



Universidad Carlos III de Madrid
Escuela Politécnica Superior

Grado de Ingeniería Informática
especialidad en Computación

*Diseño y desarrollo de un sistema
domótica sobre plataforma Android*
Trabajo de Fin de Grado

Autor: Daniel Amigo Herrero

Tutor: Miguel Ángel Patricio Guisado

En Colmenarejo, 04 de Julio de 2017

Agradecimientos

Este apartado tratará de explicar a los lectores del documento las razones por las cuales este proyecto despertó interés al equipo de desarrollo.

Quiero comenzar destacando el apoyo y sinceridad que me ofreció mi tutor, Miguel Ángel Patricio, desde el primer momento. Sin el este trabajo no habría existido. Yo tenía una idea sobre la que realizar el trabajo de fin de grado, y mientras que muchos profesores me dijeron, sin problema, que lo podíamos llevar a cabo, fue Miguel Ángel quien realmente pensó sobre la idea y me indicó que no era una buena idea realizar dicha aplicación, puesto que existían alternativas muy consolidadas y que hacían exactamente lo que yo quería hacer.

Además de darme su más sincera opinión sobre mi idea, Miguel Ángel me propuso una que me llamó la atención, desarrollar una aplicación para dar soporte a un sistema domótico, pero sin centralita, basado en dispositivos inteligentes por sí mismos.

Tras pensar varios días e informarme respecto al tema, pude observar como la idea es muy interesante, al no encontrar ninguna aplicación en el mercado que de soporte a sistemas domóticos basados en IoT. Además, me proporcionaba las dos prioridades que me había marcado para este trabajo de fin de grado, desarrollar una aplicación para dispositivos Android, y realizar como trabajo de fin de grado algo que sea útil.

A mis compañeros Miguelón, David y Álvaro, por tantas horas de estudio y de trabajos juntos, apoyándonos entre todos para sacar la carrera.

A mis padres y mi hermano, por confiar en mí y estar siempre ahí, en las buenas y en las malas.

Y a mi novia Sandy, quién me aguanta todos los días y me saca una sonrisa cuando más la necesito. Te quiero.

Resumen

El sector de la domótica está creciendo mucho en los últimos años. Es un área muy grande en la que hay mucho por explorar ahora que se cuenta con una tecnología más que preparada para ello. Además de las ventajas físicas que conllevan, la seguridad y el ahorro energético que ofrecen está más que demostrado.

Por otro lado, tenemos los dispositivos móviles o smartphones, que hace diez años no existían y se han arraigado a la sociedad actual de una manera nunca antes vista. No se concibe una sociedad sin ellos, todos tenemos mínimo un dispositivo inteligente.

Si unimos ambos conceptos, tenemos una aplicación para dispositivos móviles (Android en nuestro caso), capaz de controlar los dispositivos domóticos. Pero existe un problema, este concepto ya existe, controlando dispositivos de todo tipo. Pero, como se verá a lo largo del documento, estas soluciones cuentan con dispositivos de altísimo coste para su funcionalidad. En este proyecto nos vamos a enfocar en un ámbito más genérico que dispositivos fijos, trabajando con la tecnología Arduino, dando soporte a todo tipo de dispositivos, al poderse programar para realizar cualquier acción.

Además, esta solución implica otro beneficio más, puesto que los dispositivos Arduino interactúan con la aplicación vía WiFi, abarcando el concepto de IoT y dando una solución práctica y de calidad a las personas que busquen un sistema domótico inalámbrico de bajo presupuesto, algo que no existe en el mundo.

Palabras clave

Android, Arduino, Domótica, IoT, aplicación, Java, Thinger, Firebase.

