dicha fase se producirá el Documento de Diseño del Sistema (DDS) en el que se especificará el diseño completo del sistema de información.

5.2. Definición del sistema

La aplicación a desarrollar será una herramienta para la gestión de encuestas, esta ha de cubrir las siguientes funcionalidades:

- Bases de datos
- Google Maps
- TTS
- Localización
- Gestión archivos

5.2.1 Bases de datos

La aplicación debe ser capaz de generar y manejar bases de datos. Se necesitan una base para la gestión de usuarios y otra para la de favoritos.

En el primer caso, la información mínima que se necesita es nombre, clave y e-mail (el sistema añadirá un identificador numérico por su parte).

En el segundo, un usuario, inicio, destino, modo y nombre de ruta, y al igual que en la base de usuarios, el sistema añadirá un identificador numérico.

5.2.2. Google Maps

La aplicación utiliza la tecnología de Google Maps para mostrar los mapas de seguimiento de las rutas, así como para poder trazar las rutas sobre las mismas.

También se utilizan las rutas que diseña la herramienta para la realización en nuestra aplicación.

5.2.3. TTS

La parte fundamental de este proyecto es la capacidad de traducción e interpretación de la voz humana para, a partir de este parámetro, desarrollar diversas instrucciones.

El servicio TTS de Google nos permite tanto realizar la traducción voz-texto como la contraria, texto-voz.

5.2.4. Localización

Para el seguimiento de rutas, necesitamos utilizar los servicios de localización de los que disponen los dispositivos Android.

Dicha localización se puede utilizar mediante dos métodos, el GPS interno del dispositivo o mediante triangulación de la posición del dispositivo a través de la red. Este segundo método será el preferido siempre que se disponga de conexión de datos.

5.2.5. Gestión de archivos

La aplicación trabaja con varios archivos de almacenamiento, bases de datos SQL como se indica en el apartado 5.2.1 y, por otro lado, archivos con extensión “.txt” y “.xml” de los cuales se debe manejar su creación, interpretación, edición y eliminación.

5.3. Identificación de subsistemas de análisis

El objetivo de esta actividad es facilitar el análisis del sistema de información llevando a cabo la descomposición del sistema en subsistemas, con el fin de abordar el desarrollo de la forma más sencilla y organizada.

5.3.1. Determinación de los subsistemas de análisis

La definición del diagrama de subsistemas se ha realizado siguiendo una separación funcional, es decir, cada subsistema de análisis identificará un conjunto de funcionalidades ofrecidas por el sistema.

En este caso, la separación de subsistemas atiende a la separación por interfaces, trabajando todo lo relacionado en una interfaz en una o varias clases, pero sin mezclar las mismas.

5.4. Establecimiento de requisitos

En este punto se llevará a cabo la definición, análisis y validación de los requisitos a partir de la información facilitada por el usuario, completándose el catálogo de requisitos obtenido en el punto 3.3. Definición de Requisitos del Sistema.

El objetivo fundamental es conseguir un catálogo detallado de los requisitos, a partir del cual se pueda comprobar que los productos generados en las actividades de modelización se ajustan a los requisitos de usuario.

5.4.1. Obtención de requisitos

A continuación se presentan los requisitos de software identificados en el análisis. El origen de estos requisitos se encuentra en los requisitos de usuario definidos en el Estudio de Viabilidad del Sistema, junto con decisiones tomadas por el equipo de desarrollo.

Los tipos de Requisitos de Software identificados en el proceso han sido los siguientes:

- Requisitos funcionales (F): Son aquellos requisitos que especifican el propósito del sistema, los cuales derivan directamente de los casos de uso.
- Requisitos de interfaz (I): Especifican hardware y/o software con los que debe interactuar el sistema.

- Requisitos de operación (O): Son todos aquellos requisitos que especifican como se va a resolver el problema.
- Requisitos de documentación (D): Requisitos que especifican criterios de realización de la documentación asociada al proyecto.
- Requisitos de calidad (C): Estos requisitos especifican los atributos del software que aseguran que serán adecuados para su propósito.

Cada requisito software debe incluir en su definición, además de la descripción del mismo, una serie de atributos, los cuales proporcionen toda la información necesaria para su seguimiento posterior y su clasificación. Estos atributos se describen a continuación:

- Identificador: Cada requisito debe estar identificado de forma única. Este identificador tendrá el siguiente formato: RS-XX-nnn o RS-XX-nnn, donde:

- RS: Indica que se trata de un requisito software.
- XX: Indicará el tipo de requisitos software al que corresponde (ver listado anterior).
- nnn: Tomará valores numéricos dentro del rango 000-999.

- Prioridad: Se asignará una prioridad a cada requisito con el fin de poder realizar una planificación correcta durante fases posteriores. Su clasificación puede tomar los valores: alta, media y baja.

- Necesidad: Los requisitos clasificados como esenciales no pueden ser eliminados, mientras que los demás requisitos estarán siempre sujetos a modificación en el caso de que exista una causa que lo justifique. Por lo tanto, se clasificarán descendientemente de acuerdo a su necesidad de la siguiente forma: esencial, deseable y opcional.

- Claridad: Identifica la falta o existencia de ambigüedad de un requisito, esto es, si puede ser interpretado de varias formas dependiendo del contexto. Los requisitos serán clasificados de acuerdo a su claridad de forma descendente de la siguiente forma: alta, media y baja.

- Fuente: Identifica el origen del requisito, que puede estar en un requisito de usuario, o en el propio equipo de desarrollo del proyecto.

- Estabilidad: Algunos requisitos pueden no estar sujetos a cambios durante el proyecto debido a su naturaleza, mientras que otros pueden estar sujetos a determinados cambios por el desarrollo de la etapa de diseño. Los requisitos poco estables deberán ser observados con mayor cuidado durante el desarrollo del proyecto al ser susceptibles a cambios. Cada requisito se clasificará en: estable o inestable.

- Verificabilidad: Indica si el cumplimiento de un requisito en el sistema puede ser susceptible de comprobación, esto es, si se puede verificar que el requisito se ha incorporado en el diseño y que en el sistema se puede verificar su cumplimiento. La verificabilidad de cada requisito se clasificará en: alta, media y baja.

5.4.1.1 Requisitos funcionales

Identificador	RS-F-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-001
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Crear usuarios para el uso de la aplicación		

Tabla 27. Requisito de funcionalidad RS-F-001

Identificador	RS-F-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-001
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Crear rutas de favoritos para usarlas a posteriori		

Tabla 28. Requisito de funcionalidad RS-F-002

Identificador	RS-F-003		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-002
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Interpretación y detección de radares desde una base de datos		

Tabla 29. Requisito de funcionalidad RS-F-003

Identificador	RS-F-004		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-003
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Interpretar coordenadas provisionadas por el dispositivo en tiempo real		

Tabla 30. Requisito de funcionalidad RS-F-004

Identificador	RS-F-005		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-004
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Dibujar trazos y ubicar marcadores en mapas.		

Tabla 31. Requisito de funcionalidad RS-F-005

Identificador	RS-F-006		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-004
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Actualización de mapas según movimientos del usuario/dispositivo		

Tabla 32. Requisito de funcionalidad RS-F-006

Identificador	RS-F-007		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-005
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Entender diversas instrucciones orales para su posterior aplicación.		

Tabla 33. Requisito de funcionalidad RS-F-007

Identificador	RS-F-008		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-006
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Ubicar en la memoria del dispositivo los archivos creados/descargados		

Tabla 34. Requisito de funcionalidad RS-F-008

Identificador	RS-F-009		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-009
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Eliminar registros borrando toda la información referente a ellos del sistema.		

Tabla 35. Requisito de funcionalidad RS-F-009

Identificador	RS-F-010		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-010
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Apertura y escritura de ficheros en formato texto plano (.txt)		

Tabla 36. Requisito de funcionalidad RS-F-010

Identificador	RS-F-011		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-011
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Interpretación de algoritmos alfanuméricos para dibujar rutas sobre mapas.		

Tabla 37. Requisito de funcionalidad RS-F-011

Identificador	RS-F-012		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-013
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Cierre de procesos internos finalizados para limitar la memoria empleada por la aplicación.		

Tabla 38. Requisito de funcionalidad RS-F-012

Identificador	RS-F-013		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-014
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Activar servicios de ubicación a deseo del usuario		

Tabla 39. Requisito de funcionalidad RS-F-013

Identificador	RS-F-014		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-015
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Medir distancias recorridas y velocidades medias del usuario en modo libre.		

Tabla 40. Requisito de funcionalidad RS-F-014

Identificador	RS-F-015		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-001
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Medir distancias recorridas y velocidades medias del usuario en modo libre.		

Tabla 41. Requisito de funcionalidad RS-F-015

5.4.1.2 Requisitos de interfaz

Identificador	RS-I-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Disponer de un Smartphone con la funcionalidad de GPS		

Tabla 42. Requisito software RS-I-001

5.4.1.3 Requisitos de operación

Identificador	RS-O-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-002
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Disponer de una interfaz sencilla e intuitiva que permita configurar todos los parámetros de experimentación de forma rápida para el usuario.		

Tabla 43. Requisito software RS-O-001

Identificador	RS-O-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-003
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Señalizar cuando la aplicación está ocupada realizando un trabajo mediante el uso de una barra que muestre el progreso		

Tabla 44. Requisito software RS-O-002

Identificador	RS-O-003		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-004
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Almacenar los datos referentes a cada usuario en formato SQLite. Este proceso debe ser transparente al usuario, es decir, el usuario simplemente pulsará un botón “Guardar configuración” y la aplicación se encargará de todo el proceso.		

Tabla 45. Requisito software RS-O-003

Identificador	RS-O-004		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-005
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Desarrollar la aplicación en lenguaje Java Android mediante el entorno de desarrollo Eclipse Juno 4.2.		

Tabla 46. Requisito software RS-O-004

Identificador	RS-O-005		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-006
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Conectar con los servicios de Google para la navegación del usuario.		

Tabla 47. Requisito software RS-O-005

Identificador	RS-O-006		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-007
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Desarrollar la aplicación para software Android 4.4.2 y superiores.		

Tabla 48. Requisito software RS-O-006

5.4.1.4 Requisitos de documentación

Identificador	RS-O-006		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo de desarrollo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Proporcionar al cliente un manual de usuario online, en el que se incluya una descripción detallada de las principales funcionalidades del sistema. El idioma del manual es castellano.		

Tabla 49. Requisito software RS-D-001

5.4.1.5 Requisitos de calidad

Identificador	RS-C-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo de desarrollo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Comprobar la adecuación de todos los documentos entregados a lo largo del proyecto al estándar establecido para este desarrollo software.		

Tabla 50. Requisito calidad RS-C-001

5.4.2. Especificación de los casos de uso

Con frecuencia los requisitos se expresan de una forma natural como una interacción entre la aplicación y una agencia externa a ella como el caso de un usuario. Los casos de uso son una forma muy útil de representar esas interacciones.

A continuación se especificarán los actores del sistema y después el catálogo de casos de uso y, finalmente, se describirá en detalle cada caso de uso.

5.4.2.1. Actores del sistema

En la aplicación se detectan dos actores que intervienen en el sistema, estos son:

- **Administrador:** Aporta toda la logística interna para que el usuario pueda usar la aplicación.
- **Usuario:** Puede gestionar todas las interfaces de la aplicación y realizar las gestiones deseadas.

5.4.2.2. Listado de casos de uso

A continuación se muestra una lista de todos los casos de uso del sistema. Los casos son agrupados por su funcionalidad y de acuerdo a las divisiones del catálogo de requisitos del sistema.

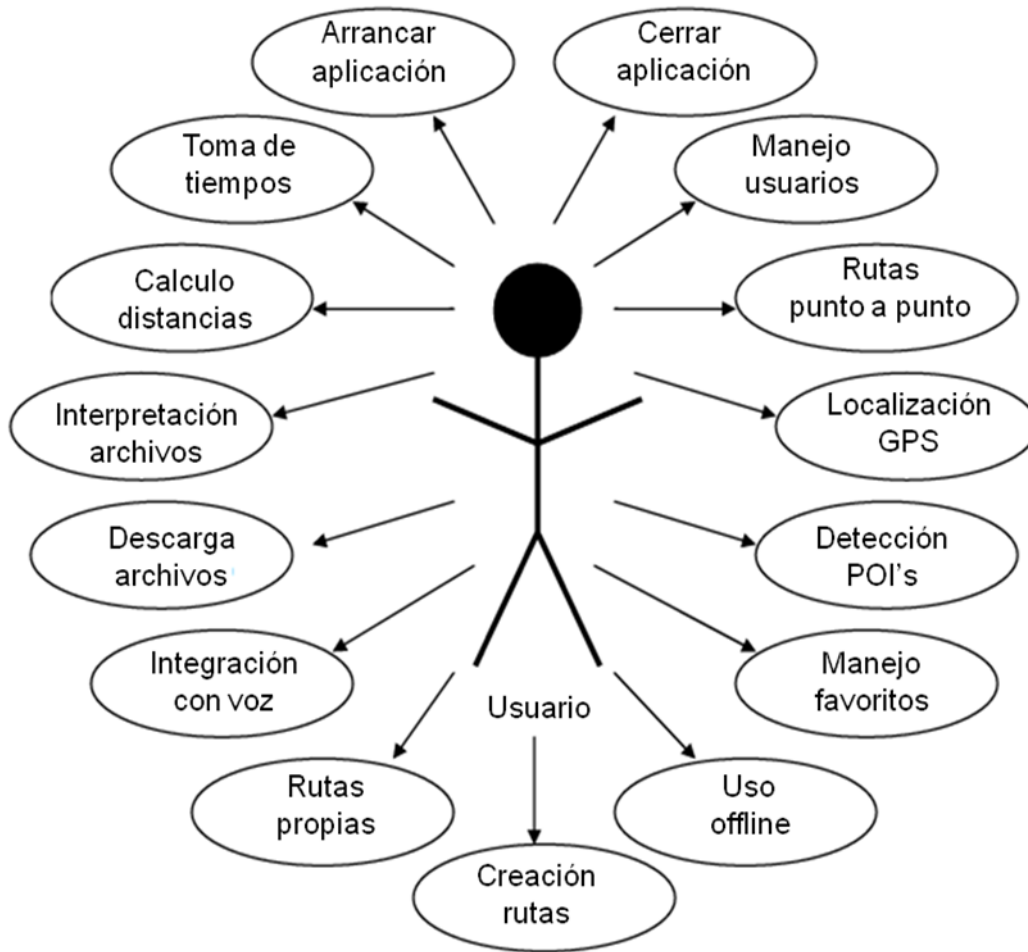


Tabla 51. Casos de uso

Cada caso de uso estará especificado por los siguientes atributos:

- **Identificador:** Identifica al caso de uso de forma única. Debe seguir el formato: CU-XXX, siendo XXX un valor numérico único para cada caso de uso.
- **Nombre:** Breve especificación textual del caso de uso.
- **Actores:** Tipo de usuario del sistema que inicia el caso de uso.
- **Objetivo:** Finalidad del caso de uso.
- **Precondiciones:** Estado previo que se debe cumplir para poder realizar una operación.
- **Postcondiciones:** Estado en el que queda el sistema tras realizar una operación.
- **Escenario básico:** Especifica la manera en la que interactúa un actor con el sistema y cuál es la respuesta que el sistema le ofrece.
- **Escenarios alternativos:** Condiciones excepcionales que afectan al escenario y respuestas del sistema ante esas situaciones.

Identificador	CU-001
Nombre	Arrancar aplicación.
Actores	Usuario.
Objetivo	Inicia la aplicación de manera correcta
Precondiciones	Ninguna
Poscondiciones	Aplicación arrancada. Interfaz gráfica mostrada
Escenario básico	El usuario inicia la ejecución de la aplicación. La aplicación muestra la interfaz de tratamiento de usuario.
Escenarios alternativos	

Tabla 52. Caso de uso CU-001

Identificador	CU-002
Nombre	Cerrar aplicación
Actores	Usuario.
Objetivo	Cerrar la aplicación de manera correcta
Precondiciones	Aplicación abierta
Poscondiciones	
Escenario básico	Encontrarse en cualquier escenario de la aplicación
Escenarios alternativos	Cerrar desde fuera de la aplicación (Uso de task killer o servicio propio Android)

Tabla 53. Caso de uso CU-002

Identificador	CU-003
Nombre	Manejo usuarios
Actores	Usuario.
Objetivo	Crear, modificar, eliminar usuarios
Precondiciones	Aplicación abierta
Poscondiciones	Posibilidad de acceder al resto de la aplicación
Escenario básico	Encontrarse en la pantalla inicial de la aplicación
Escenarios alternativos	

Tabla 54. Caso de uso CU-003

Identificador	CU-004
Nombre	Rutas punto a punto
Actores	Usuario, dispositivo
Objetivo	Crear rutas mediante voz
Precondiciones	Autenticación de usuario
Poscondiciones	Acceso mapa para realizar la ruta solicitada
Escenario básico	Encontrarse en la pantalla principal de la aplicación
Escenarios alternativos	

Tabla 55. Caso de uso CU-004

Identificador	CU-005
Nombre	Localización GPS
Actores	Dispositivo
Objetivo	Conocer la posición actual del dispositivo/usuario
Precondiciones	Creación de nueva ruta o intento de descarga desde Google Maps. Ubicación activa en el dispositivo.
Poscondiciones	Acceso mapa para realizar la ruta solicitada ó inicio de creación de ruta
Escenario básico	Encontrarse en la pantalla principal de la aplicación, hacer click sobre el micrófono y usar los comandos "Modo libre", "Ir a..." o "Ir de..."
Escenarios alternativos	

Tabla 56. Caso de uso CU-005

Identificador	CU-006
Nombre	Detección POI's
Actores	Dispositivo
Objetivo	Dar instrucciones al usuario o avisar sobre radares
Precondiciones	Estar realizando una ruta descargada desde Google Maps con ubicación activa en el dispositivo
Poscondiciones	Realizar el aviso sugerido por el dispositivo
Escenario básico	Acercarse a determinada distancia en línea recta a un radar o un punto de cambio de dirección en nuestra ruta (giro, rotonda, acceso a nueva vía)
Escenarios alternativos	

Tabla 57. Caso de uso CU-006

Identificador	CU-007
Nombre	Manejo favoritos
Actores	Usuario.
Objetivo	Crear y eliminar favoritos
Precondiciones	Autenticación del usuario
Poscondiciones	Posibilidad de cargar la ruta cuando se desee
Escenario básico	Acceder desde el menú principal haciendo click en el botón
Escenarios alternativos	Acceder haciendo click sobre el micrófono y usando el comando de voz "Favoritos" en menú principal

Tabla 58. Caso de uso CU-007

Identificador	CU-008
Nombre	Uso offline
Actores	Dispositivo
Objetivo	Usar ubicación por GPS en zonas sin red
Precondiciones	Estar en una zona sin red móvil y tener almacenada una ruta descargada desde Google Maps
Poscondiciones	Realizar la ruta a partir de la ubicación GPS del dispositivo (menos precisa que con red)
Escenario básico	Descargar una ruta en una zona con red conocida y realizarla en una zona sin ella.
Escenarios alternativos	Acceder a una zona sin red durante la realización de una ruta descargada en una zona con red.

Tabla 59. Caso de uso CU-008

Identificador	CU-009
Nombre	Creación rutas
Actores	Usuario y dispositivo
Objetivo	Elegir una ruta desde la posición actual o una a elección
Precondiciones	Autenticación de usuario, ubicación activa
Poscondiciones	Iniciar ruta importada de Google Maps.
Escenario básico	Acceder haciendo click sobre el micrófono y usando los comando de voz “Ir a...” (ubicación actual) ó “Ir de...” (ubicación a elección) en menú principal
Escenarios alternativos	Seleccionar una ruta ya descargada desde favoritos.

Tabla 60. Caso de uso CU-009

Identificador	CU-010
Nombre	Rutas propias
Actores	Usuario y dispositivo
Objetivo	Crear rutas libres para repetir a posteriori
Precondiciones	Acceder al modo libre, tener ubicación activa
Poscondiciones	Posibilidad de cargar la ruta cuando se desee para mejorar tú tiempos anteriores.
Escenario básico	Acceder haciendo click sobre el micrófono y usando el comando de voz “Modo libre” en menú principal. Hacer click sobre “Empezar” y comenzar el recorrido deseado. El dispositivo tomará una captura de la posición cada segundo.
Escenarios alternativos	

Tabla 61. Caso de uso CU-010

Identificador	CU-011
Nombre	Integración con voz
Actores	Usuario y dispositivo.
Objetivo	Comunicación con el dispositivo para realizar la acción deseada
Precondiciones	Ubicación activa, volumen activo, autenticación del usuario.
Poscondiciones	Desde acceder a favoritos, a crear rutas libres o desde Google Maps, llamar o mandar un SMS.
Escenario básico	Hacer click en el micrófono del menú principal.
Escenarios alternativos	Acceder haciendo click en el botón "Añadir nuevo favorito" de la pantalla Favoritos.

Tabla 62. Caso de uso CU-011

Identificador	CU-012
Nombre	Descarga archivos
Actores	Dispositivo
Objetivo	Descargar rutas desde Google Maps
Precondiciones	Estar en una zona con red móvil, ubicación activa, usuario autenticado.
Poscondiciones	Iniciar ruta o almacenar la misma
Escenario básico	Acceder haciendo click sobre el micrófono y usando los comando de voz "Ir a..." (ubicación actual) ó "Ir de..." (ubicación a elección) en menú principal
Escenarios alternativos	Acceder haciendo click en el botón "Añadir nuevo favorito" de la pantalla Favoritos.

Tabla 63. Caso de uso CU-012

Identificador	CU-013
Nombre	Interpretación archivos
Actores	Dispositivo
Objetivo	Leer información desde archivos .xml, .sqlite o .txt
Precondiciones	Existencia del archivo
Poscondiciones	Usar la información tomada para cargar un usuario/favorito, una ruta o la base de radares
Escenario básico	Autenticarse como usuario en la aplicación, por lo que el dispositivo accederá a comparar los datos con la BBDD de usuarios.
Escenarios alternativos	Creación de ruta prediseñada en Google Maps, ruta libre, o bien, cargar la base de datos de radares.

Tabla 64. Caso de uso CU-013

Identificador	CU-014
Nombre	Cálculo distancias
Actores	Dispositivo
Objetivo	Comparar la ubicación del usuario con la de un POI
Precondiciones	Ruta de Google Maps cargada.
Poscondiciones	Conocer la distancia en línea recta a un punto en concreto
Escenario básico	Detectar un punto concreto cercano y avisar al dispositivo para recibir un aviso.
Escenarios alternativos	

Tabla 65. Caso de uso CU-014

Identificador	CU-015
Nombre	Toma de tiempos
Actores	Dispositivo
Objetivo	Tomar el tiempo en realizar una ruta
Precondiciones	Comenzar una ruta libre, ubicación activa
Poscondiciones	Conocer el tiempo en realizar la misma o activar una cuenta atrás con dicho tiempo.
Escenario básico	Creación de una ruta libre.
Escenarios alternativos	Tratar de batir tu mejor tiempo en una ruta libre.

Tabla 66. Caso de uso CU-015

5.5 Análisis de los casos de uso

En este apartado se realiza un estudio de los casos de uso identificados en el Estudio de Viabilidad del Sistema, con el fin de identificar las clases necesarias para el sistema, así como las relaciones entre ellas y su asociación con los casos de uso.

El objetivo es identificar el mínimo número de clases que serán necesarias para el sistema, pero suficientes para cubrir todos los requisitos especificados.

5.5.1 Identificación de clases asociadas a un caso de uso

En los siguientes apartados se relacionan los casos de uso de cada uno de los módulos del sistema con las clases asociadas a cada uno de ellos.

Identificador	Clases Asociadas
CU-001	Portada
CU-002	
CU-003	Login Usuarios UsuariosCursorAdapter UsuariosDbAdapter UsuariosFormulario Mail
CU-004	AndroidXMLParsingActivity

	CalcularDistanciaPuntos GPSTracker MainActivity MostrarMapa RutasHelper
CU-005	GPSTracker
CU-006	CalcularDistanciaPuntos GPSTracker MostrarMapa
CU-007	Favoritos FavoritosCursorAdapter FavoritosDbHelper FavoritosDbAdapter FavoritosDescargaArchivo
CU-008	Favoritos MostrarMapa GPSTracker
CU-009	ModoLibre FinDeLaRuta GPSTracker
CU-010	ModoLibre FinDeLaRuta GPSTracker
CU-011	MainActivity Favoritos
CU-012	Favoritos MostrarMapa
CU-013	Favoritos Usuarios ModoLibre MostrarMapa
CU-014	CalculoDistanciaPuntos
CU-015	ModoLibre

Tabla 67. Clases asociadas a Casos de uso

5.5.2 Descripción de la interacción de objetos

A continuación se representa la interacción descrita en el apartado anterior entre los casos de uso y las clases.

Se han utilizado diagramas de secuencia de manera que se pueda observar con una mayor claridad la interacción existente entre las clases que formarán el sistema.

Cabe destacar que en este apartado se muestran los diagramas de secuencia para los casos de uso más relevantes.

A continuación se muestra el diagrama de secuencia para el caso de uso CU-003: Manejo Usuarios.

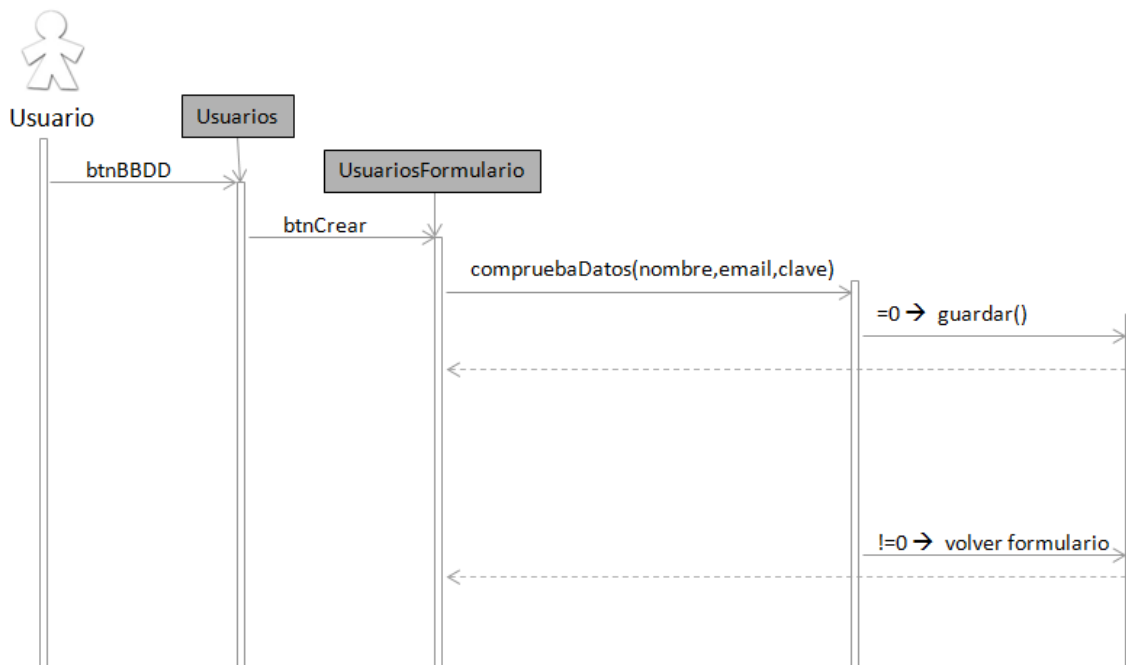


Figura 21. Diagrama de secuencia Caso de Uso CU-003

El siguiente diagrama de secuencia representa el caso de uso CU-009: Creación rutas

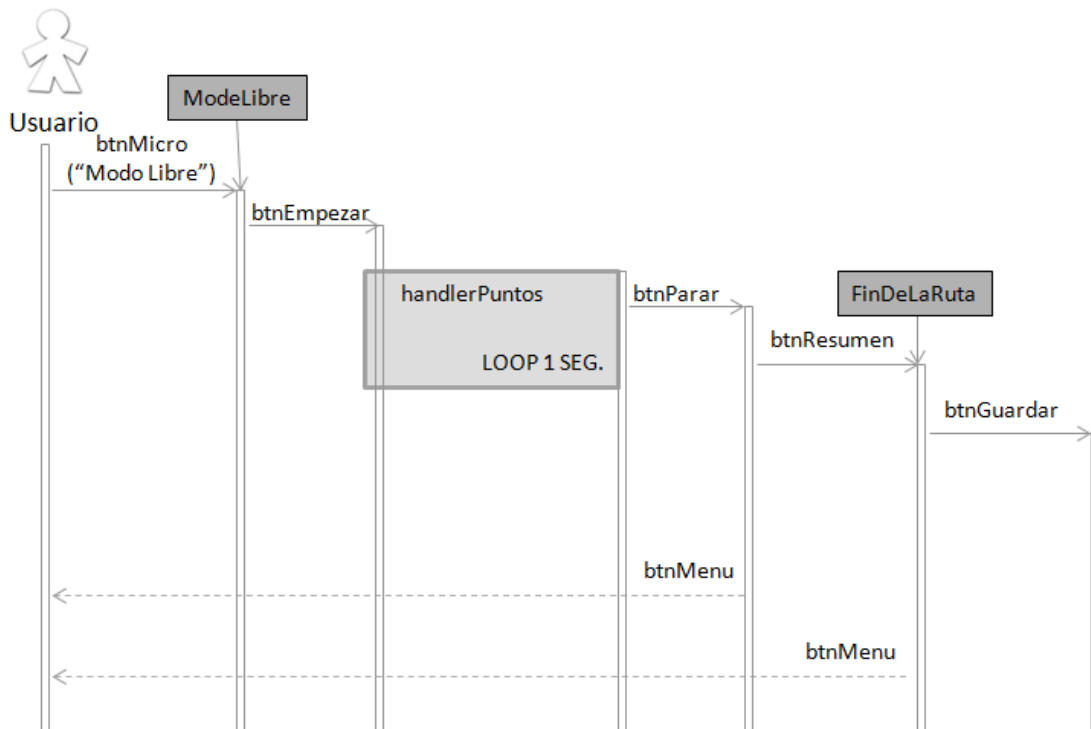


Figura 22. Diagrama de secuencia Caso de Uso CU-009

5.6 Análisis de clases

El objetivo de esta tarea es la descripción de cada una de las clases identificadas en el proceso de análisis. Para ello, se detallarán sus atributos y las relaciones existentes entre ellas.

Es importante destacar, que en esta fase de desarrollo del proyecto, el proceso de análisis, únicamente se realizará un estudio general de los requisitos sin hacer un especial énfasis en cuanto a la toma de decisiones relativas a la implementación. Por lo tanto, el modelo obtenido en este apartado será un simple análisis conceptual de lo que el cliente requiere, una representación a alto nivel del sistema sin ahondar en ningún tipo de detalle relacionado con la implementación. En la siguiente etapa del proyecto, la fase de diseño, se realizará un estudio más exhaustivo.

Además, con el fin de simplificar el diagrama se han incluido las funciones y procedimientos necesarios para la comprensión del sistema, obviando algunas funciones de obtención o modificación de propiedades, que aunque si serán implementadas no se muestran en este modelo.

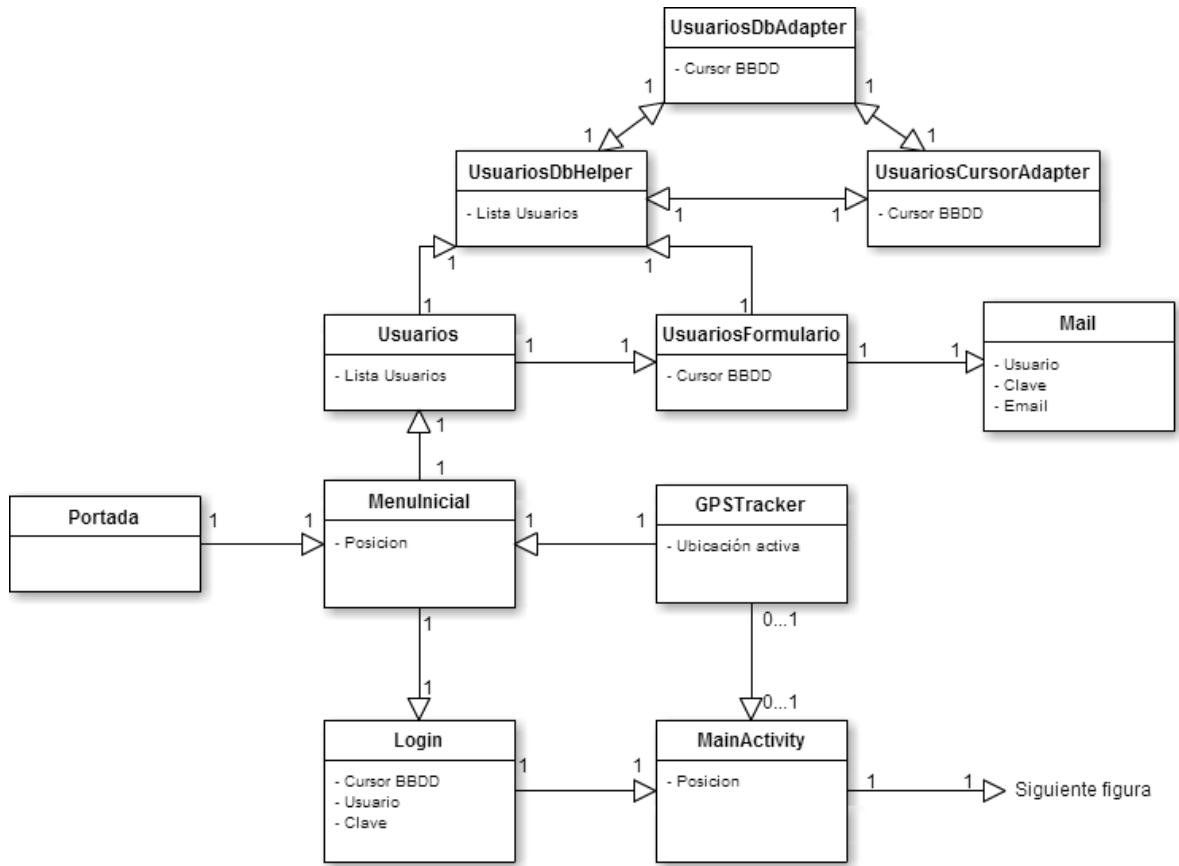


Figura 23. Diagrama conceptual de clases (Parte 1)

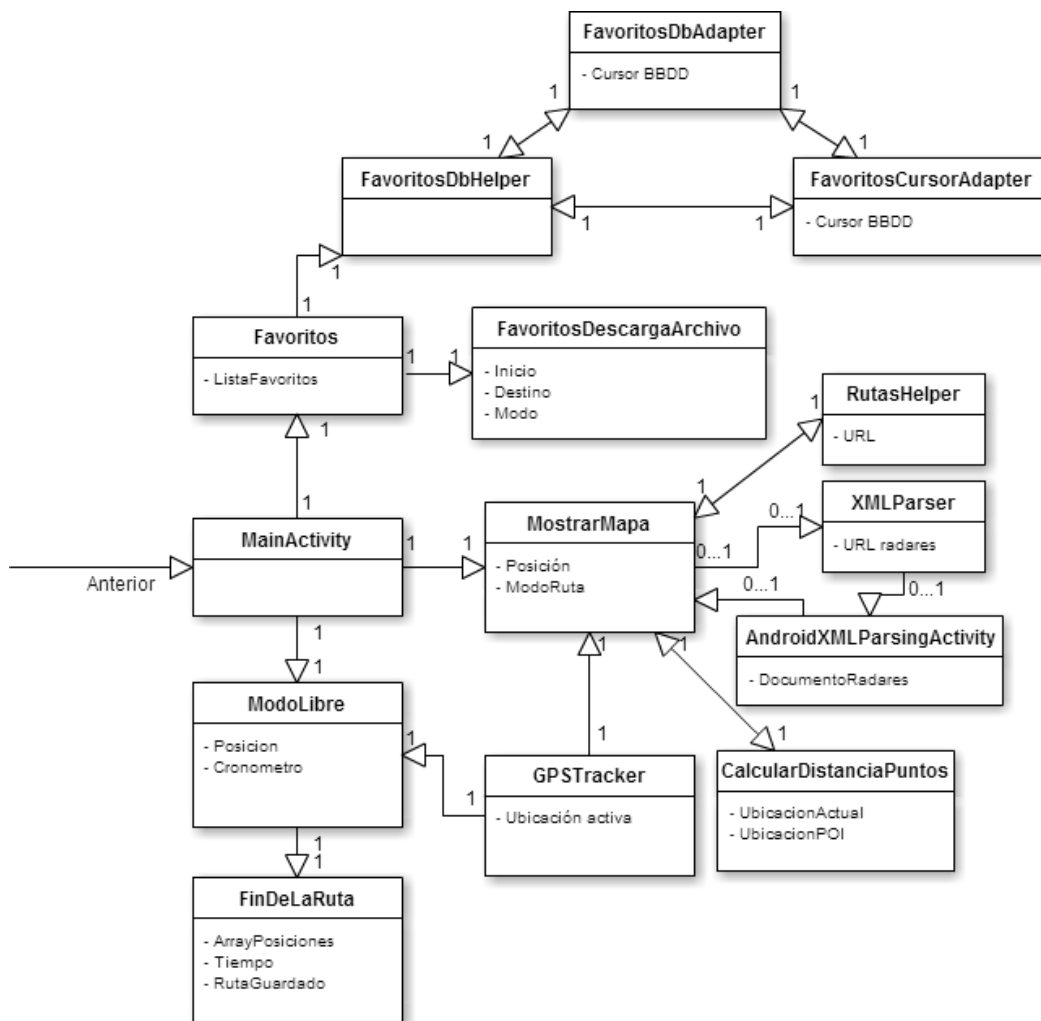


Figura 24. Diagrama conceptual de clases (Parte 2)

5.6.1 Identificación de responsabilidades y atributos

En este apartado se especifican tanto las responsabilidades asociadas a cada una de las clases mostradas en la ilustración anterior. Con ello, se pretende describir el papel que desempeñan los objetos pertenecientes a las clases en los diferentes casos de uso del sistema.

Además, se describen los atributos correspondientes a las clases, los cuales especifican propiedades de las mismas y están involucrados, a su vez, en las diferentes responsabilidades de la clase.

A continuación se describen las responsabilidades y atributos más relevantes de cada una de las clases del diagrama:

Clase	AndroidXMLParsingActivity
Responsabilidades	Interpretar el documento de radares para poder integrarlo durante la ruta.
Atributos	<ul style="list-style-type: none"> - DocumentoRadares Documento que recibe con toda la información sobre los radares fijos de la red de carreteras españolas.

Tabla 68. Clase AndroidXMLParsingActivity

Clase	CalcularDistanciaPuntos
Responsabilidades	Medir la distancia entre el punto actual y cualquier otro punto de interés (radar, punto de la ruta...)
Atributos	<ul style="list-style-type: none"> - UbicacionActual Coordenadas geográficas actuales del dispositivo - UbicacionPOI Ubicación del punto de interés con el cual se quiere medir la distancia.

Tabla 69. Clase CalcularDistanciaPuntos

Clase	Favoritos
Responsabilidades	Gestión de las rutas favoritas del usuario.
Atributos	<ul style="list-style-type: none"> - ListaFavoritos Lista con los nombres de las rutas según el usuario para mostrarlos por pantalla.

Tabla 70. Clase Favoritos

Clase	FavoritosCursorAdapter
Responsabilidades	Proporciona un cursor de acceso a la base de datos para recorrerla según la especificación necesaria.
Atributos	<ul style="list-style-type: none"> - CursorBBDD Cursor específico para recorrer la base a través del nombre de usuario o el id.

Tabla 71. Clase FavoritosCursorAdapter

Clase	FavoritosDbAdapter
-------	--------------------

Responsabilidades	Proporciona los distintos métodos y contantes de la tabla
Atributos	- CursorBBDD Cursor básico para el recorrido de la tabla.

Tabla 72. Clase FavoritosDbAdapter

Clase	FavoritosDbHelper
Responsabilidades	Crea o abre la tabla con las especificaciones definidas previamente por el equipo de desarrollo.
Atributos	

Tabla 73. Clase FavoritosDbHelper

Clase	FavoritosDescargaArchivo
Responsabilidades	Descarga el archivo con la ruta seleccionada por el cliente y su modo.
Atributos	<ul style="list-style-type: none"> - Inicio Dirección inicial de la ruta - Destino Dirección final de la ruta - Modo Manera de realizar la ruta (a pie o en vehículo).

Tabla 74. Clase FavoritosDescargaArchivo

Clase	FinDeLaRuta
Responsabilidades	Presentación resumen de la ruta recién almacenada en el modo libre.
Atributos	<ul style="list-style-type: none"> - ArrayPosiciones Conjunto de posiciones que componen la ruta generada por el usuario - Tiempo Tiempo tardado en realizar la misma. - RutaGuardado Path para almacenar la misma en caso de que el usuario lo solicite.

Tabla 75. Clase FinDeLaRuta

Clase	GPSTracker
-------	------------

Responsabilidades	Servicio en segundo plano que nos proporciona la ubicación cuando se solicita por parte de la aplicación.
Atributos	<ul style="list-style-type: none"> - Ubicación activa Solo se requiere que los servicios de ubicación del dispositivo se encuentren activados.

Tabla 76. Clase GPSTracker

Clase	Login
Responsabilidades	Acceso a la aplicación previa identificación como usuario.
Atributos	<ul style="list-style-type: none"> - CursorBBDD Acceso a la BBDD de usuarios para poder comprobar si los datos son correctos - Usuario Nombre del usuario introducido por pantalla - Clave Clave del usuario introducida por pantalla

Tabla 77. Clase Login

Clase	Mail
Responsabilidades	Envío de email de confirmación con los datos del usuario tras un registro correcto.
Atributos	<ul style="list-style-type: none"> - Email Cuenta de correo indicada por el usuario para el registro. - Usuario Nombre del usuario introducido por pantalla - Clave Clave del usuario introducida por pantalla

Tabla 78. Clase Mail

Clase	MainActivity
Responsabilidades	Clase central de la aplicación desde la cual elegir la acción principal a realizar.
Atributos	<ul style="list-style-type: none"> - Posicion Coordenadas de situación actual del dispositivo

Tabla 79. Clase MainActivity

Clase	MenuInicial
-------	-------------

Responsabilidades	Clase inicial de la aplicación desde la cual elegir si acceder a la clase MainActivity o al manejo de la base de datos de usuarios.
Atributos	- Posicion Coordenadas de situación actual del dispositivo

Tabla 80. Clase MenuInicial

Clase	ModoLibre
Responsabilidades	Clase para generar rutas según la posición del dispositivo del usuario.
Atributos	- Posicion Coordenadas de situación actual del dispositivo - Cronometro Mide el tiempo que el usuario ha tardado en realizar la ruta.

Tabla 81. Clase ModoLibre

Clase	MostrarMapa
Responsabilidades	Clase de seguimiento de la ruta solicitada por el usuario
Atributos	- Posicion Coordenadas de situación actual del dispositivo - ModoRuta Para saber si los radares están activados o no según si el usuario realiza el trayecto en vehículo o a pie.

Tabla 82. Clase MostrarMapa

Clase	Portada
Responsabilidades	Presentación de la aplicación
Atributos	

Tabla 83. Clase Portada

Clase	RutasHelper
-------	-------------

Responsabilidades	Carga, lectura y almacenamiento de los parámetros de la ruta, tanto guardada de manera previa o descargada en el momento
Atributos	<ul style="list-style-type: none"> - Posicion Coordenadas de situación actual del dispositivo - ModoRuta Para saber si los radares están activados o no según si el usuario realiza el trayecto en vehículo o a pie.

Tabla 84. Clase RutasHelper

Clase	Usuarios
Responsabilidades	Muestra por pantalla todos los usuarios para poder editar sus datos, eliminarlos o crear uno nuevo.
Atributos	<ul style="list-style-type: none"> - ListaUsuarios Cursor con los nombres de los usuarios para mostrarlos por pantalla y poder acceder a cada uno.

Tabla 85. Clase Usuarios

Clase	UsuariosCursorAdapter
Responsabilidades	Proporciona un cursor de acceso a la base de datos para recorrerla según la especificación necesaria.
Atributos	<ul style="list-style-type: none"> - CursorBBDD Cursor específico para recorrer la base a través del nombre de usuario o el id.

Tabla 86. Clase UsuariosCursorAdapter

Clase	UsuariosDbAdapter
Responsabilidades	Proporciona los distintos métodos y contantes de la tabla
Atributos	<ul style="list-style-type: none"> - CursorBBDD Cursor básico para el recorrido de la tabla.

Tabla 87. Clase UsuariosDbAdapter

Clase	UsuariosDbHelper
-------	------------------

Responsabilidades	Crea o abre la tabla con las especificaciones definidas previamente por el equipo de desarrollo.
Atributos	

Tabla 88. Clase UsuariosDbHelper

Clase	UsuariosFormulario
Responsabilidades	Crea o abre la tabla con las especificaciones definidas previamente por el equipo de desarrollo.
Atributos	- CursorBBDD Cursor para conocer el siguiente id disponible para un usuario

Tabla 89. Clase UsuariosFormulario

Clase	XMLParser
Responsabilidades	Apertura del documento que almacena la información sobre radares
Atributos	- URL Para abrir el documento de radares

Tabla 90. Clase XMLParser

5.6.2 Identificación de asociaciones

En esta tarea se estudian las relaciones de asociación entre las distintas clases identificadas en el diagrama. Estas relaciones han sido obtenidas partir de la especificación de casos de uso.

Cada una de estas asociaciones será especificada con su cardinalidad, número de instancias que pueden estar relacionadas con una instancia del otro lado de la asociación, y con una breve descripción de su funcionalidad.

A continuación se describen las asociaciones identificadas en los diagramas de clases anteriores:

Clases asociadas	Portada – MenuInicial
Cardinalidades	- Portada (1) - MenuInicial(1)
Descripción	Una interfaz de presentación que da paso al menú inicial de la aplicación.

Tabla 91. Asociación clases Portada y MenuInicial

Clases asociadas	GPSTracker – MenuInicial
------------------	--------------------------

Cardinalidades	- GPSTracker (1) - MenuInicial(0..1)
Descripción	El sistema detecta si la ubicación está activa y es correcta y en caso positivo nos permite acceder al menú principal de la aplicación. En caso negativo, nos obliga a activarla.

Tabla 92. Asociación clases GPSTracker y MenuInicial

Clases asociadas	MenuInicial – Usuarios
Cardinalidades	- MenuInicial (1) - Usuarios(1)
Descripción	Botón de acceso entre la interfaz menú y la de manejo de la base de datos de usuarios.

Tabla 93. Asociación clases MenuInicial y Usuarios

Clases asociadas	Usuarios – UsuariosFormulario
Cardinalidades	- Usuarios (1) - UsuariosFormulario(1)
Descripción	Botón de acceso entre la interfaz menú y la de manejo de la base de datos de usuarios.

Tabla 94. Asociación clases Usuarios y UsuariosFormulario

Clases asociadas	Usuarios – UsuariosFormulario
Cardinalidades	- Usuarios (1) - UsuarioDbHelper(1)
Descripción	Chequea si la base de datos de usuario ya existe y, en caso de que no, la crea.

Tabla 95. Asociación clases Usuarios y UsuariosDbHelper

Clases asociadas	UsuariosDBHelper– UsuariosCursorAdapter - UsuariosDbAdapter
-------------------------	------------------------------------------------------------------------

Cardinalidades	<ul style="list-style-type: none"> - UsuariosDbHelper(1) - UsuariosCursorAdapter(1) - UsuariosDbAdapter(1)
Descripción	Triple relación recíproca entre estas clases conectadas entre sí para dar al usuario el soporte necesario al acceder a la base de datos

Tabla 96. Asociación clases UsuariosDbHelper, UsuariosDbAdapter y UsuariosCursorAdapter

Clases asociadas	UsuariosFormulario – Mail
Cardinalidades	<ul style="list-style-type: none"> - UsuariosFormulario (1) - Mail(1)
Descripción	Clase para el envío de un email con los datos del usuario una vez se haya realizado un registro correcto.

Tabla 97. Asociación clases UsuariosFormulario y Mail

Clases asociadas	MenuInicial – Login
Cardinalidades	<ul style="list-style-type: none"> - MenuInicial (1) - Login(1)
Descripción	Autenticación del usuario para acceder a su zona privada.

Tabla 98. Asociación clases MenuInicial y Login

Clases asociadas	Login – MainActivity
Cardinalidades	<ul style="list-style-type: none"> - Login (1) - MainActivity(1)
Descripción	Acceso al menú principal del área privada tras la interfaz de autenticación del usuario.

Tabla 99. Asociación clases Login y MainActivity

Clases asociadas	MainActivity –GPSTracker
Cardinalidades	<ul style="list-style-type: none"> - MainActivity (0..1) - GPSTracker(0..1)
Descripción	El sistema detecta si la ubicación está activa y es correcta y en caso positivo nos permite solicitar la creación de una ruta. En caso negativo, nos obliga a activarla.

Tabla 100. Asociación clases MainActivity y GPSTracker

Clases asociadas	MainActivity -ModoLibre
-------------------------	--------------------------------

Cardinalidades	- MainActivity (1) - ModoLibre(1)
Descripción	Acceso a la interfaz del modo libre si el comando de voz pronunciado es “modo libre”

Tabla 101. Asociación clases MainActivity y ModoLibre

Clases asociadas	ModoLibre -FinDeLaRuta
Cardinalidades	- ModoLibre (1) - FinDeLaRuta(1)
Descripción	Al finalizar una ruta creada de manera libre, se ofrece al usuario la posibilidad de ver diversos parámetros de la misma, así como el recorrido realizado.

Tabla 102. Asociación clases ModoLibre y FinDeLaRuta

Clases asociadas	ModoLibre -GPSTracker
Cardinalidades	- ModoLibre (1) - GPSTracker(1)
Descripción	El sistema detecta si la ubicación está activa y es correcta y en caso positivo nos permite crear una ruta propia. En caso negativo, nos obliga a activarla.

Tabla 103. Asociación clases ModoLibre y GPSTracker

Clases asociadas	MainActivity -MostrarMapa
Cardinalidades	- MainActivity (1) - MostrarMapa(1)
Descripción	Acceso a la interfaz de seguimiento de ruta si previamente se han introducido unos valores aceptables.

Tabla 104. Asociación clases MainActivity y MostrarMapa

Clases asociadas	MostrarMapa - RutasHelper
-------------------------	----------------------------------

Cardinalidades	- MostrarMapa (1) - RutasHelper(1)
Descripción	Conexión, descarga e interpretación de la ruta solicitada por el usuario para su realización en la interfaz MostrarMapa

Tabla 105. Asociación clases MostrarMapa y RutasHelper

Clases asociadas	MostrarMapa –XMLParser - AndroidXMLParsingActivity
Cardinalidades	- MostrarMapa (1) - XMLParser(1) - AndroidXMLParsingActivity(1)
Descripción	En esta triangulación de clases, el dispositivo solicita la información de radares, la interpreta y la coloca a disposición del usuario para su uso durante la ruta.

Tabla 106. Asociación clases MostrarMapa, XMLParser y AndroidXMLParsingActivity

Clases asociadas	MostrarMapa -CalcularDistanciaPuntos
Cardinalidades	- MostrarMapa (1) - CalcularDistanciaPuntos(1)
Descripción	El dispositivo recibe la ubicación actual del usuario y la compara con todos los POI's cercanos para indicar las siguientes instrucciones.