Índice de Contenidos

1. Introducción.....	15
1.1. Motivación.....	15
1.1.1. Entorno Socio-económico	15
1.2. Objetivos.....	16
1.2.1. Estudio de la solicitud	16
1.2.2. Identificación de los interesados en el sistema.....	18
1.2.3. Objetivos del proyecto.....	18
1.3. Estructura de la memoria.....	19
1.4. Marco Regulador	20
1.4.1. Ley Orgánica de Protección de Datos	20
1.4.2. Estándares técnicos.....	21
2. Estado del Arte	22
2.1. Tecnológico	22
2.1.1. Domótica	22
2.1.2. Dispositivos móviles	24
2.1.3. Internet de las cosas.....	28
2.1.4. Arduino.....	30
2.2. Funcional	32
2.2.1. Home	32
2.2.2. Wink	33
2.2.3. Securitas Direct	34
2.2.4. Blynk	35
2.2.5. Comparativa respecto a las aplicaciones	36
2.3. Estudio de alternativas de solución	38
2.3.1. Sistema de gestión y control de dispositivos.....	38
2.3.2. Base de datos y servidor	40
3. Gestión y organización del proyecto	44
3.1. Metodología de desarrollo software utilizada	44
3.1.1. Explicación Craig-Larman	44
3.1.2. Explicación del procedimiento a seguir	45
3.2. Organización del proyecto.....	46
3.2.1. Roles	46

3.2.2. Recursos humanos	47
3.2.3. Recursos hardware.....	48
3.2.4. Recursos software.....	49
4. Planificación y especificación de requisitos.....	53
4.1. Identificación de requisitos.....	53
4.1.1. Procedimiento a seguir	53
4.1.2. Requisitos de usuario.....	54
4.1.3. Requisitos software	60
4.2. Prototipado	70
4.2.1. Explicación de wireframes	70
4.2.2. Conclusiones sacadas	76
4.3. Casos de uso	77
4.3.1. Actores involucrados	77
4.3.2. Casos de uso	78
4.3.3. Matriz de trazabilidad.....	99
4.4. Priorización.....	101
4.4.1. Características.....	101
4.4.2. Ponderación	102
4.4.3. Iteraciones.....	103
4.5. Planificación y presupuesto	105
4.5.1. Planificación	105
4.5.2. Presupuesto.....	107
5. Diseño de la aplicación.....	113
5.1. Diseño de la arquitectura	113
5.2. Diagramas de actividad	115
5.2.1. Iniciar sesión.....	115
5.2.2. Nuevo dispositivo	116
5.2.3. Nueva escena	117
5.2.4. Gestionar dispositivo	118
5.2.5. Gestionar escena	122
5.2.6. Gestionar localización	125
5.2.7. Gestionar perfil	129
5.2.8. Acerca de la aplicación.....	130

5.3. Diseño de la Base de Datos	130
6. Pruebas	132
6.1. Hardware	133
6.2. Flujo.....	134
6.2.1. Inicio de sesión con cuenta de Google	134
6.2.2. Nuevo dispositivo.....	134
6.2.3. Nueva escena	135
6.2.4. Gestionar dispositivo	135
6.2.5. Gestionar escena.....	138
6.2.6. Gestionar localización	140
6.2.7. Gestionar perfil.....	141
6.2.8. Acerca de	142
7. Conclusiones y líneas futuras	143
7.1. Conclusiones.....	143
7.1.1. Conclusiones a nivel de proyecto	143
7.1.2. Conclusiones a nivel personal	144
7.2. Líneas futuras	144
8. Anexo A. Manual de uso	146
8.1. Instalación de la aplicación.....	146
8.2. Creación de dispositivos Arduino	146
8.2.1. Descargar librerías.....	146
8.2.2. Código Arduino	147
8.3. Manual de la aplicación.....	148
8.3.1. Iniciar sesión.....	148
8.3.2. Menú principal.....	149
8.3.3. Dispositivos	150
8.3.4. Escenas	153
8.3.5. Localización	156
8.3.6. Gestión del perfil	158
8.3.7. Acerca de	158
9. Anexo B. Acrónimos, definiciones y referencias	159
9.1. Acrónimos y definiciones.....	159
9.2. Referencias	165

10. Anexo C. Inglés	168
10.1. Introduction	169
10.1.1. Motivation	169
10.1.2. Objectives.....	171
10.1.3. Regulatory environment	171
10.2. State of the art.....	172
10.2.1. Technological	172
10.2.2. Functional.....	172
10.2.3. Study of alternatives of solution.....	173
10.3. Management and organization of the Project.....	173
10.3.1. Methodology of software development used.....	173
10.3.2. Organization of the project.....	174
10.3.3. Hardware resources	175
10.3.4. Software resources.....	176
10.4. Planning and specification of requirements.....	177
10.5. Prototyping	178
10.6. Use cases	178
10.7. Prioritization	179
10.8. Planning and budget	179
10.8.1. Planning	179
10.8.2. Budget.....	180
10.9. Application design.....	181
10.9.1. Architecture design.....	181
10.9.2. Activity diagrams	182
10.9.3. Design of the database.....	182
10.10. Tests.....	182
10.11. Conclusions	183
10.11.1. Conclusions to the project level.....	183
10.11.2. Conclusions to the personal level.....	184
10.12. Future lines	184