Tabla 107. Asociación clases MostrarMapa y CalcularDistanciaPuntos

Clases asociadas	MainActivity –Favoritos
Cardinalidades	- MostrarMapa (1) - Favoritos(1)
Descripción	Se accede a la interfaz de favoritos previa pulsación en el botón correspondiente o tras la expresión "favoritos" en el reconocedor de voz.

Tabla 108. Asociación clases MainActivity y Favoritos

Clases asociadas	Favoritos -FavoritosDescargaArchivo
-------------------------	--------------------------------------------

Cardinalidades	<ul style="list-style-type: none"> - Favoritos (1) - FavoritosDescargaArchivo(1)
Descripción	Conexión, descarga y almacenamiento de la ruta solicitada por el usuario para su realización en la interfaz MostrarMapa.

Tabla 109. Asociación clases Favoritos y FavoritosDescargaArchivo

Clases asociadas	FavoritosDBHelper– FavoritosCursorAdapter – FavoritosDbAdapter
Cardinalidades	<ul style="list-style-type: none"> - FavoritosDbHelper(1) - FavoritosCursorAdapter(1) - FavoritosDbAdapter(1)
Descripción	Triple relación recíproca entre estas clases conectadas entre sí para dar al usuario el soporte necesario al acceder a la base de datos de rutas favoritas.

Tabla 110. Asociación clases FavoritosDbHelper, FavoritosDbAdapter y FavoritosCursorAdapter

5.6.3 Identificación de generalizaciones

En este apartado se estudia otro tipo de relación entre clases, la generalización, donde una clase comparte estructura y/o comportamiento con una o más clases. El término superclase se refiere a la clase que guarda la información común, mientras que el término subclase se refiere a cada uno de los descendientes de la superclase.

En esta aplicación existe un ejemplo de lo siguiente, la clase GPSTracker que es la que envía en todo momento al resto de clases la ubicación del dispositivo del usuario en cada momento a través de las coordenadas geográficas del mismo.

Del mismo modo, la clase Login es la que almacena y distribuye los datos del usuario en todo momento a las demás clases que necesiten saber quien está conectado en cada momento (por ejemplo para mostrar un saludo al mismo o para decidir las rutas de favoritos a mostrar)

5.7 Definición de las interfaces de usuario

Uno de los aspectos más importantes de esta aplicación es la interfaz, ya que es el medio de comunicación entre el usuario y el sistema. Por lo que el principal objetivo de todas las interfaces que componen la aplicación es que sean lo más intuitiva posible y fácil de manejar, para que pueda ser usada por todos los públicos.

Los usuarios finales de la aplicación no tienen por qué tener extendidos conocimientos informáticos, por lo que es fundamental que la interfaz de la aplicación resulte atractiva a la vista, clara, intuitiva y que sea fácil de manejar.

Es importante además hacer las interfaces los más modulables y reutilizables posible de cara a las posibles actualizaciones que puedan desarrollarse en un futuro.

La calidad de toda interfaz es valorada por una serie de criterios como es el tiempo que requiere el usuario para aprender a manejar la aplicación con soltura, por lo tanto éste debe minimizarse al mínimo. De este modo, se diseñará de tal forma que el usuario cometa el mínimo número de errores posible, que le sea fácil de recordar para que una vez haya aprendido cómo funciona, no le tenga que dedicar más tiempo a esta parte y, lo que es más importante, que el usuario esté satisfecho con el funcionamiento de la aplicación.

5.7.1. Identificación de perfiles y diálogos

Existen dos zonas diferenciadas en la aplicación, separadas por el hecho de que el usuario este autenticado o no lo esté.

La primera interfaz de la aplicación nos da la oportunidad de acceder al registro de nuevos usuarios (todo el manejo de dicha base de datos) aparte de actualizar toda la base de datos de radares y acceder a los servicios de ubicación de nuestro dispositivo.

Una vez autenticados, podemos acceder a todos los servicios de navegación, principalmente gestión de favoritos de cada usuario, así como realizar todo tipo de rutas, tanto libres como previamente fijadas.

5.7.2. Especificación de formatos individuales de la interfaz de la pantalla.

En este punto se especificará los formatos de pantallas que deberá cumplir la interfaz del software. Se definirán qué tipo de interfaz se usará y la interacción que tendrá el usuario con ella.

De manera general, todas las pantallas que conforman la aplicación seguirán la siguiente estructura:

- **Cabecera:** En la parte superior, contiene el icono de la aplicación a la izquierda y a su continuación, o un indicativo de la interfaz en la que nos situamos o bien el nombre de la aplicación "PFC Manuel" en mayúsculas .
- **Botones dispositivo:** Parte baja de la pantalla, una línea que contiene los botones de los botones nativos del dispositivo.
- **Área de contenido:** El resto de la pantalla, con diversas aplicaciones posteriormente explicadas en el apartado dedicado a analizar las interfaces.

Se usarán los siguientes elementos .xml para la interacción con el usuario:

- **TextView:** Para dar mensajes al usuario sobre como rellenar los EditText (explicado a continuación).

- **EditText:** Cajas de texto donde se introducen caracteres para diversas funcionalidades (como guardar un archivo o crear un usuario)
- **Button:** Elementos clickables para intercambiar las interfaces o realizar alguna función, llevan escrita su función de manera simple y concisa.
- **ImageButton:** Igual que un Button, pero en vez de llevar la función escrita, se insinúa mediante una imagen (por ejemplo, el micrófono indica que si se pulsa en ese botón, se escribe)
- **Map:** Soportado por un objeto Fragment, nos muestra el mapa con zoom a nuestra ubicación.
- **Toast:** Mensajes intermitentes que aparecen en la parte inferior de la pantalla para mostrar al usuario alguna tarea que se ha podido realizar o no se ha podido realizar (No se pudo guardar, no se ha entendido su mensaje, etc...)

5.7.2.1. Descripción de pantallas que conforman la aplicación

La aplicación desarrollada utiliza principalmente siete interfaces (más una de portada que es solo decorativa).

Para comenzar, aun sin ser una interfaz de la aplicación, se muestra como aparece el icono en la pantalla del dispositivo, para iniciar la aplicación mediante un click sobre él.

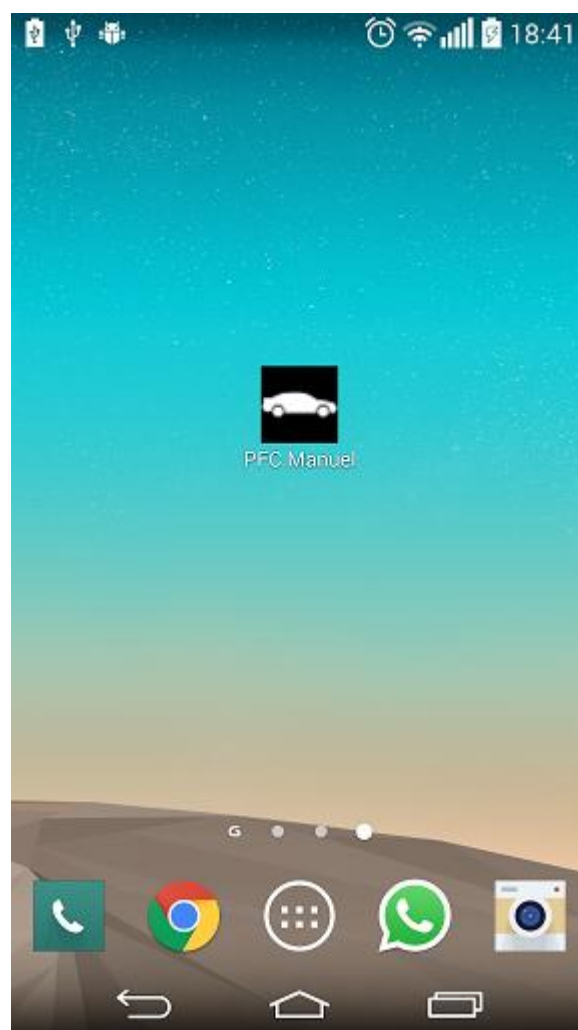


Figura 25. Interfaz principal dispositivo

Vamos a proceder a mostrar y describir todas, siguiendo el orden lógico de acceso por parte del usuario.

La primera pantalla que aparece al pulsar el icono de la aplicación, es la interfaz Portada.

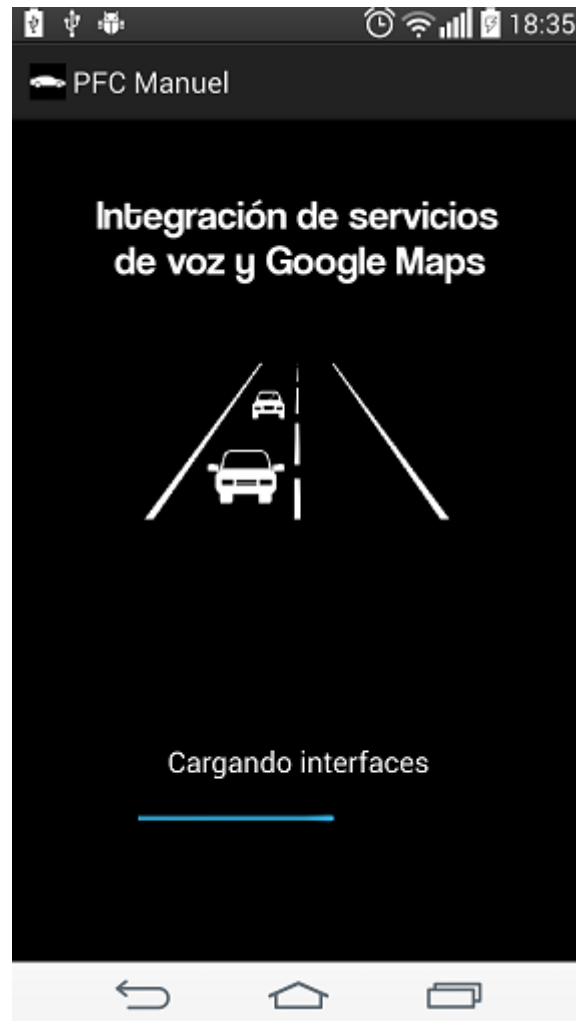


Figura 26. Interfaz Portada

Esta interfaz no tiene ninguna funcionalidad, meramente es una decoración. Presenta el título de la aplicación, con una imagen que nos hace ver parte del objetivo de la misma, y una barra de carga durante 5 segundos para posteriormente acceder a la siguiente interfaz.

La interfaz Menulnicial, nos permite actualizar la base de datos de radares, y, por otro lado, acceder a las tres siguientes interfaces:

- **MainActivity**
- **Usuarios**
- **Servicios de ubicación** (esta interfaz es propia del dispositivo ,no se ha desarrollado nada)

A continuación se muestra una captura de dicha interfaz:

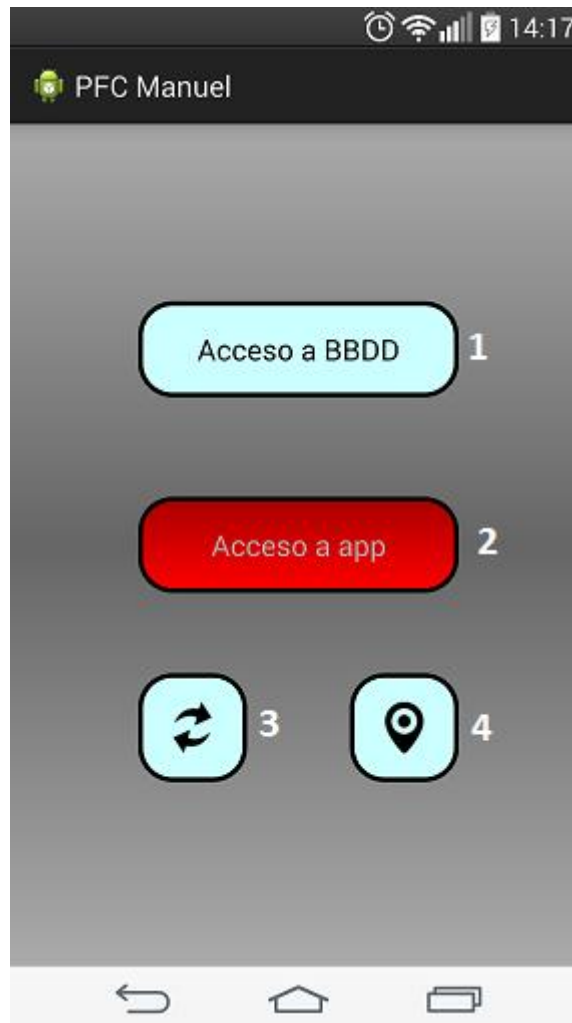


Figura 27. Interfaz Menulnicial

Para realizar la explicación de cada interfaz, se ha numerado cada punto de acción para proceder a explicar a continuación cada uno:

- **Punto 1:** Es el botón que se utiliza para acceder a la zona de administración de usuarios (interfaz Usuarios)
- **Punto 2:** Es el botón que se utiliza para acceder a la interfaz MainActivity. En este caso, aparece en rojo al no estar permitido el acceso debido a no estar activada la ubicación.
- **Punto 3:** Es el botón para actualizar la base de datos de radares.
- **Punto 4:** Es el botón para acceder a los servicios de ubicación del dispositivo

La segunda interfaz es la de usuarios, para crear nuevos usuarios o editar los parámetros de los ya existentes.

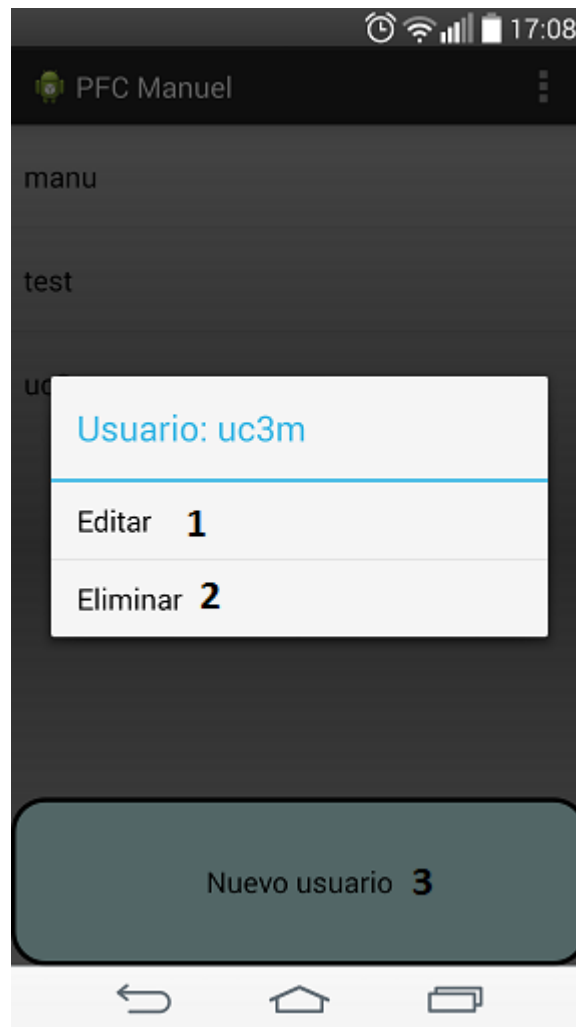


Figura 28. Interfaz Usuarios

En esta interfaz vemos una lista de usuarios ya existentes (creados previamente por el usuario del dispositivo). Al hacer click largo en un usuario, aparece un cuadro central con dos opciones:

- **Punto 1:** En el cuadro central, etiqueta editar. Se accede a la interfaz UsuariosFormulario para poder modificar los datos del usuario.
- **Punto 2:** En el cuadro central, etiqueta eliminar. Para eliminar el usuario y todas las rutas de favoritos asociadas a él.
- **Punto 3:** Botón para la creación de un nuevo usuario. Se accede a la interfaz UsuariosFormulario

La tercera interfaz es UsuariosFormulario, que tal y como indica su nombre, es la utilizada para la creación ó edición de usuarios. Se muestra a continuación una composición de ambos modos.



Figura 29. Interfaz UsuariosFormulario

- **Punto 1:** Introducción de datos de usuario, contraseña e e-mail (en ambos modos)
- **Punto 2:** Botón de creación de un usuario tras cumplir con las reglas de las casillas de datos (modo creación).
- **Punto 3:** Botón de edición de usuarios tras haber realizado las variaciones deseadas (modo edición)
- **Punto 4:** Botón de cancelación y vuelta a la interfaz Usuarios (en ambos modos).

Si en la interfaz MenuInicial accedemos a la aplicación, primero debemos autenticarnos como usuarios, en la interfaz Login.

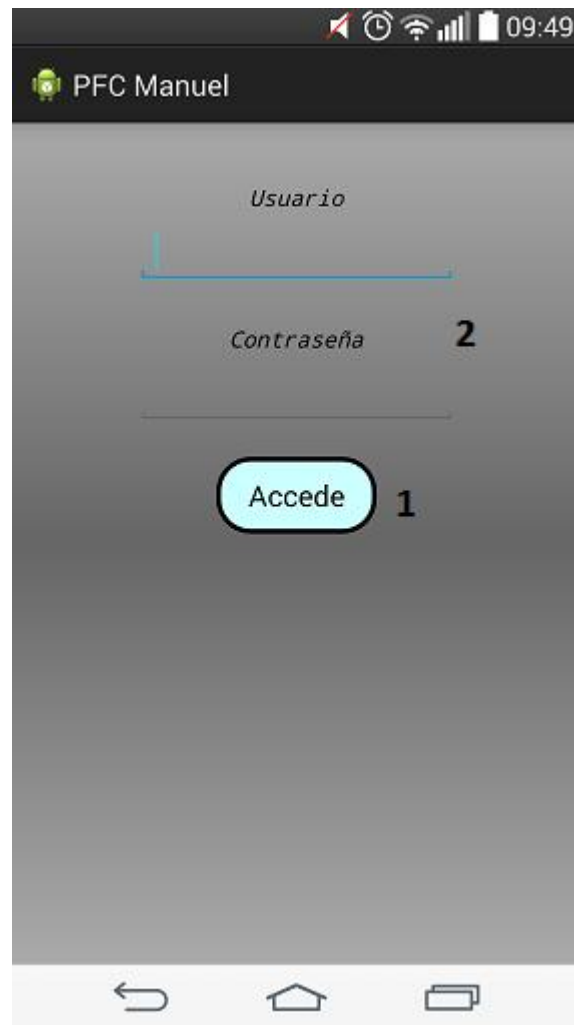


Figura 30. Interfaz Login

- **Punto 1:** Botón de acceso a la interfaz MainActivity tras la autenticación como usuario.
- **Punto 2:** Introducción de datos de (usuario y contraseña) para la autenticación de usuario.

La siguiente interfaz a la que se accede tras la autenticación como usuario es MainActivity. Desde esta pantalla el usuario podrá acceder a todas las funcionalidades de la aplicación mediante comandos de voz.

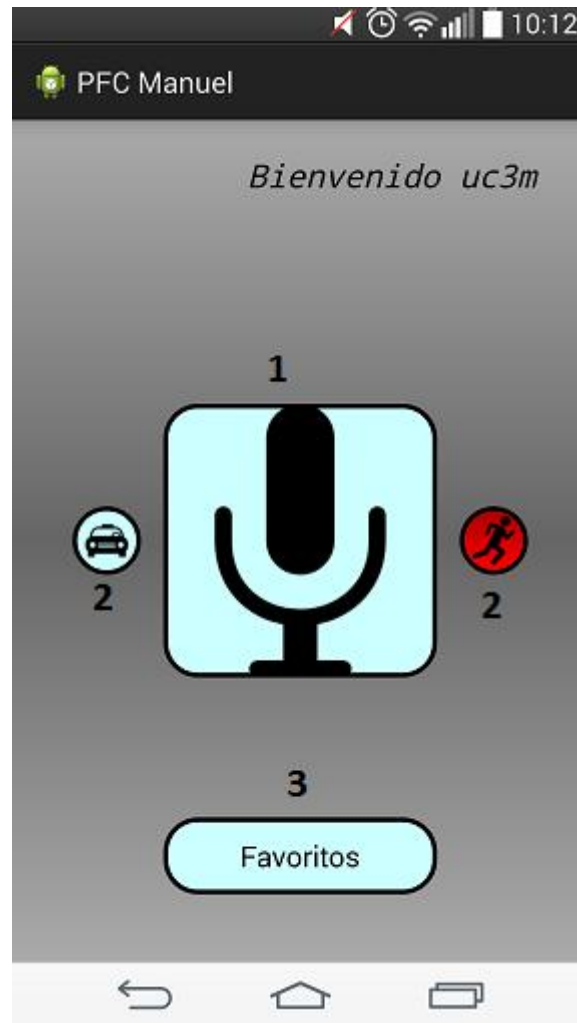


Figura 31. Interfaz MainActivity

- Punto 1: Botón de activación del sistema de reconocimiento de voz
- Punto 2: Botón de selección de modo (en azul el modo activado, en rojo el desactivado. En el ejemplo, activado el modo vehículo).
- Punto 3: Botón de acceso a la interfaz de favoritos.

Si pulsamos el botón de acceso a favoritos, también accesible por voz, llegaremos a la siguiente interfaz:

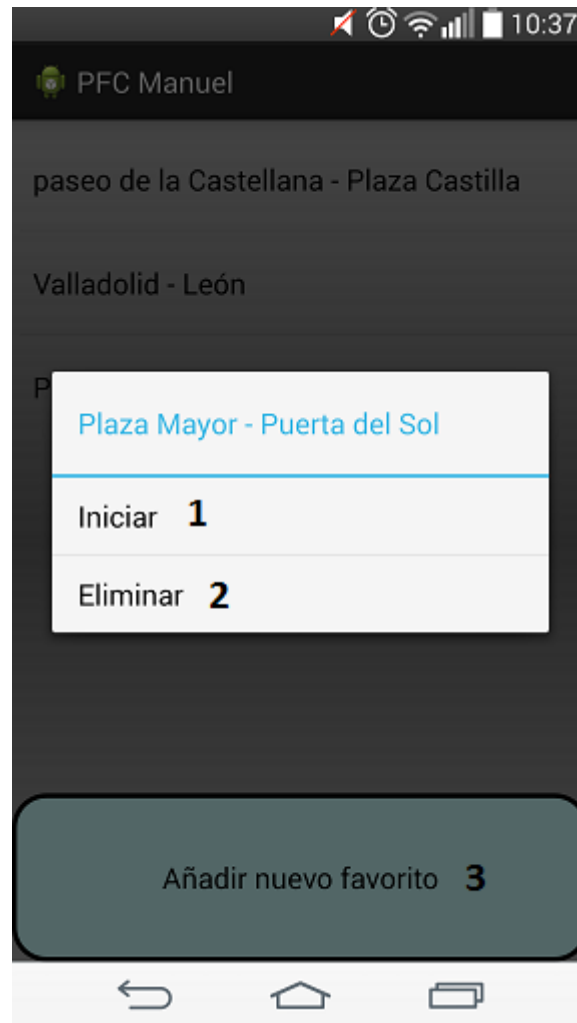


Figura 32. Interfaz Favoritos

En esta interfaz vemos una lista de rutas de favoritos ya existentes (creados previamente por el usuario del dispositivo). Al hacer click en una ruta, aparece un cuadro central con dos opciones.

- Punto 1: En el cuadro central, etiqueta iniciar. Se accede a la interfaz *MostrarMapa* para poder seguir las instrucciones de ruta.
- Punto 2: En el cuadro central, etiqueta eliminar. Para eliminar la ruta.
- Punto 3: Botón para la creación de una nueva ruta de favoritos. Se activa el reconocimiento por voz.

La siguiente interfaz es *MostrarMapa* a la cual se accede desde *MainActivity* o iniciando una ruta descargada desde la interfaz *Favoritos*, en la cual se irá desarrollando la ruta solicitada por el usuario.



Figura 33. Interfaz MostrarMapa

- **Punto 1:** Indicaciones de la ruta en proceso
- **Punto 2:** Distancia en kilómetros o metros (si es inferior a 1000) al siguiente hito de la ruta.
- **Punto 3:** Mapa de la zona en la que se está ubicado o donde empieza la ruta en cuestión.
- **Punto 4:** Línea azul que marca la ruta que debemos seguir.

Desde MainActivity, también podemos acceder a la interfaz ModoLibre que nos permite crear rutas propias o repetir rutas propias creadas previamente. En este caso mostraremos varias capturas debido a que la misma tiene varias modificaciones.



Figura 34. Interfaz ModoLibre (modelo 1)

- **Punto 1:** Botón de cargar rutas previamente creadas por el usuario
- **Punto 2:** Botón para empezar una nueva ruta libre por parte del usuario
- **Punto 3:** Mapa de la zona en la que se está ubicado o donde empieza la ruta en cuestión.
- **Punto 4:** El punto azul representa la ubicación del usuario en ese momento

Si pulsamos sobre el botón empezar tal y como se muestra en el modelo 1, cambiará a la siguiente pantalla, de la cual solo se explicarán las variaciones:

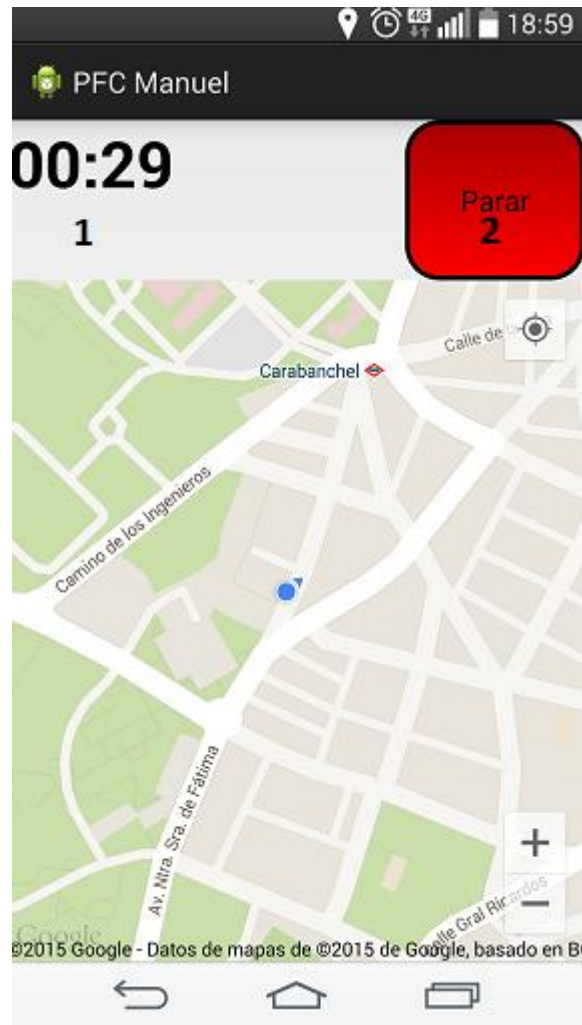


Figura 35. Interfaz ModoLibre (modelo 2)

- **Punto 1:** Un cronómetro que toma el tiempo que tardamos en realizar la ruta libre.
- **Punto 2:** Botón parar, para finalizar la ruta y elegir si se quiere acceder a la interfaz FinDeLaRuta o volver a MainActivity.