Índice de Tablas

Tabla 1. Organismos del ámbito domótico por localización.....	21
Tabla 2. Comparación aplicaciones domótica.....	37
Tabla 3. Ventajas / desventajas sistema propio.....	38
Tabla 4. Ventajas / desventajas Thinger.....	39
Tabla 5. Ventajas / desventajas base de datos MySQL + Servidor.....	40
Tabla 6. Ventajas / desventajas Google Firebase.....	41
Tabla 7. Ventajas / desventajas Amazon Web Services (AWS).....	42
Tabla 8. Currículo Daniel Amigo.....	47
Tabla 9. Recurso hardware. Portátil.....	48
Tabla 10. Recurso hardware. Teléfono Móvil.....	48
Tabla 11. Recurso hardware. Arduino.....	48
Tabla 12. Estructura requisitos.....	53
Tabla 13. Requisito de usuario 01.....	54
Tabla 14. Requisito de usuario 02.....	54
Tabla 15. Requisito de usuario 03.....	54
Tabla 16. Requisito de usuario 04.....	54
Tabla 17. Requisito de usuario 05.....	55
Tabla 18. Requisito de usuario 06.....	55
Tabla 19. Requisito de usuario 07.....	55
Tabla 20. Requisito de usuario 08.....	55
Tabla 21. Requisito de usuario 09.....	55
Tabla 22. Requisito de usuario 10.....	56
Tabla 23. Requisito de usuario 11.....	56
Tabla 24. Requisito de usuario 12.....	56
Tabla 25. Requisito de usuario 13.....	56
Tabla 26. Requisito de usuario 14.....	57
Tabla 27. Requisito de usuario 15.....	57
Tabla 28. Requisito de usuario 16.....	57
Tabla 29. Requisito de usuario 17.....	57
Tabla 30. Requisito de restricción 01.....	58
Tabla 31. Requisito de restricción 02.....	58
Tabla 32. Requisito de restricción 03.....	58
Tabla 33. Requisito de restricción 04.....	58
Tabla 34. Requisito de restricción 05.....	59
Tabla 35. Requisito de restricción 06.....	59
Tabla 36. Requisito funcional 01.....	60
Tabla 37. Requisito funcional 02.....	60
Tabla 38. Requisito funcional 03.....	60
Tabla 39. Requisito funcional 04.....	61
Tabla 40. Requisito funcional 05.....	61
Tabla 41. Requisito funcional 06.....	61

Tabla 42. Requisito funcional 07.....	62
Tabla 43. Requisito funcional 08.....	62
Tabla 44. Requisito funcional 09.....	62
Tabla 45. Requisito funcional 10.....	62
Tabla 46. Requisito funcional 11.....	63
Tabla 47. Requisito funcional 12.....	63
Tabla 48. Requisito funcional 13.....	63
Tabla 49. Requisito funcional 14.....	63
Tabla 50. Requisito funcional 15.....	64
Tabla 51. Requisito funcional 16.....	64
Tabla 52. Requisito funcional 17.....	64
Tabla 53. Requisito funcional 18.....	64
Tabla 54. Requisito funcional 19.....	65
Tabla 55. Requisito funcional 20.....	65
Tabla 56. Requisito funcional 21.....	65
Tabla 57. Requisito funcional 22.....	66
Tabla 58. Requisito funcional 23.....	66
Tabla 59. Requisito funcional 24.....	66
Tabla 60. Requisito funcional 25.....	66
Tabla 61. Requisito funcional 26.....	67
Tabla 62. Requisito funcional 27.....	67
Tabla 63. Requisito no funcional 01.....	67
Tabla 64. Requisito no funcional 02.....	67
Tabla 65. Requisito no funcional 03.....	68
Tabla 66. Requisito no funcional 04.....	68
Tabla 67. Requisito no funcional 05.....	68
Tabla 68. Requisito no funcional 06.....	68
Tabla 69. Requisito no funcional 07.....	69
Tabla 70. Requisito no funcional 08.....	69
Tabla 71. Requisito no funcional 09.....	69
Tabla 72. Requisito no funcional 10.....	69
Tabla 73. Estructura casos de uso.....	77
Tabla 74. UC_01 Inicio de sesión con cuenta de Google.....	78
Tabla 75. UC_02 Nuevo dispositivo	80
Tabla 76. UC_03 Nueva escena	81
Tabla 77. UC_04 Gestionar dispositivo	82
Tabla 78. UC_05 Gestionar escena	83
Tabla 79. UC_06 Gestionar localización.....	83
Tabla 80. UC_07 Gestionar perfil	84
Tabla 81. UC_08 Acerca de la aplicación	84
Tabla 82. UC_09 Cerrar sesión	85
Tabla 83. UC_10 Revocar acceso.....	86
Tabla 84. UC_11 Cambiar localización	87
Tabla 85. UC_12 Modificar dispositivo	88