Por otro lado, si en el modelo 1 pulsamos en el botón de cargar rutina, veremos la siguiente lista de opciones:



Figura 36. Interfaz ModoLibre (modelo 3)

- **Punto 1:** El botón cargar cambia a modo no clickable, debido a que ya hemos pulsado sobre él para acceder a esta parte de la aplicación.
- **Punto 2:** Lista de las rutas que hemos creado hasta ahora. En este caso, existen dos rutas generadas y guardadas por el usuario.

Como última variante de esta interfaz, si hacemos click en una de las rutas que se muestran en el modelo 3, accederemos al cuarto y último modelo de esta interfaz:

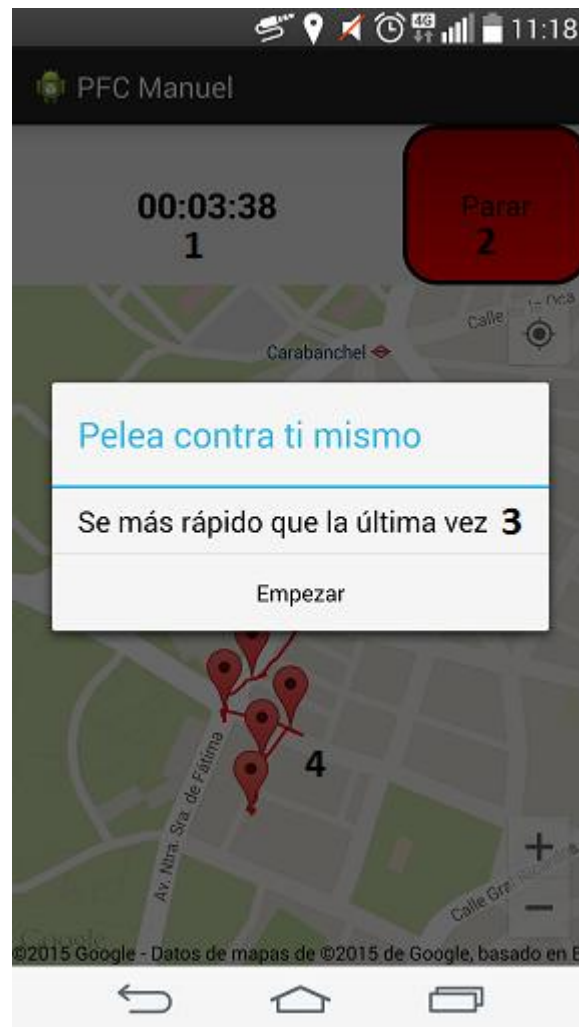


Figura 37. Interfaz Modolibre (modelo 4)

- **Punto 1:** Cuenta atrás con el tiempo que se tardó en realizar la ruta en la última ocasión.
- **Punto 2:** Pasa a estar bloqueado debido a que la competición para mejorar el tiempo de la anterior ocasión que se realizó la ruta no puede pararse.
- **Punto 3:** Cuadro central para dar la opción al usuario de elegir cuando se empieza.
- **Punto 4:** La ruta recorrida, con sus propios checkpoints de paso para evitar trampas.

La última interfaz es FinDeLaRuta, a la cual se accede desde el modelo 2 de la interfaz Modolibre:

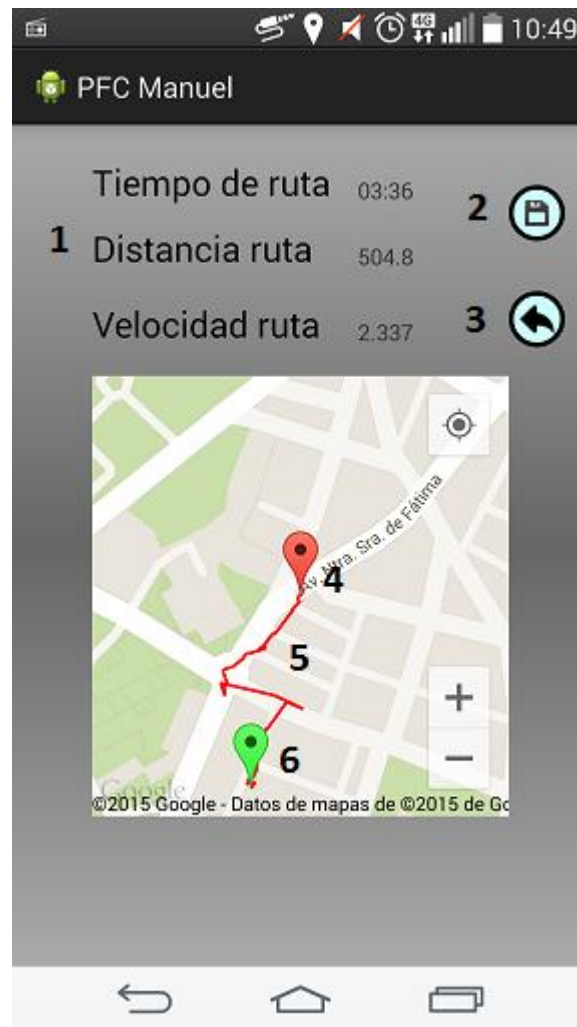


Figura 38. Interfaz FinDeLaRuta

- **Punto 1:** Estadísticas de ruta (Tiempo total, distancia recorrida y velocidad media).
- **Punto 2:** Botón guardar ruta, nos muestra un cuadro central para introducir el nombre del archivo donde almacenaremos la ruta.
- **Punto 3:** Botón de volver a la interfaz MainActivity
- **Punto 4:** Marcador de inicio de ruta.
- **Punto 5:** Dibujo de la ruta de inicio (Marcador rojo) a final (Marcador verde)
- **Punto 6:** Marcador de final de ruta.

En este punto, merecería la pena comentar que se comentará en el anexo A el motivo de porque las líneas son inestables, en el se explicará la diferencia entre el algoritmo usado por Google respecto al método de unión utilizado para la realización de rutas libres.

5.8 Análisis de consistencia y especificación de requisitos

En esta sección se detalla cómo se asegurará la calidad entre los distintos modelos que se han generado como productos, comprobando la consistencia entre ellos.

5.8.1 Verificación y análisis de consistencia entre modelos

Mediante esta tarea se comprobará que cada modelo generado es correcto, comprobando la coherencia de los modelos y la existencia de ambigüedad o redundancia.

Se ha comprobado que los casos de uso se pueden realizar con las clases definidas en el modelo de clases.

Por último se muestran las matrices de trazabilidad que permiten asegurar la consistencia existente entre el Estudio de Viabilidad del Sistema y el Análisis.

5.8.1.1 Matrices de Requisitos de usuario de capacidad-Casos de uso

	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008	CU-009	CU-010	CU-011	CU-012	CU-013	CU-014	CU-015
RU-C-001			X				X								
RU-C-002				X		X	X						X	X	
RU-C-003				X	X	X		X	X	X				X	
RU-C-004				X		X			X	X					
RU-C-005				X		X	X				X				
RU-C-006						X	X	X	X			X			
RU-C-007									X	X					
RU-C-008							X				X	X			
RU-C-009									X						
RU-C-010									X	X					X
RU-C-011				X		X			X	X					
RU-C-012			X				X								
RU-C-013		X		X	X				X					X	X
RU-C-014				X	X	X		X	X	X		X		X	X
RU-C-015									X	X					X
RU-C-016						X						X	X		

Tabla 111. Matriz de trazabilidad Requisitos de Usuario - Casos de Uso

5.8.1.2 Matrices de Requisitos software-Casos de uso

	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008	CU-009	CU-010	CU-011	CU-012	CU-013	CU-014	CU-015
RU-F-001			X												
RU-F-002							X					X			
RU-F-003	X					X								X	
RU-F-004				X	X	X		X	X	X				X	
RU-F-005				X		X			X	X				X	
RU-F-006						X			X	X				X	
RU-F-007							X		X		X				
RU-F-008							X		X			X			
RU-F-009			X				X			X					
RU-F-010									X	X					
RU-F-011				X		X									
RU-F-012		X			X				X						X
RU-F-013				X		X		X	X					X	
RU-F-014														X	X
RU-F-015	X														

Tabla 112. Matriz de trazabilidad Requisitos Software - Casos de Uso

5.8.2 Validación de los modelos

Se ha comprobado la integridad de las matrices de trazabilidad asegurando que los modelos planteados son consistentes.

6. Diseño del Sistema

6.1. Alcance

El presente documento presenta una especificación detallada de los componentes en los que se dividirá el sistema, permitiendo cubrir todas las decisiones de diseño que se verán reflejadas en la fase construcción.

Por otro lado, se especificará el entorno tecnológico necesario para que el sistema pueda entrar en ejecución. Contendrá además la planificación de capacidades, los requisitos de administración, el control de accesos, la seguridad y la operación.

A continuación, se describen con mayor detalle las principales tareas que se presentan en este documento:

- **Definición de la arquitectura del sistema:** Se realizará una división del sistema con el objetivo de organizar y facilitar el diseño. Estos subsistemas serán partes lógicas y coherentes, que tendrán interfaces de comunicación entre ellas bien definidas. Una vez definidos estos subsistemas, habrá que acoplarlos adecuadamente dentro de la arquitectura del sistema.

- **Diseño de la arquitectura de soporte:** En esta tarea se determinarán los mecanismos genéricos de diseño que servirán posteriormente de ayuda para el diseño detallado del sistema de información. Por tanto, se deberá realizar en paralelo con el diseño detallado, ya que hay mucho en común entre ambas actividades y los cambios de una tarea afectan a la otra.

- **Diseño de casos de uso reales:** Se partirá de los casos de uso identificados en la fase de análisis del sistema y se relacionarán con las clases identificadas en el diseño, en lugar de trabajar con el modelo conceptual del documento de Análisis.

- **Diseño de clases:** El objetivo es transformar el modelo de clases del análisis en un modelo de clases de diseño. Dicho modelo es mucho más extenso que el anterior, ya que recoge información de los atributos, métodos y las relaciones que existen entre las distintas clases. Para ello, es fundamental considerar las especificaciones tecnológicas elegidas para la implementación.

- **Diseño físico de datos:** Indica el formato de los ficheros que la aplicación va a utilizar.

- **Verificación y aceptación de la arquitectura del sistema:** En este apartado se comprobará la trazabilidad entre componentes, clases y casos de uso.

- **Especificación técnica del plan de pruebas:** Se especificará en detalle el entorno de pruebas a realizar para comprobar el correcto funcionamiento del sistema.

- **Establecimiento de requisitos de implantación:** En esta actividad se completará el catálogo de requisitos con aquéllos relacionados con la implantación del sistema en el entorno.

Todo esto permite preparar los medios y recursos necesarios para que los usuarios puedan manejar la aplicación sin problemas.

Es importante destacar que el documento que aquí se presenta es una adaptación del diseño planteado por Métrica Versión 3, ya que debido a las características del proyecto (se trata de un proyecto pequeño) algunas de las actividades son demasiado extensas para el problema tratado. Muestra de ello es que la actividad definida por métrica versión 3 DSI 8 Generación de Especificaciones de Construcción no se ha desarrollado. Además, por la misma razón, en otras actividades no se ha profundizado demasiado como es el caso de la actividad DSI 3 Diseño de casos de uso reales en el que se han definido los casos de uso más representativos de cada uno de los módulos del sistema.

6.2 Definición de la arquitectura

6.2.1 Definición de los niveles de la arquitectura

La arquitectura del sistema a desarrollar en el presente proyecto atiende a la arquitectura ARM propia de los servicios de Android

6.2.2. Identificación de requisitos de diseño y construcción.

Para la construcción de esta aplicación se ha solicitado una serie de requisitos a nivel de código:

- Diseñar un código ordenado y legible para que en algún momento otras personas ajenas al equipo desarrollador puedan realizar modificaciones
- Tratamiento de las excepciones que pueden ir surgiendo a través de los diversos métodos del código
- Visualización de mensajes de aviso para el usuario.
- Utilización de procesos en background
- Mensajes en modo “debug” para corregir fallos.

6.2.3 Especificación de estándares y normas de diseño y construcción

Entre las actividades a realizar en este proyecto se encuentra la codificación del sistema, lo cual hace indispensable la tarea de definir un estándar que sirva de guía tanto a diseñadores

como a programadores a la hora de generar el código fuente. A continuación se detallan las consideraciones a tener en cuenta:

6.2.3.1 Nombres de Ficheros

Los nombres de los ficheros que conforman el código fuente del sistema deberán ser representativos de la funcionalidad que contiene cada uno.

Además, su longitud no deberá superar los 25 caracteres y la primera letra deberá estar en escrita en mayúscula. En caso de ser un nombre compuesto la primera letra de cada palabra será escrita en mayúscula y el resto en minúsculas.

Ejemplo: NombreDeFichero

6.2.3.2 Idioma

Los idiomas utilizados a lo largo de todo el desarrollo del proyecto serán el castellano y el inglés debido a la integración de partes de código ya existentes con posteriores modificaciones, aun así, todos los comentarios los comentarios que se muestren al usuario serán en castellano así como los requerimientos para el mismo.

6.2.3.3 Cabeceras de las clases

Cada una de las clases generadas mostrará al principio de su código un conjunto de líneas de comentario siguiendo el formato que se muestra a continuación:

'Nombre de la clase:

'Descripción:

'Autor:

6.2.3.4 Visibilidad de las clases

Las clases podrán ser públicas (public) o privadas (private).

6.2.3.5 Identificadores de las clases

Los identificadores de las clases estarán formados únicamente por letras, como excepción, la letra "ñ" ha sido eliminada del código para evitar conflictos con posibles modificaciones en

dispositivos que no dispongan de este carácter. Por otro lado, en las partes indicadas por el usuario, no se puede evitar esto, ya que no se puede hacer a un usuario que no viaje a Valdepeñas (Ciudad Real) o a la calle Nuñez de Balboa (Madrid)

Además, los identificadores siempre comenzarán con una letra mayúscula. En el caso de que sea un nombre compuesto, la primera letra a partir de la segunda palabra será mayúscula también.

Ejemplo: IdentificadorCompuesto

6.2.3.6 Constantes

En caso de utilizar constantes, su nombre estará formado únicamente por letras mayúsculas. En el caso de que sea una palabra compuesta, las diferentes palabras se separarán con guiones bajos “_”.

Ejemplo: CONSTANTE

6.2.3.7 Variables

Los nombres de variables estarán formados por letras minúsculas. En el caso de ser una palabra compuesta, la primera letra de cada palabra posterior a la primera palabra será mayúscula.

Ejemplo: nombreVariable

6.2.3.8 Funciones y procedimientos

En cuanto a las funciones y procedimientos, es muy importante que el nombre describa con la mayor precisión posible la funcionalidad que desempeña. Para ello, los nombres utilizados serán verbos que describan la acción que hace la función o procedimiento siempre que sea posible.

Los nombres deberán ser escritos en letras minúsculas únicamente, utilizando mayúsculas en el caso de tratarse de un nombre compuesto, en cuyo caso la primera letra de cada palabra posterior a la primera será escrita en mayúsculas.

Ejemplo: nombreFuncion

Los posibles parámetros de entrada o salida de las funciones y procedimientos seguirán el formato de nomenclatura descrito en el apartado 6.3.2.7. Variables

6.2.4 Identificación de los subsistemas de la aplicación

No aplica, el sistema no contiene subsistemas.

6.2.5. Especificación del entorno tecnológico

En este apartado se describe el conjunto de especificaciones referentes al entorno tecnológico del sistema diseñado. Se definirán el conjunto de elementos que compondrán el eje central de la infraestructura técnica que va a adoptar el sistema, además de abordar aquellos inconvenientes técnicos que puedan afectar el desarrollo del proyecto.

6.2.5.1. Hardware

Para el desarrollo del proyecto se emplearán un sistema: un ordenador un ordenador portátil Asus.

6.2.5.2 Software

El sistema operativo instalado en la máquina empleada para la ejecución del sistema es Microsoft Windows 7.

6.2.6. Especificación de requisitos de seguridad y operación

Para analizar la especificación de los requisitos de operación se remite al lector al apartado 5.4.1.3. Requisitos de operación.

6.2.7 Estudio de la seguridad requerida en el proceso de diseño del

sistema

Al tratarse de un sistema que no maneja información sensible para el usuario del propio sistema no es necesario supervisar la seguridad de las actividades del proceso de diseño.

6.2.8 Análisis de los riesgos del entorno tecnológico

No aplica, los riesgos asociados al entorno tecnológico son mínimos debido al software utilizado.

6.3 Diseño de la arquitectura de soporte

En este punto se presentan los elementos estructurales más significativos del sistema: componentes y clases. Además, son indicadas las colaboraciones entre los distintos componentes a través de la definición de sus interfaces.

A partir de esta representación preliminar se irán definiendo las clases asociadas a los componentes, sus interfaces y atributos hasta llegar al diseño detallado del sistema.

6.3.1 Diseño de subsistemas de soporte

En este punto se presentan los elementos estructurales más significativos del sistema: componentes y clases.

Como se introdujo en el apartado Definición de la arquitectura, el sistema tiene una arquitectura específica compuesta por tres componentes principales: modelo, vista y controlador.

A continuación se describen estos componentes y la interacción entre ellos, para posteriormente analizar las clases que los componen.

A partir de esta representación preliminar se irán definiendo las clases asociadas a los componentes, sus interfaces y atributos hasta llegar al diseño detallado del sistema.

A continuación se indican las características particulares de los componentes que constituyen la aplicación. Para cada uno de ellos se especificará:

- **Identificador:** será único para cada uno de los componentes. Para ello utilizaremos la siguiente nomenclatura "C-XXX", donde XXX es el número del componente. Además, se incluirá el nombre del componente.
 - **Tipo:** Tipo de elemento, en este caso será componente.
 - **Propósito:** Fin del componente.
 - **Función:** De qué se encarga el componente.
 - **Nomenclatura:** Que nombre tendrán las clases que pertenezcan al componente.
 - **Dependencias:** Componente que utiliza el elemento tratado para realizar sus funciones.
- **Interfaz:** Servicios proporcionados para el resto de elementos.

- **Recursos:** Hace referencia a elementos de tipo HW que el componente puede necesitar para realizar su función, por ejemplo, impresoras, buffers.
- **Referencias:** Documentación necesaria para poder entender el componente.
- **Proceso:** Algoritmos especiales de procesamiento.
- **Datos:** Datos internos del elemento, atributos, relaciones de interés que no se hayan indicado en la descripción, posibles valores de alguno de los elementos.

6.3.2 Diseño de casos de uso reales

Este punto tiene como propósito especificar el comportamiento del sistema de información para un caso de uso, mediante objetos o subsistemas de diseño que interactúan y determinar las operaciones de las clases e interfaces de los distintos subsistemas de diseño.

Una vez identificadas las clases participantes dentro de un caso de uso, es necesario completar los escenarios que se recogen del análisis, incluyendo las clases de diseño que correspondan y teniendo en cuenta las restricciones del entorno tecnológico, es decir, detalles relacionados con la implementación del sistema.

6.3.3 Identificación de clases asociadas a un caso de uso

A continuación se muestran las clases identificadas en el diseño y su asociación con los distintos caso de uso del sistema.

Identificador	Clases Asociadas
CU-001	Portada
CU-002	
CU-003	Login Usuarios UsuariosCursorAdapter UsuariosDbAdapter UsuariosFormulario Mail
CU-004	AndroidXMLParsingActivity CalcularDistanciaPuntos GPSTracker MainActivity MostrarMapa RutasHelper
CU-005	GPSTracker

CU-006	CalcularDistanciaPuntos GPSTracker MostrarMapa
CU-007	Favoritos FavoritosCursorAdapter FavoritosDbHelper FavoritosDbAdapter FavoritosDescargaArchivo
CU-008	Favoritos MostrarMapa GPSTracker
CU-009	ModoLibre FinDeLaRuta GPSTracker
CU-010	ModoLibre FinDeLaRuta GPSTracker
CU-011	MainActivity Favoritos
CU-012	Favoritos MostrarMapa
CU-013	Favoritos Usuarios ModoLibre MostrarMapa
CU-014	CalculoDistanciaPuntos
CU-015	ModoLibre

Tabla 113. Clases de diseño asociadas a Casos de uso

6.3.2 Diseño de la realización de los casos de uso

En este apartado se especifican de forma extendida los casos de uso detallados en el análisis, haciendo corresponder los distintos escenarios con acciones concretas identificadas en las clases obtenidas en el desarrollo del documento.

A continuación se muestra el diagrama de secuencia para el caso de uso CU-003: Manejo Usuarios.

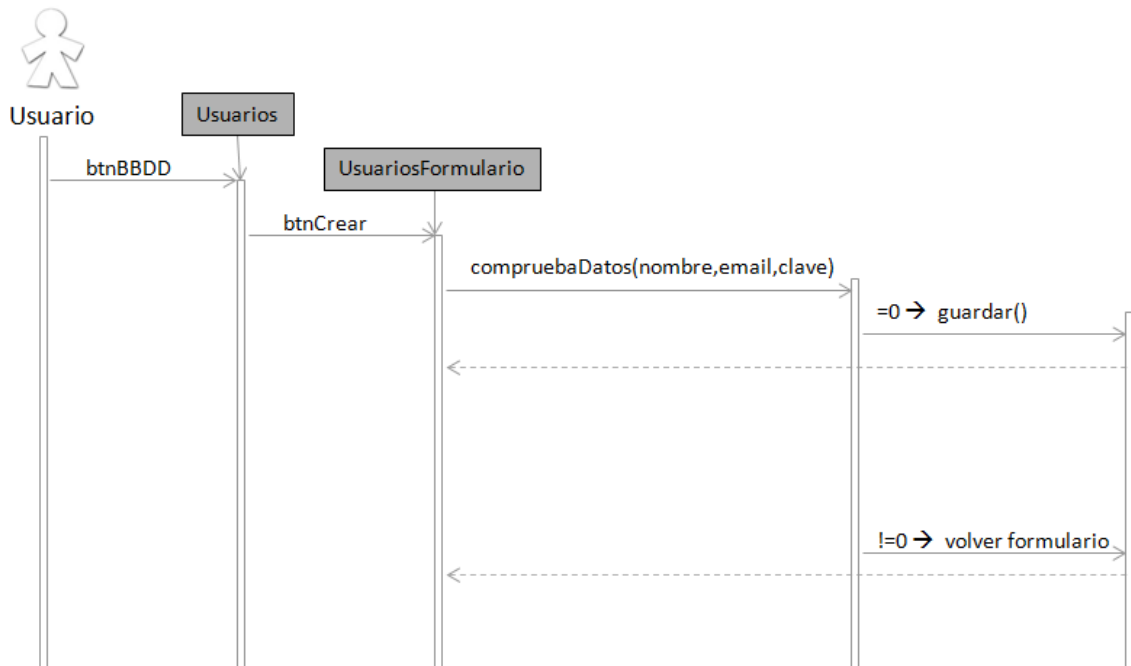


Figura 39. Diagrama de secuencia Caso de Uso CU-003

El siguiente diagrama de secuencia representa el caso de uso CU-009: Creación rutas

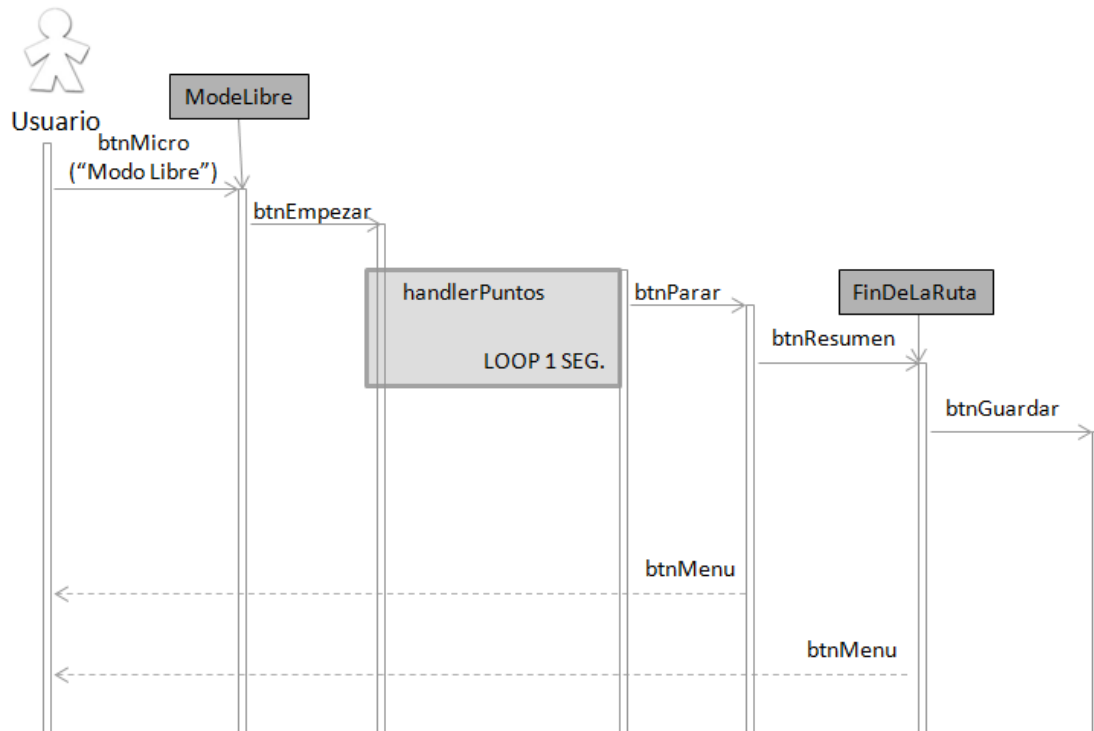


Figura 40. Diagrama de secuencia Caso de Uso CU-009

6.4 Diseño de clases

En esta sección se desarrolla el modelo de clases del sistema que se está diseñando, completando el realizado durante la fase de Análisis del Sistema.