Tabla 86. UC_13 Borrar dispositivo	89
Tabla 87. UC_14 Cambiar estado acción	90
Tabla 88. UC_15 Recargar sensores	91
Tabla 89. UC_16 Modificar escena.....	92
Tabla 90. UC_17 Borrar escena	93
Tabla 91. UC_18 Activar / desactivar escena.....	93
Tabla 92. UC_19 Petición Put	94
Tabla 93. UC_20 Petición Get.....	95
Tabla 94. UC_21 Escribir en la Base de Datos	97
Tabla 95. UC_22 Borrar de la Base de Datos.....	97
Tabla 96. UC_23 Revocar acceso de la Base de Datos	98
Tabla 97. UC_24 Leer de la Base de Datos.....	98
Tabla 98. Matriz de Trazabilidad	100
Tabla 99. Ponderación casos de uso	102
Tabla 100. Priorización iteración 1.....	103
Tabla 101. Priorización iteración 2.....	104
Tabla 102. Priorización iteración 3.....	104
Tabla 103. Importe salario mínimo rol.....	109
Tabla 104. Desglose Seguridad Social	109
Tabla 105. Coste por hora final rol.....	110
Tabla 106. Presupuesto recursos humanos	110
Tabla 107. Presupuesto recursos hardware.....	110
Tabla 108. Presupuesto recursos software.....	111
Tabla 109. Presupuesto costes indirectos	111
Tabla 110. Presupuesto cálculo total de costes.....	112
Tabla 111. Componentes diagrama de actividad.....	115
Tabla 112. Tabla estructura Prueba	132
Tabla 113. Prueba PH.01	133
Tabla 114. Prueba PH.02.....	133
Tabla 115. Prueba PF.01	134
Tabla 116. Prueba PF.02	134
Tabla 117. Prueba PF.03	134
Tabla 118. Prueba PF.04	135
Tabla 119. Prueba PF.05	135
Tabla 120. Prueba PF.06	136
Tabla 121. Prueba PF.07	136
Tabla 122. Prueba PF.08	136
Tabla 123. Prueba PF.09	137
Tabla 124. Prueba PF.10	137
Tabla 125. Prueba PF.11	137
Tabla 126. Prueba PF.12	138
Tabla 127. Prueba PF.13	138
Tabla 128. Prueba PF.14	139
Tabla 129. Prueba PF.15	139

Tabla 130. Prueba PF.16	139
Tabla 131. Prueba PF.17	140
Tabla 132. Prueba PF.18	140
Tabla 133. Prueba PF.19	140
Tabla 134. Prueba PF.20	141
Tabla 135. Prueba PF.21	141
Tabla 136. Prueba PF.22	141
Tabla 137. Prueba PF.23	142
Tabla 138. Prueba PF.24	142
Tabla 139. Prueba PF.25	142
Tabla 140. Prueba PF.26	142
Tabla 141. Comparison to other application	172
Tabla 142. Computer resources	175
Tabla 143. Android resources.....	176
Tabla 144. Arduino resources.....	176
Tabla 145. Example requirement.....	177
Tabla 146. Use cases requirements.....	178
Tabla 147. Prioritization characteristics	179
Tabla 148. Budget summary.....	180
Tabla 149. Test example.....	183