Los modelos aquí presentados servirán de guía al programador, por lo que los identificadores, relaciones, atributos y funciones son orientativos, dejando al programador libertad para programar bajo su propio criterio.

6.4.1 Identificación de clases de diseño

A continuación se muestran los modelos de clases de diseño de la aplicación:

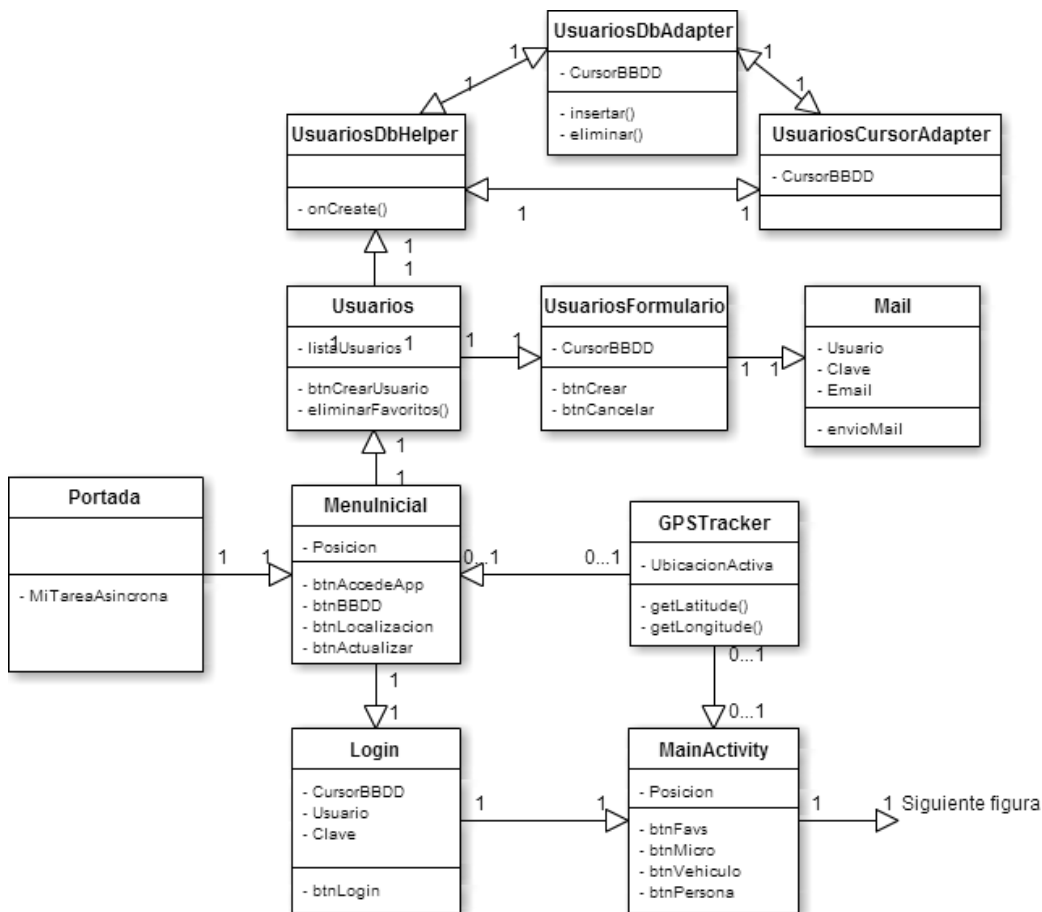


Figura 41. Diagrama de clases de diseño (Parte 1)

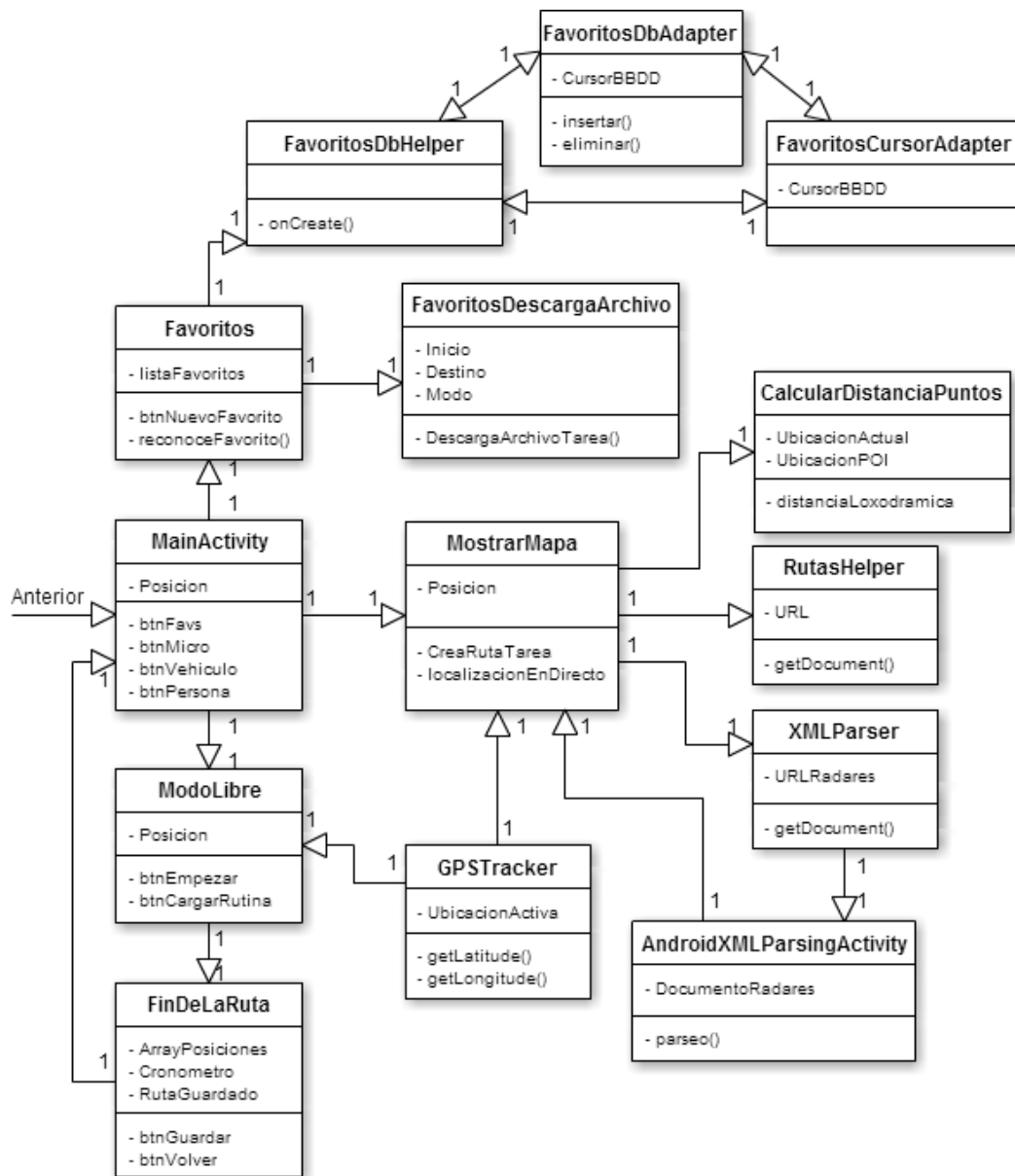


Figura 42. Diagrama de clases de diseño (Parte 2)

6.4.2 Identificación de atributos y métodos de clases de diseño

A continuación se describen las clases de diseño identificadas, incluyendo el detalle de sus atributos y métodos.

Clase	AndroidXMLParsingActivity
Descripción	Interpretación del documento que contiene los diversos radares ubicados en España
Dependencias	XMLParser, MostrarMapa
ATRIBUTOS	
Identificador	DocumentoRadares
Descripción	Documento recibido desde XMLParser con las ubicaciones de todos los radares
FUNCIONES Y PROCEDIMIENTOS	
Signatura	parseo()
Descripción	Lectura y almacena toda la información y devuelve un array con la misma a MostrarMapa

Tabla 114. Atributos, funciones y procedimientos clase AndroidXMLParsingActivity

Clase	CalcularDistanciaPuntos
Descripción	Calcular la distancia entre la ubicación actual y los diversos puntos de interés
Dependencias	MostrarMapa
ATRIBUTOS	
Identificador	UbicacionActual
Descripción	Ubicación tomada por el dispositivo
Identificador	UbicacionPOI
Descripción	Ubicación de un punto de interés, sea el siguiente paso en una ruta o un radar de la base de datos.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	distanciaLoxodramica()
Descripción	Función para el cálculo de distancia a partir de los atributos mostrados arriba.

Tabla 115. Atributos, funciones y procedimientos clase CalcularDistanciaPuntos

Clase	Favoritos
Descripción	Manejo de la base de datos de Favoritos
Dependencias	MostrarMapa, FavoritosDbHelper, FavoritosDescargaArchivo
ATRIBUTOS	
Identificador	ListaFavoritos
Descripción	Recibe todos los favoritos que tiene un usuario filtrando por el nombre del mismo y los muestra por pantalla
FUNCIONES Y PROCEDIMIENTOS	
Signatura	btnNuevoFavorito
Descripción	Activa el reconocimiento de voz para que el usuario diga la ruta a guardar
Signatura	reconoceFavorito()
Descripción	Comprueba si la ruta dicha por el usuario es viable y procede a almacenarla.

Tabla 116. Atributos, funciones y procedimientos clase Favoritos

Clase	FavoritosCursorAdapter
Descripción	Apertura de la base de datos de datos a partir del cursor recibido
Dependencias	FavoritosDbHelper, FavoritosDbAdapter
ATRIBUTOS	
Identificador	CursorBBDD
Descripción	Cursor recibido para abrir la base de datos.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	No aplica
Descripción	No aplica

Tabla 117. Atributos, funciones y procedimientos clase FavoritosCursorAdapter

Clase	FavoritosDbAdapter
Descripción	Definición de constantes de la base y manejo de la misma
Dependencias	FavoritosDbHelper, FavoritosCursorAdapter
ATRIBUTOS	
Identificador	CursorBBDD
Descripción	Cursor recibido para abrir la base de datos según los parámetros requeridos.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	eliminar()
Descripción	Elimina un registro de la tabla
Signatura	insertar()
Descripción	Inserta un nuevo registro en la tabla

Tabla 118. Atributos, funciones y procedimientos clase FavoritosDbAdapter

Clase	FavoritosDbHelper
Descripción	Creación o apertura según necesidad de la tabla
Dependencias	Favoritos, FavoritosCursorAdapter, FavoritosDbAdapter
ATRIBUTOS	
Identificador	No aplica
Descripción	No aplica
FUNCIONES Y PROCEDIMIENTOS	
Signatura	onCreate()
Descripción	Crea o abre la tabla a partir de las constantes de FavoritosDbAdapter

Tabla 119 Atributos, funciones y procedimientos clase FavoritosDbHelper

Clase	FavoritosDescargaArchivo
Descripción	Descargar la ruta solicitada por el usuario
Dependencias	Favoritos
ATRIBUTOS	
Identificador	Inicio
Descripción	Dirección de inicio de la ruta a guardar
Identificador	Destino
Descripción	Dirección de destino de la ruta a guardar
Identificador	Modo
Descripción	Modo de realizar la ruta a guardar
FUNCIONES Y PROCEDIMIENTOS	
Signatura	DescargaArchivoTarea
Descripción	Tarea asíncrona que se encarga de la descarga de la ruta desde Internet

Tabla 120 Atributos, funciones y procedimientos clase FavoritosDescargaArchivo

Clase	FinDeLaRuta
Descripción	Muestra de un resumen de una ruta generada por el usuario
Dependencias	ModoLibre, MainActivity
ATRIBUTOS	
Identificador	ArrayPosiciones
Descripción	Array bidimensional con todos los pares de coordenadas por los que ha pasado el usuario almacenados en orden.
Identificador	Cronometro
Descripción	Tiempo de recorrido de la ruta
Identificador	RutaGuardado
Descripción	Path de la carpeta de almacenamiento
FUNCIONES Y PROCEDIMIENTOS	
Signatura	btnGuardar
Descripción	El usuario ha de indicar un nombre para guardar el archivo
Signatura	btnVolver
Descripción	El usuario regresa a MainActivity y elimina todos los datos almacenados de manera provisional.

Tabla 121. Atributos, funciones y procedimientos clase FinDeLaRuta

Clase	GPSTracker
Descripción	Recopilación de coordenadas geográficas en tiempo real
Dependencias	MenuInicial, ModoLibre, MostrarMapa
ATRIBUTOS	
Identificador	Ubicación activa
Descripción	Es necesario que los servicios de ubicación del terminal estén activos.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	getLatitude()
Descripción	Nos devuelve la latitud del dispositivo
Signatura	getLongitude()
Descripción	Nos devuelve la longitud del dispositivo

Tabla 122. Atributos, funciones y procedimientos clase GPSTracker

Clase	Login
Descripción	Autenticación de usuarios
Dependencias	MenuInicial, MainActivity
ATRIBUTOS	
Identificador	CursorBBDD
Descripción	Cursor para acceder al registro necesario para la comprobación
Identificador	Usuario
Descripción	Nombre de usuario introducido por pantalla
Identificador	Clave
Descripción	Clave de usuario introducido por pantalla
FUNCIONES Y PROCEDIMIENTOS	
Signatura	btnLogin
Descripción	Realiza la comprobación de datos y accede a MainActivity si estos son correctos

Tabla 123. Atributos, funciones y procedimientos clase Login

Clase	Mail
Descripción	Presentación de la aplicación
Dependencias	UsuariosFormulario
ATRIBUTOS	
Identificador	Usuario
Descripción	Nombre de usuario introducido por pantalla
Identificador	Clave
Descripción	Clave de usuario introducido por pantalla
Identificador	Email
Descripción	Email de usuario introducido por pantalla
FUNCIONES Y PROCEDIMIENTOS	
Signatura	envioMail
Descripción	Envía un mail con la clave y el nombre de usuario al correo indicado en el registro.

Tabla 124. Atributos, funciones y procedimientos clase Mail

Clase	MainActivity
Descripción	Clase principal de la aplicación desde la que desarrollar todas las acciones
Dependencias	Favoritos, MostrarMapa, ModoLibre, GPSTracker, Login
ATRIBUTOS	
Identificador	Posicion
Descripción	Ubicación actual del dispositivo
FUNCIONES Y PROCEDIMIENTOS	
Signatura	btnFavs
Descripción	Acceso a la interfaz de favoritos
Signatura	btnMicro
Descripción	Inicio del reconocimiento de voz
Signatura	btnVehiculo
Descripción	Selección de modo vehículo al crear una ruta
Signatura	btnPersona
Descripción	Selección del modo andar al crear una ruta

Tabla 125. Atributos, funciones y procedimientos clase MainActivity

Clase	MenuInicial
Descripción	Menú inicial de la aplicación
Dependencias	Portada, Usuarios, Login, GPSTracker
ATRIBUTOS	
Identificador	Posicion
Descripción	Ubicación actual del dispositivo
FUNCIONES Y PROCEDIMIENTOS	
Signatura	btnAccedeApp
Descripción	Accede a la interfaz Login
Signatura	btnBBDD
Descripción	Accede a la interfaz Usuarios
Signatura	btnActualizar
Descripción	Actualiza la base de datos de radares si existe alguna nueva versión
Signatura	btnLocalizacion
Descripción	Accede a la pantalla de servicios de localización del dispositivo

Tabla 126. Atributos, funciones y procedimientos clase MenuInicial

Clase	ModoLibre
Descripción	Creación de rutas libres de usuario
Dependencias	MainActivity, FinDeLaRuta
ATRIBUTOS	
Identificador	Posicion
Descripción	Ubicación actual del dispositivo
FUNCIONES Y PROCEDIMIENTOS	
Signatura	btnEmpezar
Descripción	Comienza una ruta libre
Signatura	btnCargarRutina
Descripción	Muestra la lista de rutas almacenadas para iniciar la deseada

Tabla 127. Atributos, funciones y procedimientos clase ModoLibre

Clase	MostrarMapa
Descripción	Presentación de la aplicación
Dependencias	MainActivity, CalcularDistanciaPuntos, GPSTracker, RutasHelper,XMLParser,AndroidParsingActivity
ATRIBUTOS	
Identificador	Posicion
Descripción	Ubicación actual del dispositivo
FUNCIONES Y PROCEDIMIENTOS	
Signatura	CreaRutaTarea
Descripción	Tarea asíncrona para la consulta, almacenamiento y dibujo de la ruta seleccionada por el usuario.
Signatura	localizacionEnDirecto()
Descripción	Actualización de la posición en tiempo real para el seguimiento de la ruta.

Tabla 128. Atributos, funciones y procedimientos clase MostrarMapa

Clase	Portada
Descripción	Presentación de la aplicación
Dependencias	MenuInicial
ATRIBUTOS	
Identificador	No aplica
Descripción	No aplica
FUNCIONES Y PROCEDIMIENTOS	
Signatura	MiTareaAsincrona
Descripción	Contador de tiempo para la presentación

Tabla 129. Atributos, funciones y procedimientos clase Portada

Clase	RutasHelper
Descripción	Creación de la ruta a través de la URL recibida
Dependencias	MostrarMapa
ATRIBUTOS	
Identificador	URL
Descripción	Dirección web para consultar los pasos a seguir por el usuario
FUNCIONES Y PROCEDIMIENTOS	
Signatura	getDocument()
Descripción	Descarga del documento para su interpretación

Tabla 130. Atributos, funciones y procedimientos clase RutasHelper

Clase	Usuarios
Descripción	Manejo de la base de datos de Usuarios
Dependencias	MenuInicial, UsuariosDbHelper, UsuariosFormulario
ATRIBUTOS	
Identificador	ListaUsuarios
Descripción	Recibe todos los usuarios que tiene un usuario filtrando por el nombre del mismo y los muestra por pantalla
FUNCIONES Y PROCEDIMIENTOS	
Signatura	btnCrearUsuario
Descripción	Accede a la interfaz de formulario de usuarios
Signatura	eliminarFavoritos ()
Descripción	Conecta con la base de datos de Favoritos para eliminar las rutas que dependen del usuario a eliminar.

Tabla 131. Atributos, funciones y procedimientos clase Usuarios

Clase	UsuariosCursorAdapter
Descripción	Apertura de la base de datos de datos a partir del cursor recibido
Dependencias	UsuariosDbHelper, UsuariosDbAdapter
ATRIBUTOS	
Identificador	CursorBBDD
Descripción	Cursor recibido para abrir la base de datos.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	No aplica
Descripción	No aplica

Tabla 132. Atributos, funciones y procedimientos clase UsuariosCursorAdapter

Clase	UsuariosDbAdapter
Descripción	Definición de constantes de la base y manejo de la misma
Dependencias	UsuariosDbHelper, UsuariosCursorAdapter
ATRIBUTOS	
Identificador	CursorBBDD
Descripción	Cursor recibido para abrir la base de datos según los parámetros requeridos.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	eliminar()
Descripción	Elimina un registro de la tabla
Signatura	insertar()
Descripción	Inserta un nuevo registro en la tabla

Tabla 133. Atributos, funciones y procedimientos clase UsuariosDbAdapter

Clase	UsuariosDbHelper
Descripción	Creación o apertura según necesidad de la tabla
Dependencias	Usuarios, UsuariosCursorAdapter, UsuariosDbAdapter, UsuariosFormulario
ATRIBUTOS	
Identificador	No aplica
Descripción	No aplica
FUNCIONES Y PROCEDIMIENTOS	
Signatura	onCreate()
Descripción	Crea o abre la tabla a partir de las constantes de UsuariosDbAdapter

Tabla 134. Atributos, funciones y procedimientos clase UsuariosDbHelper

Clase	UsuariosFormulario
Descripción	Formulario de entrada de nuevos usuarios
Dependencias	Mail, Usuarios, UsuariosDbHelper
ATRIBUTOS	
Identificador	CursorBBDD
Descripción	Recibe un cursor para navegar por la base de datos y saber el lugar en el que colocar el siguiente registro.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	btnCrear
Descripción	Añade un registro a la base de datos de usuarios con los datos añadidos por el mismo
Signatura	btnCancelar
Descripción	Cancela la creación de un usuario y vuelve a la lista de los mismos.

Tabla 135. Atributos, funciones y procedimientos clase UsuariosFormulario

Clase	XMLParser
Descripción	Descargar la ruta solicitada por el usuario
Dependencias	MostrarMapa, AndroidXMLParsingActivity
ATRIBUTOS	
Identificador	Inicio
Descripción	Dirección de inicio de la ruta a guardar
Identificador	Destino
Descripción	Dirección de destino de la ruta a guardar
Identificador	Modo
Descripción	Modo de realizar la ruta a guardar
FUNCIONES Y PROCEDIMIENTOS	
Signatura	DescargaArchivoTarea
Descripción	Tarea asíncrona que se encarga de la descarga de la ruta desde Internet

Tabla 136. Atributos, funciones y procedimientos clase XMLParser

6.5 Almacenamiento de datos

La aplicación, usa tres modelos de almacenamiento e interpretación de datos.

El primero, es el formato SQL, para almacenar tanto usuarios como rutas de favoritos.

El segundo, el formato XML de la base de datos de radares, donde la etiqueta que nos interesa de cada radar, es la etiqueta <coordinates>.

El tercero, es el formato TXT de las rutas que el usuario crea por sí mismo, en las cuales se separa cada toma de coordenadas mediante una nueva línea y mediante el separador ;; entre latitud y longitud.

A continuación, y por el orden anterior, se muestra una captura de cada uno de los modelos de almacenamiento.



_id	usuario	direccion_punto_inicial	direccion_punto_final	nombre_ruta	tipo_ruta
1	manu	Madrid	Córdoba	Madrid - Córdoba	walking
2	manu	Madrid	Sevilla	Madrid - Sevilla	driving
3	manu	Madrid	Barcelona	Madrid - Barcelona	driving
4	test	calle Guabairo	Plaza de España	calle Guabairo - Plaza de Espai	walking
5	test	Valdepeñas	Ciudad Real	Valdepeñas - Ciudad Real	driving
6	test	calle Guabairo	metro de Aluche	calle Guabairo - metro de Aluch	walking
7	uc3m	paseo de la Castellana	Plaza Castilla	paseo de la Castellana - Plaza	driving
8	uc3m	Valladolid	León	Valladolid - León	driving
9	uc3m	Plaza Mayor	Puerta del Sol	Plaza Mayor - Puerta del Sol	walking

Figura 43. Captura de base de datos FAVORITOS

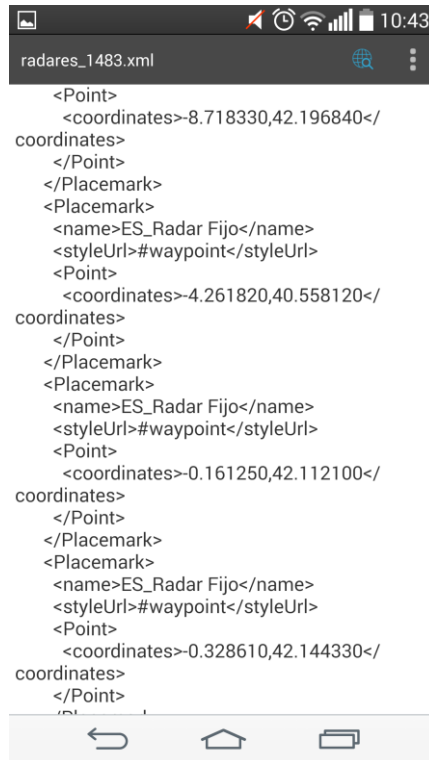


Figura 44. Captura de documento de radares

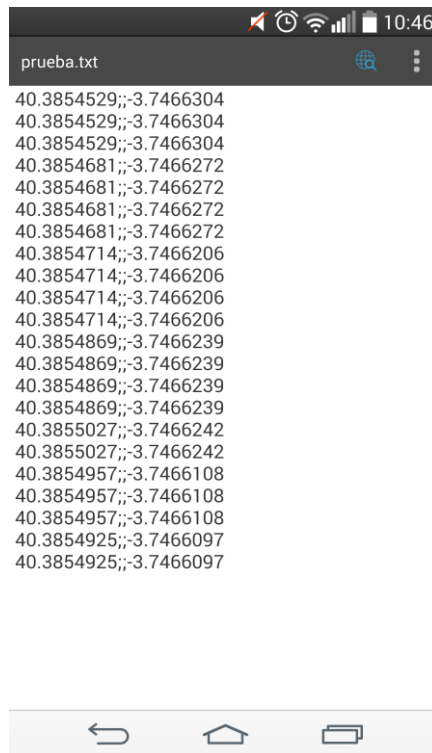


Figura 45. Captura de ruta generada libremente por el usuario

6.5.1. Modelado de bases de datos

En este apartado vamos a analizar las bases de datos en formato SQLite existentes en la aplicación. El resto de tipos de almacenamiento de datos no tienen necesidad de análisis al ser simples.

Existen dos bases de datos, de construcción similar, en las que se almacenan una serie de datos.