Índice de Ilustraciones

Ilustración 1. Comparativa venta de móviles [15].....	27
Ilustración 2. Distribución ventas móviles 2016 Q4 [15].....	28
Ilustración 3. Ilustración IoT	28
Ilustración 4. Progresión dispositivos conectados por persona [19]	29
Ilustración 5. NodeMCU	31
Ilustración 6. Apple Home.....	32
Ilustración 7. Wink	33
Ilustración 8. Securitas Direct	34
Ilustración 9. Blynk	35
Ilustración 10. Imagen Thinger	38
Ilustración 11. Dashboards Thinger.....	39
Ilustración 12. Imagen Google Firebase.....	40
Ilustración 13. Imagen Amazon Web Services (AWS).....	42
Ilustración 14. Fases Craig-Larman.....	44
Ilustración 15. Roles del proyecto	46
Ilustración 16. Estructura de proyecto en Android Studio	50
Ilustración 17. Entrada TimeCamp.....	51
Ilustración 18. Lista tiempos TimeCamp	52
Ilustración 19. Wireframe inicio de sesión.....	70
Ilustración 20. Wireframe menú principal 1.....	71
Ilustración 21. Wireframe menú principal 2.....	71
Ilustración 22. Wireframe botón flotante	72
Ilustración 23. Wireframe nuevo dispositivo 1	72
Ilustración 24. Wireframe nuevo dispositivo 2	72
Ilustración 25. Wireframe vista dispositivo.....	73
Ilustración 26. Wireframe nueva escena 1.....	73
Ilustración 27. Wireframe nueva escena 2.....	73
Ilustración 28. Wireframe nueva escena 3.....	74
Ilustración 29. Wireframe vista escena.....	74
Ilustración 30. Wireframe menú de navegación.....	75
Ilustración 31. Wireframe seleccionar localización 1	75
Ilustración 32. Wireframe seleccionar localización 2	75
Ilustración 33. Casos de uso Inicio.....	78
Ilustración 34. Casos de uso Pantalla principal	79
Ilustración 35. Casos de uso Perfil	84
Ilustración 36. Casos de uso Localización.....	86
Ilustración 37. Casos de uso Dispositivo.....	87
Ilustración 38. Casos de uso Escena.....	91
Ilustración 39. Casos de uso Thinger	94
Ilustración 40. Casos de uso Firebase.....	96
Ilustración 41. Diagrama Gantt	106

Ilustración 42. Diseño de la arquitectura	114
Ilustración 43. Diagrama de actividad Iniciar sesión	115
Ilustración 44. Diagrama de actividad Nuevo dispositivo.....	116
Ilustración 45. Diagrama de actividad Nueva escena.....	117
Ilustración 46. Diagrama de actividad Modificar dispositivo	118
Ilustración 47. Diagrama de actividad Borrar dispositivo.....	119
Ilustración 48. Diagrama de actividad Ver valores sensores.....	120
Ilustración 49. Diagrama de actividad Modificar valor acción	121
Ilustración 50. Diagrama de actividad Modificar escena	122
Ilustración 51. Diagrama de actividad Borrar escena.....	123
Ilustración 52. Diagrama de actividad Activar / desactivar escena.....	124
Ilustración 53. Diagrama de actividad Modificar localización.....	125
Ilustración 54. Diagrama de actividad Borrar localización	126
Ilustración 55. Diagrama de actividad Cambiar localización.....	127
Ilustración 56. Diagrama de actividad Nueva localización	128
Ilustración 57. Diagrama de actividad Cerrar sesión.....	129
Ilustración 58. Diagrama de actividad Revocar acceso.....	129
Ilustración 59. Diagrama de actividad Acerca de.....	130
Ilustración 60. Diagrama Base de datos	131
Ilustración 61. Instalación de la aplicación	146
Ilustración 62. Manual Iniciar sesión	148
Ilustración 63. Manual Menú principal 1	149
Ilustración 64. Manual Menú principal 2	149
Ilustración 65. Manual Menú principal 3	149
Ilustración 66. Manual Nuevo dispositivo 1.....	150
Ilustración 67. Manual Nuevo dispositivo 2.....	150
Ilustración 68. Manual Dispositivo menú principal 2	151
Ilustración 69. Manual Dispositivo menú principal 1	151
Ilustración 70. Manual Vista dispositivo.....	152
Ilustración 71. Manual Editar dispositivo.....	152
Ilustración 72. Manual Borrar dispositivo.....	153
Ilustración 73. Manual Nueva escena 1	154
Ilustración 74. Manual Nueva escena 2.....	154
Ilustración 75. Manual Nueva escena 3.....	154
Ilustración 76. Manual Vista escena.....	155
Ilustración 77. Manual Editar escena	155
Ilustración 78. Manual Borrar escena.....	156
Ilustración 79. Manual Gestión localizaciones 1	157
Ilustración 80. Manual Gestión localizaciones 2.....	157
Ilustración 81. Manual Gestión del perfil.....	158
Ilustración 82. Manual Acerca de 1.....	158
Ilustración 83. Manual Acerca de 2.....	158
Ilustración 84. Craig Larman summarized	174
Ilustración 85. Roles	175

Ilustración 86. Architecture design..... 181

1. Introducción

El presente documento realiza una visión completa y detallada con toda la información respecto al desarrollo de la aplicación de domótica para dispositivos Android realizada como trabajo de fin de grado.

1.1. Motivación

Los seres humanos constantemente han buscado la facilidad a la hora de llevar a cabo acciones. Desde la prehistoria, siempre hemos tenido la curiosidad y la visión para crear cosas que puedan ser de utilidad. La tecnología siempre ha existido, y ha seguido una curva exponencial en nuestra existencia, hasta la actualidad, donde la vida se rige por los cambios tecnológicos que se producen.

La máquina de vapor, los motores de combustión interna, los dispositivos electrónicos... Así hemos llegado a una sociedad en la que se habla constantemente de revolución tecnológica, de cómo el internet de las cosas (IoT) y las redes sociales son el futuro cercano, pero vemos como el tiempo pasa y no llegamos a ver productos finales que exploten necesidades reales.

Con esta motivación surge la idea de esta aplicación. En un ámbito en auge, como la domótica, altamente deseado y prácticamente inaccesible para el público general por la alta inversión necesaria para su disfrute, no existe ninguna solución que exprese un concepto tan interesante como el IoT. Por tanto, existe un nicho de mercado inexplorado por los desarrolladores de aplicaciones móviles, el cual abarcará la aplicación Android a desarrollar.

Imagina tener por 10€ un dispositivo domótico, programable, para el uso que quieras, fácil de usar, al no requerir ningún cable ni una instalación en la vivienda y con una aplicación de calidad para su control. ¿No es increíble?

1.1.1. Entorno Socio-económico

A día de hoy, es una realidad encontrarse en viviendas de nueva construcción con sistemas domóticos instalados de fábrica, diseñados íntegramente para las necesidades de cada familia. Pero aparte, está surgiendo una parte de los usuarios que demandan dispositivos que sacien sus necesidades con dispositivos domóticos que se puedan adaptar a sus viviendas actuales. Esto viene fuertemente influenciado por la evolución de la tecnología, especialmente de los Smartphones, que nos facilitan hasta tal punto la vida, que queremos que esta ayuda se dé también en nuestras viviendas.

En el aspecto económico, este proyecto es más que viable para cualquier ciudadano. La tecnología se ha desarrollado tanto, que se pueden conseguir placas Arduino a menos de cinco euros. Con una de ellas, e Internet en la vivienda, el cual está más que implantado en la sociedad, teniendo más de un 75% de la población española acceso a él [1], queda

más que claro que el aspecto económico no es ningún impedimento para los ciudadanos. Añadir también, que un sistema domótico bien instalado y programado es capaz de ahorrar para el usuario, si se maximizan las horas de luz, así como previniendo errores humanos, como luces encendidas u otros imprevistos.

En el aspecto social es algo diferente. La población española está separada por los cambios tecnológicos. La población joven y de mediana edad, está adaptada al desarrollo tecnológico. Han crecido de la mano de redes sociales, infinidad de dispositivos diferentes, y consumen de Internet a diario... verían con buenos ojos la implantación de sistemas domóticos en sus viviendas. No tienen miedo, en general, a las máquinas, las ven como una ayuda. En cambio, la población de mayor edad, en su mayoría, podría mostrar un mayor rechazo a que le gestione una máquina la puerta de su casa, por ejemplo. Están arraigados en sus costumbres, como es lógico. Es algo extraño para ellos, no han experimentado desde pequeños la evolución tecnológica a este nivel, y podría ser un impedimento para el proyecto. No hay que olvidar que este grupo de la población es el que más empleo tiene, y, por tanto, mayor posibilidad de ser un potencial comprador de sistemas domóticos.