6.5.1.1. Base de datos de usuarios

Esta base de datos sirve para almacenar los diversos usuarios de la aplicación. Tiene dos tipos de variables almacenadas, por un lado las que introduce el usuario por pantalla y por otro lado las introducidas por sistema.

- Pantalla: Nombre de usuario, clave, e-mail (Introducidos mediante teclado)
- Sistema: Identificador (único, para evitar repeticiones)

6.5.1.2. Base de datos de favoritos

Esta base de datos sirve para almacenar las diversas rutas que el usuario almacene. Tiene dos tipos de variables almacenadas, por un lado las que introduce el usuario por pantalla y por otro lado las introducidas por sistema.

- Pantalla: Inicio, destino, modo (Introducidos mediante la voz)
- Sistema: Identificador (único, para evitar repeticiones), usuario, nombre ruta

En este caso el usuario viene dado desde que se guarda el dato al autenticarse en usuario en la aplicación. Por otro lado, la ruta se descarga y se almacena como nombre de ruta el mismo nombre que tiene el archivo .xml descargado.

6.5.1.3. Base de datos de notificaciones

Esta base de datos sirve para almacenar el dato temporal del momento de creación de una notificación para, a partir de él, eliminarla cuando la aplicación lo solicite.

6.5.1.4. Relación entre bases

Las bases no tienen relación entre ellas salvo en un momento, la eliminación de un usuario existente.

En este momento, desde la base de datos de usuarios se toma el nombre del mismo y con él se genera un cursor que se introduce en la base de datos de FAVORITOS y selecciona todas las entradas en las que el usuario coincide con el usuario tomado desde la base de datos de USUARIOS.

En los casos que existe coincidencia, también se elimina esa entrada en la base, así como los archivos vinculados a ella. Para entenderlo, se muestra un pequeño ejemplo:

El usuario "UC3M" tiene almacenada la ruta con inicio Madrid y destino Valencia. Datos a eliminar:

- Base USUARIOS: Entrada del usuario UC3M
- Base FAVORITOS: Entradas en las que la columna usuario sea igual a UC3M.
- Archivos: Archivos de rutas vinculadas a UC3M, en este caso "Madrid – Valencia.xml"

6.5.2. Diseño lógico de datos

Este punto contiene el modelo de datos para la aplicación de Gestor Encuestas. Se mostrará el Modelo Entidad-Relación (modelo conceptual), el modelo relacional (modelo lógico), el diagrama de tablas con sus relaciones, un listado de tablas y su finalidad dentro de la aplicación, y finalmente se detallará cada tabla con sus campos, claves, etc.

6.5.2.1. Modelo entidad/relación

Este tipo de modelos representan conceptualmente la información, mediante el cual se pretende 'visualizar' los objetos que pertenecen a la Base de Datos como entidades, las cuales tienen unos atributos y se vinculan mediante relaciones. Brevemente consiste en los siguientes pasos:

1. Se parte de una descripción textual del problema o sistema de información a automatizar (los requisitos).
2. Se hace una lista de los sustantivos y verbos que aparecen.
3. Los sustantivos son posibles entidades o atributos.
4. Los verbos son posibles relaciones.
5. Analizando las frases se determina la cardinalidad de las relaciones y otros detalles.
6. Se elabora el diagrama entidad-relación.
7. Se completa el modelo con listas de atributos y una descripción de otras restricciones que no se pueden reflejar en el diagrama.

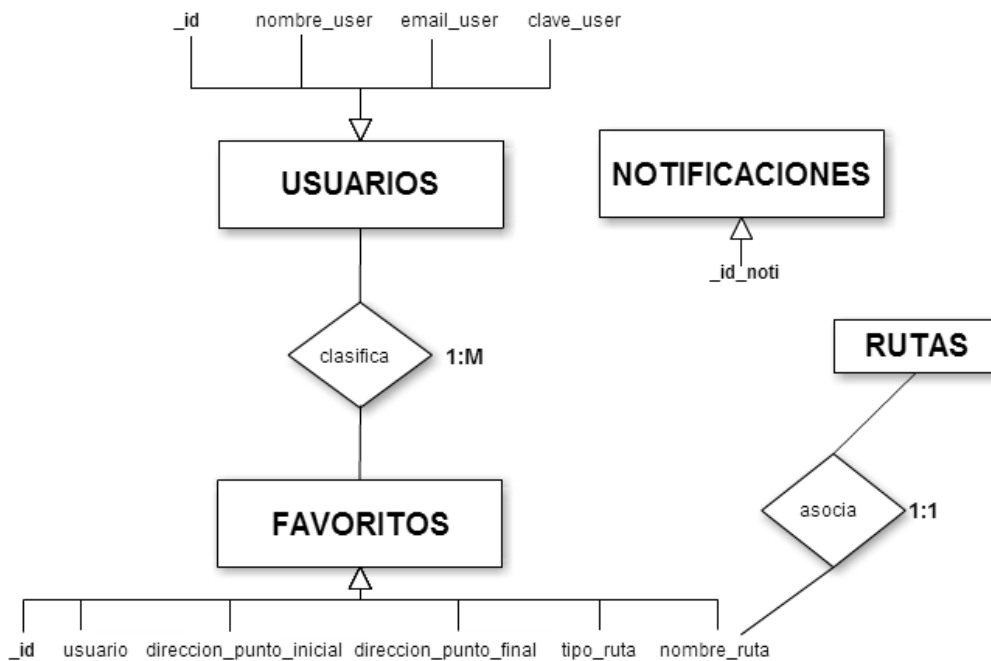


Figura 46. Modelo entidad/relación

6.5.2.2 Modelo relacional

Este modelo considera la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se puede denominar tupla o registro y a cada columna también se le puede llamar campo o atributo.

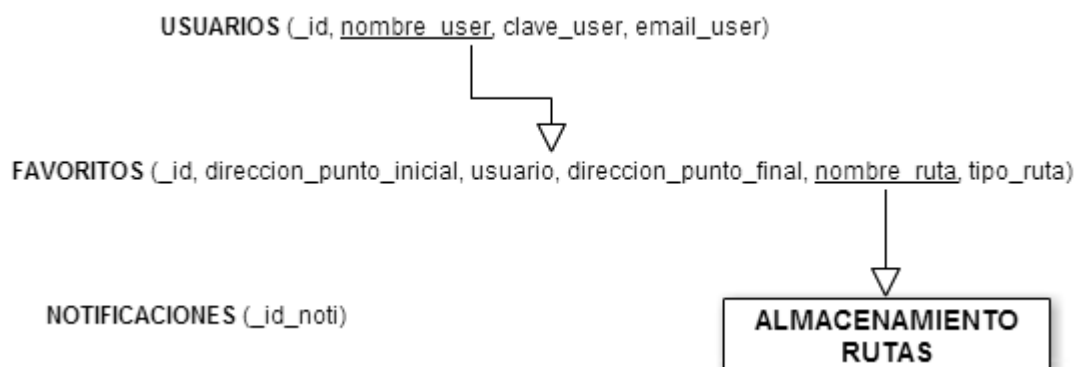


Figura 47. Modelo relacional

6.5.3. Diseño físico de datos

6.5.3.1. Modelo de tablas

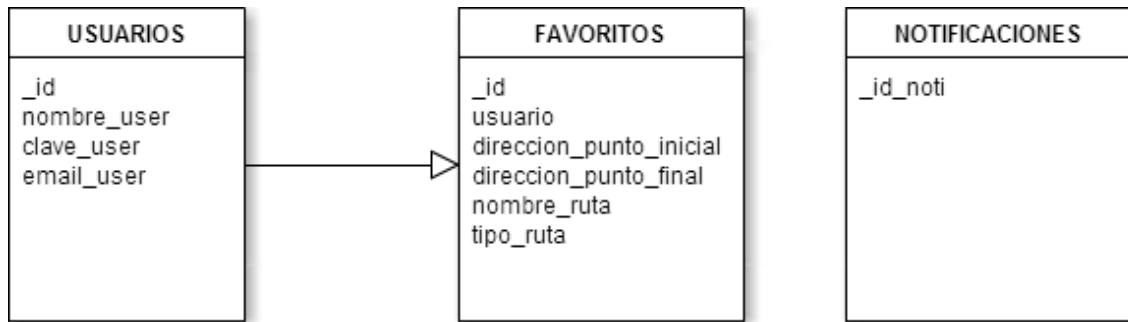


Figura 48. Modelo de tablas

6.5.3.2 Descripción de tablas

6.5.3.2.1 Tabla USUARIOS

NOMBRE	TIPO	OBLIGATORIO	SIGNIFICADO
_id	Integer primary key	SI	Identificador único del usuario
nombre_user	text	SI	Nombre del usuario
clave_user	text	SI	Contraseña del usuario
email_user	text	SI	Email del usuario

Tabla 137. Tabla USUARIOS

6.5.3.2.2. Tabla FAVORITOS

NOMBRE	TIPO	OBLIGATORIO	SIGNIFICADO
_id	Integer primary key	SI	Identificador único de la ruta
usuario	text	SI	Nombre del usuario que solicita la ruta
direccion_punto_inicial	text	SI	Inicio de la ruta
direccion_punto_final	text	SI	Final de la ruta
nombre_ruta	text	SI	Nombre para asociar con el archivo descargado
tipo_ruta	text	SI	Modo de ruta: Pie o vehículo

Tabla 138. Tabla FAVORITOS

6.5.3.2.2. Tabla NOTIFICACIONES

NOMBRE	TIPO	OBLIGATORIO	SIGNIFICADO
_id_noti	Integer	SI	Identificador único de la notificación

Tabla 139. Tabla NOTIFICACIONES

6.6 Verificación y aceptación de la arquitectura del sistema

El objetivo de la tarea documentada en este apartado es el de garantizar la calidad y viabilidad de las especificaciones del diseño del sistema de información antes de realizar el diseño detallado.

6.6.1 Análisis de consistencia de las especificaciones de diseño

Esta tarea pretende asegurar que las especificaciones de diseño realizadas hasta el momento son coherentes, comprobando que no son ambiguas, que no hay información duplicada y que todos los casos de uso quedan cubiertos con el diseño.

Estas comprobaciones se fundamentan en técnicas matriciales o de revisión entre los elementos comunes de los distintos modelos.

6.6.1.1 Trazabilidad casos de uso – clases de diseño

	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008	CU-009	CU-010	CU-011	CU-012	CU-013	CU-014	CU-015
AndroidXMLParsingActivity				X											
CalcularDistanciaPuntos				X		X								X	
Favoritos							X	X			X	X	X		
FavoritosCursorAdapter							X								
FavoritosDbAdapter							X								
FavoritosDbHelper							X								
FavoritosDescargaArchivo							X								
FinDeLaRuta									X	X					
GPSTracker				X	X	X		X	X	X					
Login			X												
Mail			X												
MainActivity				X							X				
MenuInicial															
ModoLibre									X	X			X		X
MostrarMapa				X		X		X				X	X		
Portada	X														
RutasHelper				X											
Usuarios			X										X		
UsuariosCursorAdapter			X												
UsuariosDbAdapter			X												
UsuariosDbHelper															
UsuariosFormulario			X												
XMLParser															

Tabla 140. Matriz de trazabilidad Clases de diseño - Casos de Uso

6.7 Especificación Técnica del Plan de Pruebas

En este apartado se realiza la especificación detallada de las diferentes pruebas a cumplimentar una vez generada la aplicación.

El plan de pruebas permitirá detectar errores y subsanarlos, así como comprobar que la aplicación cumple todas las funcionalidades requeridas, obteniendo así un sistema robusto y consistente.

El plan de pruebas completo ha sido agrupado en el punto Plan de Validación y Verificación del Software.

7. Plan de Validación y Verificación del Software

El proceso de pruebas de software tiene dos objetivos:

1. Para demostrar al desarrollador y al cliente que el software satisface requerimientos.
2. Para descubrir defectos en el comportamiento del software.

Cada una de las pruebas llevará asociada una descripción completa, explicando asimismo su finalidad, los pasos a seguir para su realización y los criterios para la aceptación de la prueba.

Una vez terminada cada prueba deberá ser rellenado el formulario propuesto en el presente documento indicando los resultados obtenidos y toda la información relativa a la misma que pueda resultar de interés para un futuro.

Teniendo como guía el presente documento, obtendremos un sistema verificado y validado, funcionando correctamente y asegurando la satisfacción del cliente debido el cumplimiento de su solicitud.

7.1. Definición de pruebas, entornos y metodología

Para poder desarrollar un sistema informático de calidad que funcione de manera correcta es imprescindible establecer un buen plan de pruebas.

En este apartado se definen los objetivos de las pruebas a realizar y cómo han de llevarse a cabo. Debe quedar recogido qué elementos van a ser probados, las características del sistema que se van a identificar, los elementos entregables, las actividades a realizar para poder construir una prueba, qué requisitos debe tener el sistema para que las pruebas se ejecuten con éxito y, por último, los criterios a tener en cuenta por el equipo de desarrollo para determinar si una prueba es finalizada con éxito o no.

Para cubrir el objetivo de elaborar un plan de pruebas detallado, se presentará los tipos de pruebas que serán necesarias para validar correctamente la implementación del sistema:

- **Pruebas unitarias:** Las pruebas unitarias tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente una vez que ha sido codificado.
- **Pruebas de integración:** Consiste en realizar pruebas para verificar que todas las partes del sistema están ensambladas e interaccionan correctamente.

- **Pruebas de aceptación:** El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento. Para que las pruebas definidas puedan ser ejecutadas de manera satisfactoria y produzcan los mismos resultados que producirían en el sistema real una vez implantado, es necesario definir las características que debe tener el sistema donde se van a realizar las pruebas.

En nuestro caso, el entorno de pruebas tendrá las mismas características que el entorno de implantación por lo que en este apartado no es necesario definir nuevas necesidades para el entorno.

7.1.1. Pruebas unitarias

Para la realización de estas pruebas, se ha utilizado el modo debug del programa con el que se ha realizado toda la aplicación, Eclipse.

Con este modo, podremos introducir puntos de corte en cualquier lugar de cada clase de la aplicación para comprobar si los valores heredados de otras clases son correctos o si los parámetros devueltos por cada método tienen un resultado positivo. En caso de hallar algún error, se revisaría la parte afectada del código.

De acuerdo con el análisis de clases realizado, estas pruebas se ejecutarán sobre las clases que permiten la entrada a las diferentes funcionalidades de la aplicación.

Solo se deberán hacer pruebas cuando haya lógica en la aplicación. Si únicamente se ejecuta un acceso a base de datos, no será necesario hacer pruebas unitarias de dicho componente.

7.1.2. Pruebas de integración y aceptación

En este apartado se especifican las pruebas que se deben realizar al sistema para comprobar su correcto funcionamiento y que cumple con todas las especificaciones.

A continuación se describe el formato de tabla utilizado para la especificación de las pruebas y los distintos campos que contiene:

- **Identificador:** Nombra de manera única a la prueba de aceptación siguiendo el formato PR-A-XXX, donde XXX tomará valores numéricos dentro del rango 000-999.
- **Descripción:** Especifica el objetivo de la prueba.

- **Precondiciones:** Describe el estado en el que se debe encontrar el sistema de forma previa a la realización de la prueba.
- **Pasos a seguir:** Establece la secuencia de pasos que debe realizar el responsable de la prueba sobre el sistema.

Los procedimientos de prueba establecidos para este proyecto se describen a continuación.

Identificador	PR - A - 001
Descripción	Crear un usuario
Precondiciones	1. Aplicación previamente iniciada
Pasos a seguir	1. Acceder a base de datos --> Crear usuario 2. Cumplir las condiciones para los diversos campos de la creación

Tabla 141. Prueba PR-A-001

Identificador	PR - A - 002
Descripción	Envío de e-mail de confirmación
Precondiciones	1. Usuario creado correctamente 2. E-mail insertado existente
Pasos a seguir	1. Creación usuario 2. Envío de e-mail a la cuenta proporcionada desde la cuenta propia de la aplicación con su información más relevante (usuario y clave)

Tabla 142. Prueba PR-A-002

Identificador	PR - A - 003
Descripción	Introducir un correcto nombre de usuario
Precondiciones	1. Estar en la interfaz FormularioUsuarios
Pasos a seguir	1. Introducir un usuario con una longitud entre 4 y 10 caracteres

Tabla 143. Prueba PR-A-003

Identificador	PR - A - 004
Descripción	Introducir un correcto email
Precondiciones	1. Estar en la interfaz FormularioUsuarios
Pasos a seguir	1. Introducir un email con caracteres "." y "@" al menos.

Tabla 144. Prueba PR-A-004

Identificador	PR - A - 005
Descripción	Introducir una correcta clave
Precondiciones	1. Estar en la interfaz FormularioUsuarios
Pasos a seguir	1. Introducir una clave con una longitud entre 4 y 10 caracteres y sin espacios

Tabla 145. Prueba PR-A-005

Identificador	PR - A - 006
Descripción	Editar datos de usuario
Precondiciones	1. Estar en la interfaz Usuarios 2. Hacer click largo en un usuario y seleccionar la opción "Editar"
Pasos a seguir	1. Cambiar los datos de acuerdo a las pruebas 003, 004 y 005 y pulsar aceptar.

Tabla 146. Prueba PR-A-006

Identificador	PR - A - 007
Descripción	Eliminar un usuario
Precondiciones	1. Estar en la interfaz Usuarios
Pasos a seguir	1. Hacer click largo en un usuario y seleccionar la opción "Eliminar" 2. Aceptar eliminación

Tabla 147. Prueba PR-A-007

Identificador	PR - A - 008
Descripción	Actualización de base de datos de radares
Precondiciones	1. Aplicación iniciada
Pasos a seguir	1. Pulsar botón de actualización. 2. Comprobar el número de radares cargados.

Tabla 148. Prueba PR-A-008

Identificador	PR - A - 009
Descripción	Comprobar activación de servicios de ubicación
Precondiciones	1. Aplicación iniciada. 2. Botón "Acceder a App" en color rojo y botón con la imagen del marcador, color azul
Pasos a seguir	1. Pulsar marcador. 2. Activar servicios ubicación.

Tabla 149. Prueba PR-A-009

Identificador	PR - A - 010
Descripción	Autenticación en la aplicación
Precondiciones	<ol style="list-style-type: none"> 1. Tener servicios de ubicación activados 2. Pulsar botón acceso a App
Pasos a seguir	<ol style="list-style-type: none"> 1. Introducir nombre y clave correctos 2. Pulsar aceptar

Tabla 150. Prueba PR-A-010

Identificador	PR - A - 011
Descripción	Cargar base de datos de radares
Precondiciones	<ol style="list-style-type: none"> 1. Aplicación iniciada
Pasos a seguir	<ol style="list-style-type: none"> 1. Pulsar botón Acceso App 2. Comprobar número de radares cargados

Tabla 151. Prueba PR-A-011

Identificador	PR - A - 012
Descripción	Cambiar modo de ruta
Precondiciones	<ol style="list-style-type: none"> 1. Usuario autenticado 2. Estar en la interfaz MainActivity
Pasos a seguir	<p>Dos combinaciones:</p> <ol style="list-style-type: none"> 1. Botón persona rojo, botón vehículo azul. Ruta en vehículo 2. Botón persona azul, botón vehículo rojo. Ruta a pie. 3. Pulsar el botón del modo deseado.

Tabla 152. Prueba PR-A-012

Identificador	PR - A - 013
Descripción	Crear favorito
Precondiciones	<ol style="list-style-type: none"> 1. Aplicación iniciada 2. Pulsar botón favorito o comando de voz "Favoritos"
Pasos a seguir	<ol style="list-style-type: none"> 1. Pulsar añadir nuevo favorito 2. Dar ruta con formato "Ir de" + punto A + "a" + punto B 3. Dar modo "A pie" ó "Vehículo"

Tabla 153. Prueba PR-A-013

Identificador	PR - A - 014
Descripción	Descargar ruta en formato .XML desde Internet
Precondiciones	<ol style="list-style-type: none"> 1. Estar en interfaz "Favoritos"
Pasos a seguir	<ol style="list-style-type: none"> 1. Ruta con formato correcto (ver PR - A - 013) 2. Tener conexión de datos activa (red o WiFi) 3. Existencia de la carpeta destino (o creación)

Tabla 154. Prueba PR-A-014

Identificador	PR - A - 015
Descripción	Eliminar ruta favoritos
Precondiciones	<ol style="list-style-type: none"> 1. Estar en la interfaz Favoritos 2. Ruta previamente guardada
Pasos a seguir	<ol style="list-style-type: none"> 1. Click largo sobre el nombre de la ruta a eliminar 2. Pulsar aceptar.

Tabla 155. Prueba PR-A-015

Identificador	PR - A - 016
Descripción	Iniciar ruta desde favoritos
Precondiciones	<ol style="list-style-type: none"> 1. Estar en la interfaz Favoritos 2. Ruta previamente guardada
Pasos a seguir	<ol style="list-style-type: none"> 1. Click largo sobre el nombre de la ruta a iniciar 2. Pulsar iniciar 3. Esperar a que la el sistema lea el documento, cargue los datos importantes y dibuje el mapa

Tabla 156. . Prueba PR-A-016

Identificador	PR - A - 017
Descripción	Llamar a un contacto
Precondiciones	1. Estar en la interfaz MainActivity
Pasos a seguir	<ol style="list-style-type: none"> 1. Pulsar botón del micrófono. 2. Decir el comando "llamar" 3. Esperar respuesta y decir el nombre de un contacto de la agenda. 4. Esperar respuesta y decir "Si" para que inicie la llamada

Tabla 157. Prueba PR-A-017

Identificador	PR - A - 018
Descripción	Enviar SMS a contacto
Precondiciones	1. Estar en la interfaz MainActivity
Pasos a seguir	<ol style="list-style-type: none"> 1. Pulsar botón del micrófono 2. Decir el comando "enviar SMS" 3. Esperar respuesta y decir el nombre de un contacto de la agenda. 4. Esperar respuesta y decir el contenido del SMS 5. Esperar respuesta y decir "Si" para que envíe el SMS.

Tabla 158. Prueba PR-A-018

Identificador	PR - A - 019
Descripción	Acceder al Modo Libre
Precondiciones	1. Estar en la interfaz MainActivity
Pasos a seguir	Pulsar botón de micrófono. Decir el comando "Modo libre"

Tabla 159. Prueba PR-A-019

Identificador	PR - A - 020
Descripción	Iniciar ruta libre
Precondiciones	<ol style="list-style-type: none"> 1. Estar en la interfaz ModoLibre 2. Tener activada ubicación del dispositivo
Pasos a seguir	<ol style="list-style-type: none"> 1. Pulsar botón empezar 2. El dispositivo comienza a calcular el tiempo transcurrido y los lugares visitados

Tabla 160. Prueba PR-A-020

Identificador	PR - A - 021
Descripción	Finalizar ruta libre
Precondiciones	<ol style="list-style-type: none"> 1. Estar en la interfaz ModoLibre 2. Haber pulsado con anterioridad el botón empezar
Pasos a seguir	<p>Dos posibilidades</p> <ol style="list-style-type: none"> 1. Pulsar botón parar y aceptar cualquiera de las dos posibilidades. 2. Pulsar botón back y posteriormente hacer click en "Si".

Tabla 161. Prueba PR-A-021

Identificador	PR - A - 022
Descripción	Crear archivo provisional con todas las ubicaciones de una ruta libre.
Precondiciones	<ol style="list-style-type: none"> 1. Estar en la interfaz modo libre 2. Pulsar botón empezar
Pasos a seguir	<ol style="list-style-type: none"> 1. El fichero se crea en el instante que se pulsa el botón y responde al nombre de "dataprueba_int.txt"

Tabla 162. Prueba PR-A-022

Identificador	PR - A - 023
Descripción	Eliminar fichero provisional con ubicaciones.
Precondiciones	<ol style="list-style-type: none"> 1. Estar en la interfaz ModoLibre 2. Ruta libre empezada
Pasos a seguir	<p>Varios métodos para eliminar el archivo.</p> <ol style="list-style-type: none"> 1. Finalizar la ruta mediante pulsado botón back 2. Finalizar ruta mediante vuelta al menú (a través botón parar) 3. Almacenar/No almacenar ruta a través de interfaz FinDeLaRuta

Tabla 163. Prueba PR-A-023

Identificador	PR - A - 024
Descripción	Crear fichero con el tiempo que tarda el usuario en realizar una ruta libre
Precondiciones	<ol style="list-style-type: none"> 1. Ruta libre iniciada previamente
Pasos a seguir	<ol style="list-style-type: none"> 1. Finalizar la ruta pulsando el botón parar y escogiendo opción "Resumen". 2. Se crea ubicación automática y temporal "dataprueba_crono.txt"

Tabla 164. Prueba PR-A-024

Identificador	PR - A - 025
Descripción	Eliminar fichero con el tiempo.
Precondiciones	1. Ruta libre finalizada
Pasos a seguir	1. Almacenar ruta libre 2. Volver al menú principal

Tabla 165. Prueba PR-A-025

Identificador	PR - A - 026
Descripción	Mostrar resumen con datos de la ruta libre
Precondiciones	1. Finalizar ruta y acceder al Resumen de la misma
Pasos a seguir	1. El informe con tiempo, distancia, velocidad media y dibujo de la ruta realizada se mostrará automáticamente en la interfaz FinDeLaRuta.