A pesar de ello, debido a la baja inversión que hay que realizar para hacerlo funcionar, se puede esperar una buena acogida de la aplicación.

Por último, indicar que el proyecto no tendría apenas impacto ambiental. Requiere únicamente componentes electrónicos, los cuales tienen una cadena de reciclado bastante extensa en la sociedad, permitiendo reutilizar prácticamente todos los materiales necesarios para su construcción.

1.2. Objetivos

En este apartado se realiza un análisis de la propuesta realizada por el cliente Miguel Ángel Patricio Guisado, estudiando la misma y observando los límites de la misma, y concluyendo con los objetivos marcados para el presente proyecto. Este apartado viene marcado la gestión y organización del proyecto, establecidas en el apartado 3 del documento.

1.2.1. Estudio de la solicitud

Nuestro cliente requiere una aplicación para Android que gestione un sistema domótico completo. Este, resalta, debe ser capaz de registrar cada uno de los dispositivos a controlar a cada cuenta de usuario, e interactuar con ellos mediante la plataforma Thinger.io.

1.2.1.1. Técnica de elicitación

El jefe de proyecto aprovechó la primera reunión con el cliente para realizar una Unstructured Review con el cliente. Esta es una técnica de elicitación que tiene como objetivo de obtener mayor información para la posterior elicitación de requisitos.

Durante la entrevista, que duró aproximadamente treinta minutos, el entrevistador hizo hincapié principalmente en cómo gestionar los distintos dispositivos a los que dar soporte desde la aplicación, previa configuración en la plataforma Thinger.

Finalmente, el jefe de proyecto concluyó lo siguiente sobre la entrevista:

- Es necesario una base de datos que almacene la configuración para cada dispositivo.
- Los usuarios deben tener acceso a diferentes localizaciones, en las cuales tener sus dispositivos.
- El cliente quiere una funcionalidad para realizar varias acciones de manera automática, al mismo tiempo. Por ejemplo, comentó, “subir las persianas y apagar las luces”.

Además, finaliza haciendo hincapié en que el proyecto esté preparado ser entregado en el segundo periodo de defensa del Trabajo de Fin de Grado de la UC3M en 2017.

Observadas las necesidades del cliente, observamos las siguientes restricciones, ya sean impuestas por el o implícitas, basadas en suposiciones:

- **Restricciones económicas:** El cliente no ha impuesto ninguna restricción económica para la realización del proyecto.
- **Restricciones técnicas:**
 - Basada en las suposiciones hechas, el proyecto se restringe a utilizar la plataforma Thinger.io, por tanto, hay que adaptarse a la misma.
 - Basada en suposiciones, el sistema se implantará sobre un aparcamiento existente con una estructura predefinida.
- **Restricciones operativas:**
 - Basada en suposiciones, el sistema debe permitir el funcionamiento de la aplicación durante las 24 horas del día.
- **Restricciones legales:**
 - Basada en suposiciones, se debe mantener la confidencialidad de los datos de los clientes:
 - Datos del usuario.
 - Datos obtenidos por los diferentes dispositivos.
- **Restricciones temporales:**
 - El cliente ha impuesto la restricción temporal para la realización del proyecto, para junio-julio del 2017, sin especificar más.

1.2.2. Identificación de los interesados en el sistema

En el siguiente apartado se definen las partes interesadas en este desarrollo:

1. El cliente solicitante, al obtener el sistema requerido de gestión de un sistema domótico, con las características oportunas.
2. Los usuarios de la aplicación, al obtener una solución que cubre las necesidades que buscan.
3. El encargado del proyecto, Daniel Amigo, adquiriendo experiencia y prestigio en desarrollos software, cumpliendo los siguientes roles:
 - Jefe de proyecto: encargado de comunicarse con el cliente, en todas las fases.
 - Analista: encargado de elaborar los requisitos a partir de la información proporcionada por el jefe de proyecto, recogida a su vez del cliente.
 - Programador: implementa el diseño elaborado por los analistas, y posteriormente realizará las pruebas necesarias, definidas por el analista sobre el código.