Tabla 166. Prueba PR-A-026

Identificador	PR - A - 027
Descripción	Guardar ruta libre
Precondiciones	1. Estar en la interfaz FinDeLaRuta
Pasos a seguir	1. Pulsar botón guardar. 2. Dar un nombre a la ruta no existente. 3. Se generan dos archivos "ruta.txt" y "ruta_crono.txt"

Tabla 167. Prueba PR-A-027

Identificador	PR - A - 028
Descripción	Eliminar ruta libre
Precondiciones	1. Estar en la interfaz ModoLibre
Pasos a seguir	<ol style="list-style-type: none"> 1. Pulsar botón cargar. 2. Seleccionar una ruta mediante click largo 3. Pulsar "Si" en el cuadro" 4. Se borrarán los archivos "ruta.txt" y "ruta_crono.txt"

Tabla 168. Prueba PR-A-028

Identificador	PR - A - 029
Descripción	Cargar ruta previamente guardada
Precondiciones	1. Estar en la interfaz ModoLibre
Pasos a seguir	<ol style="list-style-type: none"> 1. Pulsar botón cargar. 2. Realizar click corto en la ruta deseada

Tabla 169. Prueba PR-A-029

Identificador	PR - A - 030
Descripción	Iniciar modo "Lucha contra ti mismo"
Precondiciones	1. Estar en interfaz ModoLibre
Pasos a seguir	<ol style="list-style-type: none"> 1. Pulsar botón cargar 2. Iniciar una ruta de las disponibles 3. Estar a menos de 50 metros del punto inicial 4. Aceptar y esperar los 5 segundos de cuenta atrás

Tabla 170. Prueba PR-A-030

Identificador	PR - A - 031
Descripción	Definición checkpoints modo “Lucha contra ti mismo”
Precondiciones	1. Cargar una ruta en la interfaz ModoLibre
Pasos a seguir	<ol style="list-style-type: none"> 1. El sistema cargará el archivo “ruta.txt” con todas las coordenadas y escogerá un par cada determinado número. 2. Se pondrá un marcador en el mapa. 3. Hay que pasar a menos de 50 metros de ella para poder continuar al siguiente.

Tabla 171. Prueba PR-A-031

Identificador	PR - A - 032
Descripción	Activar cuenta atrás modo “Lucha contra ti mismo”
Precondiciones	1. Cargar una ruta en la interfaz ModoLibre
Pasos a seguir	<ol style="list-style-type: none"> 1. El sistema cargará el archivo “ruta_crono.txt” y cargará el valor que almacena (número de segundos) 2. Se realiza la transformación a horas, minutos, segundos y se muestra por pantalla el decrecimiento de la misma.

Tabla 172. Prueba PR-A-032

Identificador	PR - A - 033
Descripción	Actualización de tiempo de ruta
Precondiciones	1. Cargar una ruta en la interfaz ModoLibre
Pasos a seguir	<ol style="list-style-type: none"> 1. Finalizar la ruta en menor tiempo del disponible. 2. El fichero “ruta_crono.txt” se actualizará con el tiempo nuevo tiempo realizado

Tabla 173. Prueba PR-A-033

Identificador	PR - A - 034
Descripción	Crear ruta desde ubicación actual con Google Maps
Precondiciones	1. Estar en la interfaz MainActivity
Pasos a seguir	<ol style="list-style-type: none"> 1. Pulsar botón micrófono 2. Decir comando "Ir a... " más destino. 3. El sistema generará la ruta a partir de la ubicación actual del dispositivo.

Tabla 174. Prueba PR-A-034

Identificador	PR - A - 035
Descripción	Crear ruta entre dos puntos a través de Google Maps
Precondiciones	1. Estar en la interfaz MainActivity
Pasos a seguir	<ol style="list-style-type: none"> 1. Pulsar botón micrófono 2. Decir comando "Ir de... " más inicio más "a" más destino. 3. El sistema generará la ruta a partir de la ubicación inicial dicha por el usuario.

Tabla 175. Prueba PR-A-035

Identificador	PR - A - 036
Descripción	Cargar radares
Precondiciones	1. Haber generado una ruta en vehículo a través de Google Maps
Pasos a seguir	1. El sistema cargará la base automáticamente tras cargar la ruta.

Tabla 176. Prueba PR-A-036

Identificador	PR - A - 037
Descripción	Detectar radar cercano
Precondiciones	1. Haber generado una ruta en vehículo a través de Google Maps
Pasos a seguir	<ol style="list-style-type: none"> 1. Acercarse a menos de 500 metros de un radar 2. Aparecerá un marcador en pantalla del mismo. 3. Se emitirá un mensaje sonoro para el usuario "Cuidado, radar a 500 metros".

Tabla 177. Prueba PR-A-037

Identificador	PR - A - 038
Descripción	Avisar de punto de ruta cercano
Precondiciones	1. Haber generado una ruta a través de Google Maps
Pasos a seguir	<ol style="list-style-type: none"> 1. Acercarse a la distancia de aviso de un punto de ruta (depende de aspectos como la longitud de ese recorrido) 2. Se dará un aviso sonoro al usuario con la instrucción a realizar.

Tabla 178. Prueba PR-A-038

Identificador	PR - A - 039
Descripción	Actualizar ruta
Precondiciones	1. Haber generado una ruta a través de Google Maps
Pasos a seguir	<ol style="list-style-type: none"> 1. Alejarse del siguiente punto más del máximo establecido (variable como en 038, según distancia) 2. Se toma la posición actual y se genera una nueva ruta con el mismo destino.

Tabla 179. Prueba PR-A-039

Identificador	PR - A - 040
Descripción	Finalizar ruta
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado una ruta a través de Google Maps 2. Llegar al último punto
Pasos a seguir	<ol style="list-style-type: none"> 1. Aparecerá un cuadro avisando de que hemos finalizado el recorrido.

Tabla 180. Prueba PR-A-040

Como conclusión, indicar que el resultado de todas las pruebas tras haberlas realizado por lo menos en 5 ocasiones, es correcto.

7.1.3 Matriz de trazabilidad Pruebas Aceptación/Requisitos de Capacidad

	RU-C-001	RU-C-002	RU-C-003	RU-C-004	RU-C-005	RU-C-006	RU-C-007	RU-C-008	RU-C-009	RU-C-010	RU-C-011	RU-C-012	RU-C-013	RU-C-014	RU-C-015	RU-C-016
PR-A-001	X											X				
PR-A-002												X				
PR-A-003												X				
PR-A-004												X				
PR-A-005												X				
PR-A-006												X				X
PR-A-007									X			X				X
PR-A-008		X											X			X
PR-A-009														X		
PR-A-010												X				
PR-A-011		X													X	
PR-A-012				X							X					
PR-A-013	X				X	X		X								
PR-A-014					X	X	X	X							X	
PR-A-015									X							X
PR-A-016		X	X	X							X			X		
PR-A-017					X											

PR-A-018				X															
PR-A-019				X															
PR-A-020			X			X	X			X					X				
PR-A-021																			X
PR-A-022			X			X	X			X					X				X
PR-A-023														X					
PR-A-024						X	X			X									
PR-A-025														X					
PR-A-026				X							X			X			X		
PR-A-027							X			X									X
PR-A-028									X										
PR-A-029			X	X							X								
PR-A-030			X	X							X				X				
PR-A-031				X							X								X
PR-A-032																			X
PR-A-033																			X
PR-A-034		X	X	X	X						X				X				
PR-A-035		X		X	X						X								
PR-A-036		X																	
PR-A-037			X	X								X				X			
PR-A-038			X													X			
PR-A-039			X	X								X				X			
PR-A-040			X													X			

Tabla 181. Matriz de trazabilidad Pruebas Aceptación – Requisitos de capacidad

7.2 Definición de las Pruebas Unitarias

Dado que las pruebas unitarias están muy ligadas a la codificación y debido a su amplitud no se detallarán en la documentación.

7.2.1 Definición de las pruebas de implantación

Las pruebas para comprobar que se ha implantado correctamente el sistema se centran en aquellas que son vitales para el usuario, es por ello que las pruebas que serán necesarias pasar obligatoriamente para comprobar el correcto funcionamiento del aplicativo un vez implantado serán las pruebas de Aceptación.

7.2.2 Definición de las pruebas de Integración

A continuación se describe el formato de tabla utilizado para la especificación de las pruebas y los distintos campos que contiene:

- **Identificador:** Nombra de manera única a la prueba de aceptación siguiendo el formato PR-I-XXX, donde XXX tomará valores numéricos dentro del rango 000-999.
- **Descripción:** Especifica el objetivo de la prueba.
- **Precondiciones:** Describe el estado en el que se debe encontrar el sistema de forma previa a la realización de la prueba.
- **Pasos a seguir:** Establece la secuencia de pasos que debe realizar el responsable de la prueba sobre el sistema.

Los procedimientos de prueba establecidos para este proyecto se describen a continuación:

Identificador	PR - I - 001
Descripción	Adaptar la aplicación al dispositivo LG G3
Precondiciones	
Pasos a seguir	Se comprueba que las interfaces están bien definidas para el tamaño de la pantalla.

Tabla 182. Prueba PR-I-001

Identificador	PR - I - 002
Descripción	Interacción Google Maps con aplicación
Precondiciones	
Pasos a seguir	Se comprueba que están tratados todos los posibles fallos en las rutas (lugares inexistentes, rutas imposibles de realizar en vehículo terrestre...)

Tabla 183. Prueba PR-I-002

8. Conclusión

En este apartado se exponen las conclusiones extraídas durante el desarrollo del proyecto fin de carrera, así como indicaciones sobre futuras líneas de actuación en la ampliación del sistema gestión de información.

8.1. Conclusiones del proyecto

La elaboración de este proyecto ha proporcionado una solución eficaz para todos los problemas planteados al inicio del mismo, creando una aplicación robusta pero susceptible a futuras ampliaciones, dotada de una interfaz intuitiva y fácil de manejar.

Ha resultado muy beneficioso el haber seguido una adaptación de la metodología Métrica Versión 3, ya que pese al amplio número de fases que la componen, obtención de requisitos, diseño de datos, elección de las tecnologías de desarrollo, estimaciones de tiempo etc. ha permitiendo un desarrollo tanto del sistema en sí, como su documentación, con un nivel de calidad bastante satisfactorio.

Además de proporcionar importantes conocimientos de programación en lenguaje Java que complementan a los ya adquiridos durante la carrera; y además sobre el mundo de la geolocalización.

Con respecto al empleo del software Eclipse para esta arquitectura, aspecto en el que no era un principiante pero tampoco un usuario experto, he de decir que en un principio resultó engorroso el amplio número de actuaciones en el código que se lleva a cabo para completar cualquier acción, pero a medida que la aplicación crecía en tamaño, complejidad y funcionalidad resultaba una determinante y útil herramienta capaz de estructurar y facilitar el control de la misma.

En cuanto a la estimación de tiempos elaborada inicialmente, se cumplieron casi todos los hitos a tiempo, salvo algunos de ellos debidos a algunos problemas hardware. Pero pese a ellos se ha conseguido cumplir los plazos de entrega.

De forma general, es posible afirmar que los objetivos marcados han sido completados de forma más o menos satisfactoria en la medida que los medios y el tiempo dedicado lo ha permitido

8.2. Ampliaciones

A continuación se plantearán algunas ampliaciones futuras y mejoras del gestor de rutas a través de la herramienta de Google Maps, con el objetivo de aportar algunas ideas sobre qué y cómo realizar un desarrollo más completo. Se va a realizar dos apartados sobre ampliaciones,

ampliaciones con necesidad de capital o ampliaciones que podrían haberse realizado sin necesidad de aporte económico (es decir, solo trabajo del equipo de desarrollo).

8.2.1. Ampliaciones con necesidad de capital

A continuación se van a enumerar distintas mejoras futuras en el caso de contar con un flujo de capital exterior:

- **Licencia de Google Maps:** Mayor número de hitos e instrucciones, eliminar restricciones de solicitudes diarias. También se adquiriría un servicio técnico por parte de Google. Es un paso obligado para poder llevar la aplicación a la Play Store de Android. Precio aproximado de 8500 euros (precio de verano de 2013).
- **Alojamiento en servidor web:** Si la aplicación estuviera en Internet alojada, se podrían alojar todas las bases de datos allí además de habilitar el compartimento de contenidos entre los usuarios (Por ejemplo rutas de favoritos)
- **Bases de radares más precisas:** Se pueden adquirir bases de datos con mayor número de parámetros como la velocidad máxima mediante el pago de una licencia (Ver documentación).
- **Mejora de interfaces:** Se podría encargar que las interfaces las realizara un experto en diseño gráfico.

8.2.2. Ampliaciones sin necesidad de capital

A continuación se van a enumerar una serie de mejoras futuras a realizar sin necesidad de una aportación de capital externo.

- **Vinculación a redes sociales:** Capacidad de poder compartir rutas mediante las diversas redes sociales o registro mediante las mismas (Google Plus, Facebook, Twitter).
- **Aumentar funciones orales:** Implementar nuevas funciones orales como el uso de cámara o linterna a través de comandos de voz desde la aplicación
- **Aumento del reporte de estadísticas:** Implementar un informe con mayor número de estadísticas de un viaje que el reportado en esta versión (tiempo, distancia, velocidad).
- **Ampliar idiomas:** Generar ficheros de varios idiomas (Inglés, francés, alemán...) para que se detecte el idioma del teléfono automáticamente y la aplicación se inicie en dicho idioma por defecto.
- **Servicios básicos de salud:** Implementar en el registro un pequeño cuestionario para el usuario (altura, peso, edad, sexo) para poder generar en la aplicación cálculos sobre el gasto calórico del usuario. Así mismo añadir una nueva base de datos alimenticia para generar dietas gastronómicas a partir de un objetivo dado por el usuario (Ejemplo: 1500 kilocalorías diarias)

9. Manual de usuario

9.1. Acceso a la aplicación

Para acceder a la aplicación es indispensable disponer de un Smartphone con sistema operativo Android. Por otro lado, en determinados momentos, es necesaria una conexión de datos.

Para todo el manual se adjuntaran imágenes del botón que haya que pulsar para la realización de la acción solicitada en la que estemos situados en cada momento.

9.1.1 Acceder a la aplicación

Para acceder a la aplicación tan solo se debe realizar un click en el botón de inicio de la aplicación:



Figura 49. Botón inicio aplicación

9.1.2. Menú inicial



Figura 50. Menú inicial

Esta es la interfaz inicial de la aplicación, que aparece tras los 5 segundos de presentación de la misma. En ella nos encontramos una serie de botones para realizar determinadas acciones.

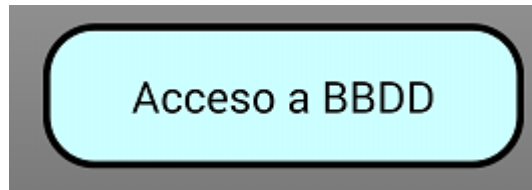


Figura 51. Botón de acceso a base de datos de usuarios

Este botón se usará para acceder a todo el manejo de la base de datos de usuarios. Su creación, edición y borrado.

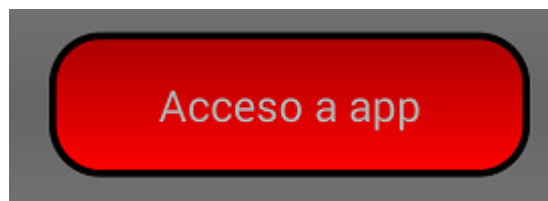


Figura 52. Botón de acceso a la aplicación

Este botón servirá para acceder a toda la aplicación de geolocalización mediante voz. Tiene dos posibles estados:

- **Color azul:** significa que los servicios de localización se encuentran activos por lo cual podrá accederse al resto de la aplicación.
- **Color rojo:** significa que los servicios de localización se encuentran desactivados por lo cual se deben activar previamente.

En este caso, vemos que los servicios de localización están desactivados.



Figura 53. Botón actualizar radares

Este botón consultará si la base de datos de radares está en su última actualización o puede actualizarse.

Si la base se actualiza, escucharemos el mensaje “Se ha actualizado la base de datos de radares”.

En el caso contrario, si nuestra versión es la última, el mensaje que oiremos será “La base de datos de radares está actualizada”.



Figura 54. Botón acceso servicios ubicación

Este botón sirve para acceder a los servicios de ubicación. Al igual que el botón de acceso a la aplicación, tendrá dos versiones:

- **Botón azul:** Los servicios de ubicación no están activos por lo que podremos acceder a través de la pulsación de este botón.
- **Botón rojo:** Los servicios de ubicación están activos por lo que no podremos hacer uso de este botón.

En el caso del ejemplo, los servicios de ubicación no están escasos.

9.1.3 Base de datos de usuarios

Para acceder a esta zona de la aplicación, debemos haber pulsado el botón de acceso a la app, tal como se ha indicado en el anterior punto.

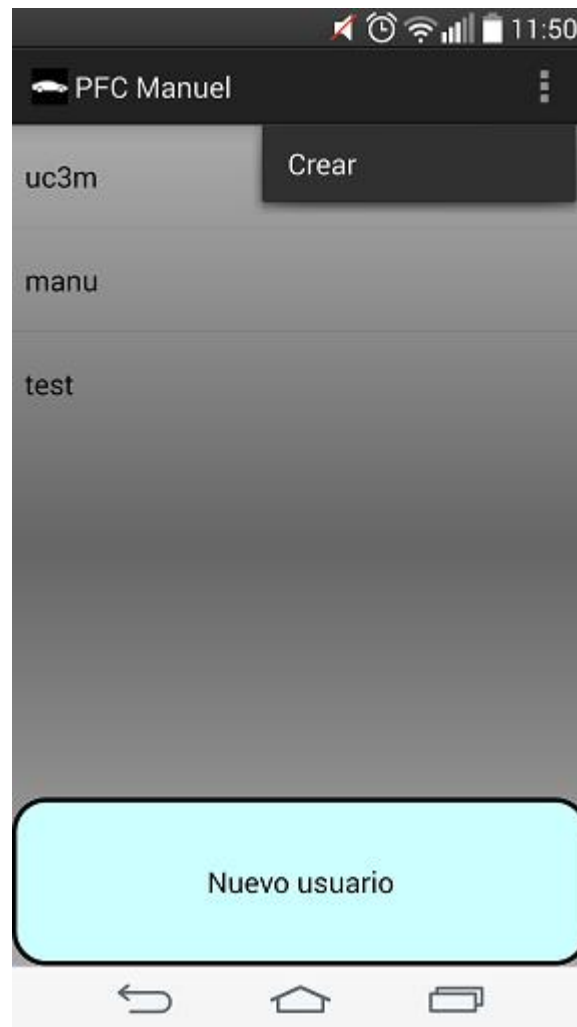


Figura 55. Interfaz lista de usuarios

Desde esta interfaz podemos consultar la lista de usuarios existentes y realizar diversas modificaciones en ellos, así como poder crear nuevos usuarios. En caso de no existir ninguno aparece la etiqueta “Sin datos”.

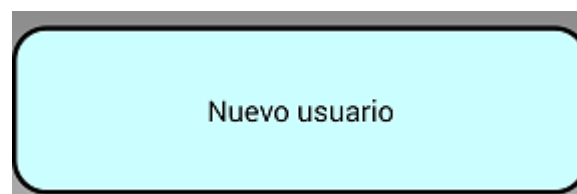


Figura 56. Botón de creación de usuario

Pulsando en este botón se accederá a la interfaz de creación de usuario. Cabe decir que existe otro método el cual se muestra en la figura **INTERFAZ LISTA DE USUARIOS** que al pulsar en los tres puntos de la esquina superior nos da la opción de crear usuario del mismo modo.

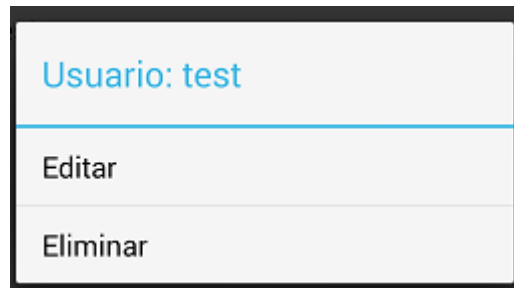


Figura 57. Opciones click largo usuario

Si hacemos un click largo en cualquier usuario (en este ejemplo “test”), se mostrará por pantalla un cuadro con las opciones a desarrollar, en este caso eliminar y editar.

En el caso de pulsar sobre la etiqueta Editar, accederemos a la figura 59 mostrada más adelante.

En el caso de pulsar el botón eliminar, se nos mostrará otro cuadro de aviso, para tener que aceptar en dos ocasiones y no eliminar usuarios de manera errónea.

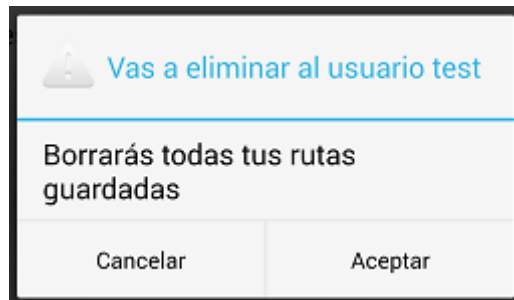


Figura 58. Confirmación borrar usuarios

Se muestra un aviso de que se eliminarán las rutas de favoritas asociadas a dicho usuario. Si se desea eliminar definitivamente, se debe pulsar el botón “Aceptar” y en caso contrario “Cancelar”.

9.1.4. Creación y edición de usuarios



Figura 59. Edición y creación de usuarios

En esta imagen recogemos tanto el modo edición como creación al usar ambos la misma interfaz de base. En el primer caso, vemos como los datos están ya rellenos mientras que en el segundo están vacíos.

Existen una serie de cláusulas para la creación:

- Usuario: Entre 4 y 10 caracteres y no puede estar repetido.
- Contraseña: Entre 4 y 10 caracteres y sin espacios
- Email: Debe contener los caracteres "@" y "."



Figura 60. Botón editar usuarios

Como bien indica su etiqueta, para modificar datos de un usuario.

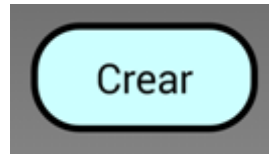


Figura 61. Botón crear usuario

Como bien indica su nombre, botón para crear usuarios. El usuario creado pasará a formar parte de la lista tal y como se muestra en la figura 55.

9.1.5. Autenticación de usuarios



Figura 62. Interfaz de autenticación de usuarios

Pantalla en la cual se solicita introducir los datos de usuario para acceder a la aplicación.



Figura 63. Botón acceso aplicación

Tras introducir los datos de manera correcta, el usuario debe hacer click en este botón para así acceder a la parte privada de la aplicación.

9.1.6. Menú principal



Figura 64. Interfaz menú principal

Esta es la interfaz principal de la aplicación, desde la cual accederemos a todas las funcionalidades relacionadas con la geolocalización. Explicamos todas las posibilidades a continuación:



Figura 65. Botones de modo de ruta

Estos botones sirven para configurar el modo en el que el usuario quiere desarrollar la ruta. El código de colores nos indica en qué estado se encuentra dicho parámetro. Por defecto, la combinación es la que se muestra, es decir, el modo que viene de serie es el de vehículo:

- Modo vehículo: Botón vehículo azul, botón persona rojo.
- Modo a pie: Botón vehículo rojo, botón persona azul



Figura 66. Botón micrófono

El botón clave de la aplicación desde el cual dirigiremos todas las principales operaciones a realizar. Al pulsarlo, aparecerá un cuadro principal e irá recogiendo aquello que digamos para así realizar la operación solicitada. Las siguientes sentencias desencadenarán una respuesta por parte del dispositivo o directamente una acción.

- **Favoritos:** Con esta palabra accederemos a la sección de favoritos al igual que con el botón mostraremos tras esta lista.
- **Llamar:** Con esta palabra generamos una conversación con nuestro dispositivo, cuya respuesta será “**¿A quién deseas llamar?**”. Se debe indicar el nombre de un contacto de la agenda (Para el ejemplo, usaremos a “Mamá”). Por último es móvil nos preguntará si estamos seguros para realizar la llamada. Se transcribe una conversación ejemplo con el usuario con letra normal y el dispositivo en negrita.

-Llamar

-¿A quién deseas llamar?

-Mamá

-¿Deseas llamar a Mamá?

-Sí

- **Enviar SMS:** Parecido al funcionamiento de una llamada, solo que además deberemos de dar al dispositivo el contenido del SMS. A continuación se vuelve a transcribir la conversación, usando otra vez a “Mamá” como contacto al que deseamos escribir y “¿Qué hora es?” como mensaje a enviar.

-Enviar SMS

-¿A quién deseas enviarlo?

-Mamá

-Dime el mensaje para Mamá

-¿Qué hora es?

-El contenido del SMS es ¿Qué hora es?. ¿Deseas enviarlo?

-Si

-Mensaje enviado.

- **Ir a... destino/ Ir al... destino:** Para realizar una ruta desde el punto actual en el que el usuario se encuentra. Ejemplo, si estás en la Puerta del Sol y tu frase es “Ir al Paseo de la Castellana” la ruta se calculará entre ambos puntos.

- **Ir de inicio a destino:** Para realizar una ruta desde cualquier punto, el usuario debe decir al dispositivo tanto inicio como destino. Por ejemplo, si queremos ir de Madrid a Valencia, diremos “Ir de Madrid a Valencia”. En este caso el destino escogerá los puntos céntricos de dichas poblaciones y nos realizará dicha ruta.

- **Modo libre:** Estas dos palabras nos harán acceder al modo libre de la aplicación.



Figura 67. Botón de acceso a favoritos

Al igual que se ha comentado en las posibilidades mediante voz, también se facilita este botón para acceder a la sección de favoritos.

9.1.7. Favoritos

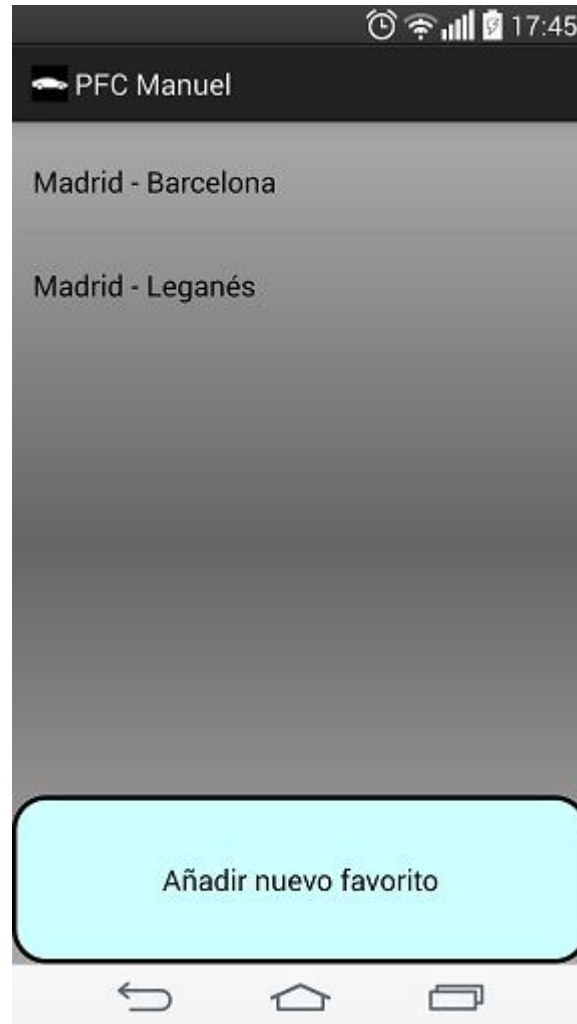


Figura 68. Interfaz Favoritos

En esta interfaz, parecida a las interfaces de manejo de usuario, se nos muestra una lista de las rutas almacenadas (En el caso de no existir ninguna, arriba aparece la etiqueta “Sin datos”).

Si realizamos click largo en alguna ruta, se nos mostrará por pantalla el siguiente cuadro con opciones:

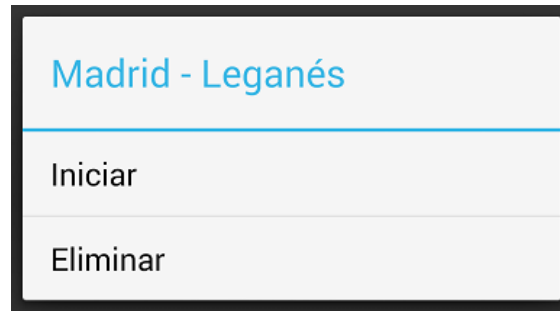


Figura 69. Opciones ruta favoritos

Las opciones son las dos claramente indicadas, la primera nos envía a realizar la ruta a través de lo indicado en la ruta. La segunda elimina el registro de la base de datos así como el documento almacenado de la propia ruta.

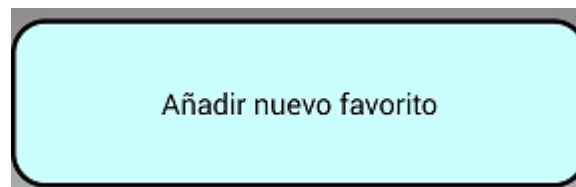


Figura 70. Botón nuevo favorito

Al hacer click sobre este botón ocurrirá al igual que en el botón micrófono (figura 66). En este caso, el diálogo será el siguiente. Usaremos como ejemplo una ruta Madrid –Valencia. En este modo solo se podrán almacenar rutas desde un punto conocido.

- Ir de Madrid a Valencia
- **¿Deseas guardar la ruta a pie o en vehículo?**
 - A pie (Si se desea realizar la ruta andando)
 - Vehículo (Si se desea en un vehículo)

9.1.8 Seguimiento de ruta

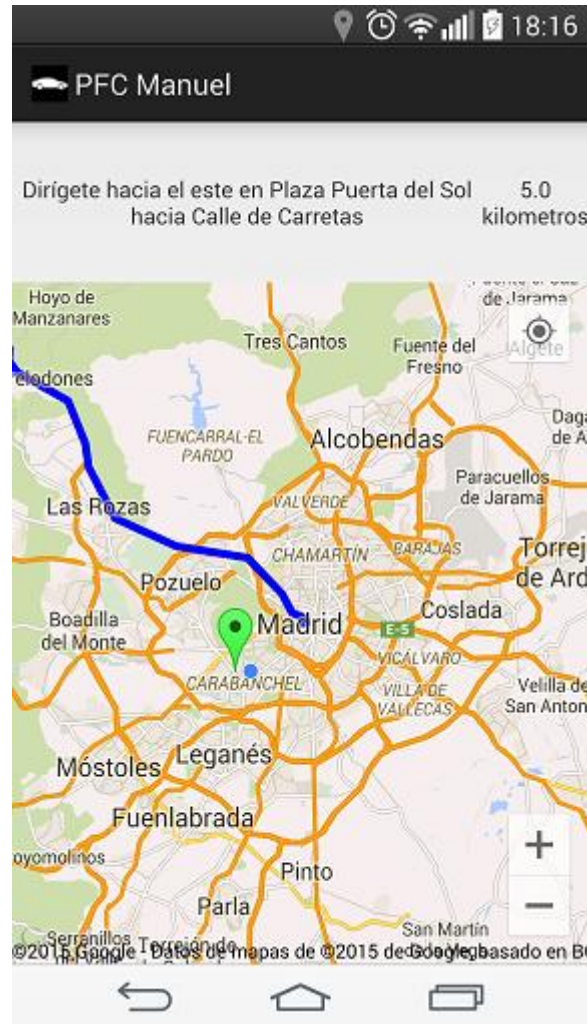


Figura 71. Interfaz seguimiento de ruta

En esta interfaz el usuario solo debe seguir las instrucciones que el dispositivo le vaya indicando. Se ha tomado como ejemplo una ruta entre Madrid y Salamanca y se ha alejado el zoom para poder mostrar un ejemplo de radar.

En la zona superior izquierda se muestra la instrucción oral que dará el móvil cuando estemos en las cercanías de este POI. La distancia de aviso es variable según la longitud del tramo en cuestión.

En la zona superior, se muestra la distancia al POI indicado anteriormente.

La línea azul que recorre el mapa es la ruta que debemos de seguir.

El marcador verde indica un radar y se mostrará cuando nos encontremos a 500 metros del mismo, aparte de un indicador oral que dirá “Ojo, radar a 500 metros”.

9.1.8. Modo Libre

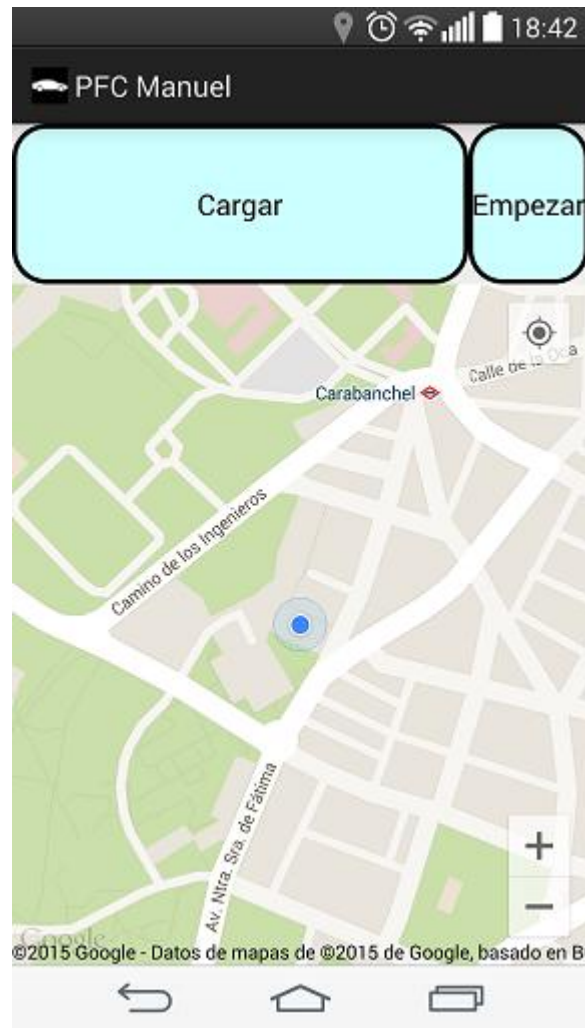


Figura 72. Interfaz Modo Libre

Interfaz principal del modo libre para el usuario. Se podrán generar rutas sin ningún tipo de limitación para luego poder repetirlas y competir contra el mismo.

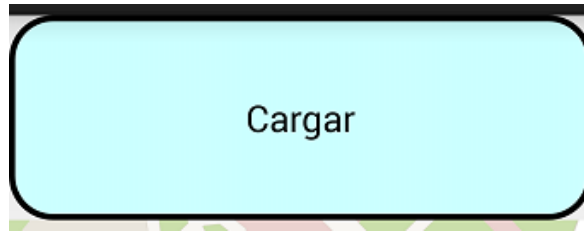


Figura 73. Botón cargar rutas

Pulsando este botón el mapa será sustituido por una lista con las rutas almacenadas para poder elegir si comenzarlas o eliminarlas.

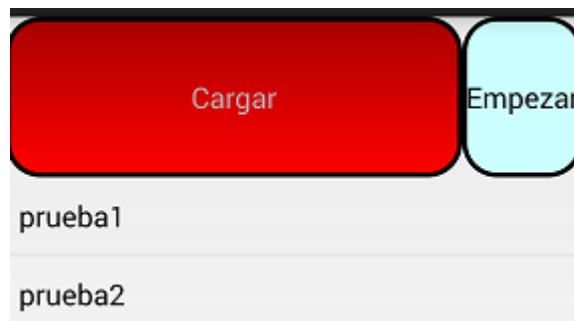


Figura 74. Listado de rutas libres almacenadas

Como se ve, el botón de cargar pasa a estar en color rojo debido a que no es posible usarlo en ese momento

Con un click corto iniciamos dicha ruta (explicado en 9.1.9) y con uno largo se nos da la opción de eliminar una ruta.

A continuación se muestra la el mapa cuando decidimos iniciar la ruta.

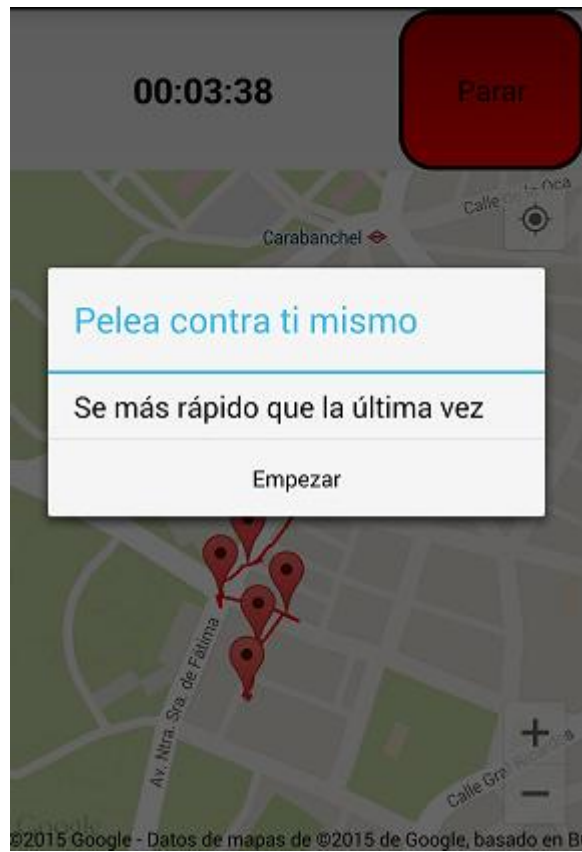


Figura 75. Modo Pelea contra ti mismo

En la zona superior izquierda vemos el tiempo del que disponemos para esta ruta. En el cuadro central, debemos pulsar el botón para comenzar la cuenta atrás de 5 segundos antes de que se inicie la cuenta del cronómetro. Los marcadores rojos indican los puntos de paso que debemos superar para que la ruta sea correcta.

A continuación se muestra los pasos si hubiéramos pulsado el botón empezar en vez de en botón cargar.



Figura 76. Botón empezar

Para comenzar una ruta en modo libre habremos de pulsar este botón. Con ello se comenzará a contar el tiempo y por otro lado a guardar en un archivo provisional. Se estarán tomando tiempos hasta pulsar el botón parar.

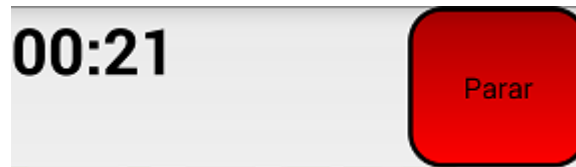


Figura 77. Etiqueta tiempo y botón parar

Durante la realización de la ruta, la parte superior de la interfaz pasará a ser de la manera que se ve en la figura de arriba. El tiempo irá contando segundo a segundo en la etiqueta. Al pulsar el botón parar, aparecerá el siguiente cuadro central.

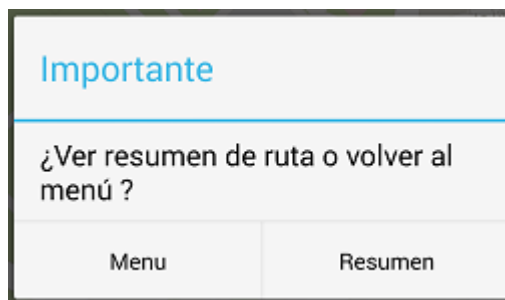


Figura 78. Cuadro Ruta Libre

Si pulsamos la etiqueta "Menu" volveremos a la interfaz principal (Figura 64)

Si el botón pulsado es el de "Resumen", accederemos a la última interfaz del programa, el resumen de la ruta.

9.1.9 Resumen de ruta libre

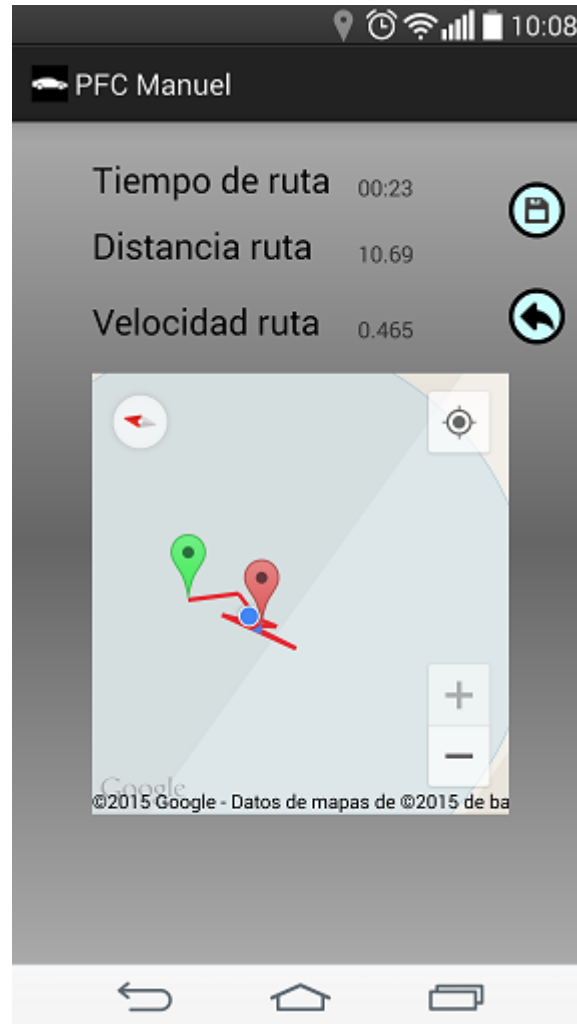


Figura 79. Interfaz resumen de ruta

Se muestran una serie de estadísticas y el dibujo de la ruta, con el marcador verde como punto final y el rojo como inicio de la misma



Figura 80. Botón guardar ruta

Este será el botón que pulsaremos para guardar una ruta libre, y nos aparecerá un recuadro para almacenar el nombre.

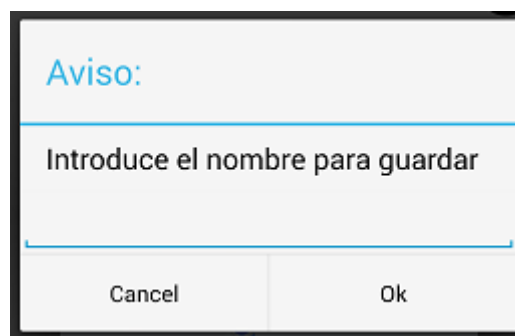


Figura 81. Cuadro central guardar ruta

La única restricción existente es que el nombre del archivo no esté repetido.



Figura 82. Botón volver

Este botón se utiliza para volver al menú principal (Figura 64) en el caso de no haber guardado, se nos recuerda por última vez si deseamos hacerlo.

10. Bibliografía

- [1] Pensamientos varios sobre programación, Android, .NET y Java
<www.sgoliver.net>
- [2] Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.
<stackoverflow.com>
- [3] Android draw route between two geo location MapV2
<<http://iamvijayakumar.blogspot.com.es/2013/04/android-draw-route-between-two-geo.html>>
- [4] How To Read XML File In Java – (DOM Parser)
<<http://www.mkyong.com/java/how-to-read-xml-file-in-java-dom-parser>>
- [5] Android SDK: Build a Speak and Repeat App
<<http://code.tutsplus.com/tutorials/android-sdk-build-a-speak-and-repeat-app--mobile-11197>>
- [6] Android Developers
< <https://developer.android.com/index.html>>
- [7] Google Developers – API de Google Maps
<<https://developers.google.com/maps/>>
- [8] Bring Location Services into Your Android App Development
<<http://www.developer.com/ws/android/programming/bring-location-services-into-your-android-app-development.html>>
- [9] Tutorial CRUD Android + SQLite (I): Crear la base de datos
<<http://notasprogramacion.sodenet.es/crud-android-sqlite-bbdd/>>
- [10] Sending Emails without User Intervention (no intents) in Android
<[http://www.jondev.net/articles/Sending_Emails_without_User_Intervention_\(no_Intents\)_in_Android](http://www.jondev.net/articles/Sending_Emails_without_User_Intervention_(no_Intents)_in_Android)>
- [11] Todo-poi La mejor base de radares de España
<<http://www.todo-poi.es/radar.htm>>

Anexo A. Diferencia unión punto a punto y algoritmo polilíneas codificadas

En este anexo vamos a tratar ambos modelos de generación de rutas, la unión punto a punto y el algoritmo de polilíneas codificadas.

En el primer caso, recibimos una serie de coordenadas que simplemente se unen a través de líneas rectas. A pesar de parecer una línea curva, es una sucesión de pequeñas líneas rectas.

Por otro lado, el algoritmo de polilíneas codificadas es el usado por Google Maps y consiste en realizar una conversión entre las coordenadas geográficas y la simbología ASCII.

El beneficio que tiene este método es que la codificación mejora la conservación del espacio ya que los puntos solo incluyen la desviación desde el punto anterior.

A continuación se especifican un par de ejemplos, desde la página de desarrolladores de Google:

1. Utiliza el valor firmado inicial:
-179.9832104
2. Utiliza el valor decimal y multiplícalo por 1e5, redondeando el resultado:
-17998321
3. Convierte el valor decimal en binario. Ten en cuenta que se deberá calcular un valor negativo con su **complemento a dos** invirtiendo el valor binario y añadiendo uno al resultado:
00000001 00010010 10100001 11110001
11111110 11101101 01011110 00001110
11111110 11101101 01011110 00001111
4. Desplaza hacia la izquierda el valor binario un bit:
11111101 11011010 10111100 00011110
5. Si el valor decimal original es negativo, invierte esta codificación:
00000010 00100101 01000011 11100001
6. Divide el valor binario en fragmentos de 5 bits (empezando por la derecha):
00001 00010 01010 10000 11111 00001
7. Coloca los fragmentos de 5 bits en orden inverso:
00001 11111 10000 01010 00010 00001
8. O cada valor con 0x20 si hay otro fragmento de bits a continuación:
100001 111111 110000 101010 100010 000001
9. Convierte cada valor en decimal:
33 63 48 42 34 1
10. Añade 63 a cada valor:
96 126 111 105 97 64
11. Convierte cada valor en su equivalente ASCII:
`~oia@

Figura 83. Ejemplo algoritmo polilíneas codificadas (valor)

También se muestra otro ejemplo de de una serie de puntos y la polilínea que resulta de hacer la codificación:

Puntos: (38.5, -120.2), (40.7, -120.95), (43.252, -126.453)

Latitud	Longitud	Latitud en E5	Longitud en E5	Cambio en la latitud	Cambio en la longitud	Latitud codificada	Longitud codificada	Punto codificado
38.5	-120.2	3850000	-12020000	+3850000	-12020000	_p~iF	~ps U	_p~iF~ps U
40.7	-120.95	4070000	-12095000	+220000	-75000	_u1L	nnqC	_u1LnnqC
43.252	-126.453	4325200	-12645300	+255200	-550300	_mqN	vxq`@	_mqNvxq`@

Polilínea codificada: `_p~iF~ps|U_u1LnnqC_mqNvxq`@`

Figura 84. Ejemplo de codificación de serie

Anexo B. Cálculo de distancias entre puntos. Navegación por estima inversa

En este anexo se va explicar el modo con el que se calculan las distancias entre dos puntos cada intervalo de tiempo definido.

La búsqueda de un método se ha realizado bajo una sola premisa, que el cálculo sea preciso en “distancias cortas”. Entrecorramos la expresión distancias cortas porque no hablamos de metros, es una comparación al tamaño de nuestro planeta, es decir, hablamos de unas distancias de varias decenas kilómetros.

Se ha elegido como método la navegación por estima inversa, un método náutico que nos proporciona valores precisos por debajo de las 200 millas náuticas (unos 370 kilómetros) y para el cual se necesitan conocer los puntos de inicio y llegada (en este caso, los pares de coordenadas decimales de cada punto). Este método es usado sobretodo en náutica, pero en nuestro caso es el indicado porque también calculamos distancias en línea recta al no disponer de más datos para realizar otro tipo de cálculo.

A continuación se muestra el método de cálculo:

PASOS	CÁLCULO A REALIZAR	PARÁMETRO	FÓRMULA
1	Cálculo de la Diferencia de latitud	ΔI	$\Delta I = I' - I$
2	Cálculo de la Diferencia de Longitud	ΔL	$\Delta L = L' - L$
3	Cálculo de la latitud media	I_m	$I_m = (I + I') / 2$
4	Cálculo del Apartamiento	A	$A = \Delta L * \text{Cos } I_m$
5	Cálculo del Rumbo	R	$\Delta I = D * \text{Cos } R;$ $\text{Tang } R = A/\Delta I$
6	Cálculo de la Distancia	D	$D = \Delta I / \text{Cos } R$

Figura 85. Cálculo de distancia mediante estimación inversa

Por último, solo faltaría convertir la distancia a metros, ya que la fórmula nos la da en millas náuticas, por lo cual se multiplica la distancia por 1.852 (relación milla náutica – kilómetro).

Anexo C. Posición del mapa en ruta. Concepto de bearing

En este anexo se explicará cómo se ha colocado el mapa en cada momento de la ruta para facilitar el seguimiento de la ruta por parte del usuario.

Los mapas que nos ofrece Google Maps están enfocados hacia el polo norte magnético, por lo cual no es una posición fija, debido a que este punto va cambiando con el paso del tiempo.

Según cálculos científicos, el polo norte magnético se mueve aproximadamente unos 55 kilómetros anuales, en la siguiente imagen se muestra aproximadamente la evolución entre 1831 y 2007.

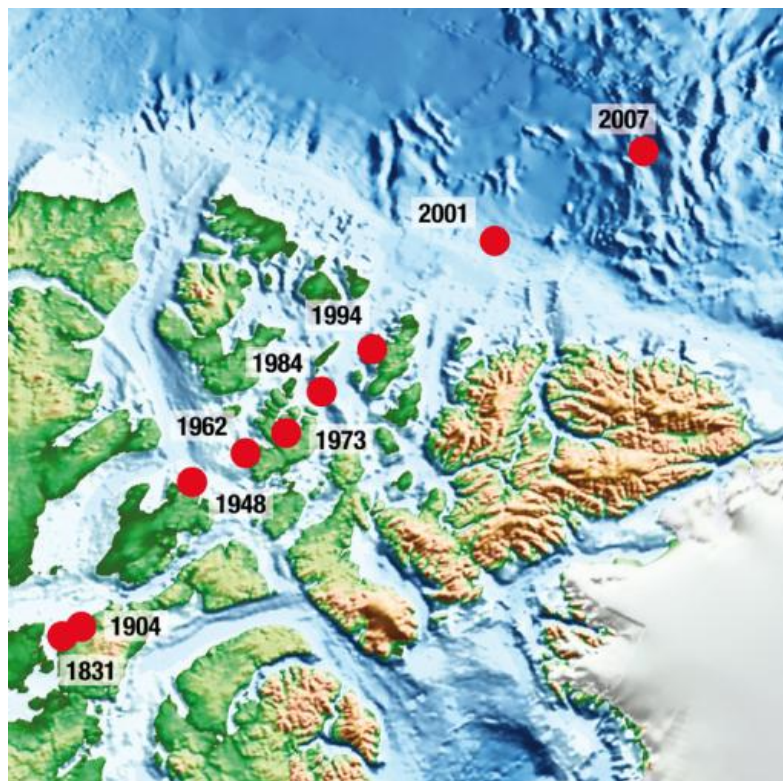


Figura 86. Variación polo norte magnético 1831-2007

En la aplicación, tomamos como polo norte magnético la aproximación que se conoce del año 2013, cuando se situaba aproximadamente en latitud 82.4 y longitud -113.4, al norte de Canadá.

El concepto de *bearing* (en castellano, rumbo) es el ángulo respecto al norte desde el punto en el que nos encontramos. Para nuestra aplicación, este punto se va a ir cambiando paso a paso.

Para el primer paso de cada ruta, el ángulo se calculará entre el polo norte magnético y el punto de inicio de nuestra ruta. A partir de ahí se calculará entre nuestra posición actual y el siguiente punto a pasar de la ruta diseñada. Con dicho ángulo, se moverá la cámara en los grados necesarios para que el siguiente paso quede aproximadamente centrado en pantalla.

La fórmula para calcularlo es la siguiente:

$$\theta = \text{atan2}(\sin \Delta\lambda \cdot \cos \varphi_2, \cos \varphi_1 \cdot \sin \varphi_2 - \sin \varphi_1 \cdot \cos \varphi_2 \cdot \cos \Delta\lambda)$$

Figura 87. Cálculo de bearing