1.2.3. Objetivos del proyecto

Por lo anteriormente expuesto, concluimos los objetivos del proyecto como:

1. Analizar en profundidad los sistemas domóticos existentes en la actualidad, observando sus funcionalidades y sus limitaciones.
2. Analizar las necesidades que debe tener una herramienta de gestión domótica actual.
3. Diseñar la arquitectura del sistema a desarrollar.
4. Y, por último, proceder al desarrollo de la aplicación.

1.3. Estructura de la memoria

La estructura de la presente memoria se va a ordenar en distintos apartados. Cabe destacar que la organización del mismo se ha diseñado para facilitar la comprensión del mismo en una lectura ordenada por cada apartado, realizando en los primeros temas una introducción a los conceptos y conocimientos a abordar en sucesivos temas.

- Apartado 1. Introducción. Se realiza un pequeño resumen explicando el contenido del documento, la motivación para realizar el presente proyecto y una visión global del documento. Además, alojará el marco regulador y el entorno socio-económico del proyecto.
- Apartado 2. Estado del arte: En este apartado se introduce al lector al contenido del documento. Se realiza una explicación sobre los distintos temas evitando tecnicismos en la medida de lo posible y se realiza un análisis sobre las aplicaciones similares de mayor calidad en el mismo ámbito que el presente proyecto. Finalmente, se hace un análisis de las alternativas de diseño a utilizar.
- Apartado 3. Gestión y organización del proyecto. En este apartado se detalla la metodología de desarrollo software a utilizar a lo largo del proyecto. Además, se detallan los recursos para la realización del proyecto, tanto humanos como materiales.
- Apartado 4. Planificación y especificación de requisitos. En este apartado se realizan unos prototipos con los que identificar los requisitos y se definen los casos de uso. Con ellos, se realiza una priorización de los mismos y se realiza la planificación del proyecto, así como el presupuesto del mismo.
- Apartado 5. Diseño de la aplicación. En este apartado los analistas especifican el diseño de la arquitectura del sistema, utilizando diversos diagramas para su comprensión.
- Apartado 6. Pruebas. En este apartado los analistas definen las pruebas a realizar sobre cada uno de los casos de uso, llevadas a cabo por los programadores.
- Apartado 7. Conclusiones y líneas futuras. En este apartado se comprueba el cumplimiento de los objetivos marcados al comienzo del proyecto, culminando el mismo. Además, gracias al conocimiento adquirido con el proyecto, se describen posibles mejoras a realizar sobre la aplicación, así como posibles extensiones paralelas al proyecto con el fin de culminar el producto ofrecido.
- Anexo A. Manual de uso. En este apartado se adjunta un manual detallado para que cualquier posible usuario de la aplicación aprenda a utilizar cada funcionalidad de la misma.
- Anexo B. Acrónimos, definiciones y referencias. En este último apartado se incluyen todas las explicaciones necesarias para la comprensión de todos los tecnicismos y términos específicos que contiene el documento.
- Anexo C. Apartado en inglés. Como es requerido para los alumnos del plan 2011.

1.4. Marco Regulatorio

En este apartado se mostrarán los resultados del estudio realizado sobre el marco regulatorio vigente, con el fin de desarrollar la aplicación Android sin quebrantar ninguna ley que pueda comprometer a la misma aplicación ni a los desarrolladores.

A continuación, se adjuntan los fragmentos de leyes en vigor que puedan afectar directa o indirectamente al proyecto realizado.

1.4.1. Ley Orgánica de Protección de Datos

La Ley Orgánica 15/1999 del 14 de diciembre de 1999, de Protección de Datos de Carácter Personal, entrada en vigor desde el 14 de enero del año 2000, tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar. [2]

El desarrollo de este proyecto ha tenido en cuenta los siguientes artículos que pudieran afectar a la aplicación:

- Artículo 4: Calidad de los datos:

“Los datos de carácter personal sólo se podrán recoger para su tratamiento, así como someterlos a dicho tratamiento, cuando sean adecuados, pertinentes y no excesivos en relación con el ámbito y las finalidades determinadas, explícitas y legítimas para las que se hayan obtenido.”

Los datos almacenados por parte de la aplicación, previo conocimiento del usuario, tienen como único fin el correcto funcionamiento de la aplicación y es únicamente el usuario quien puede acceder a los mismos, permitiendo eliminar o modificarlos cuando desee.

- Artículo 6: Consentimiento del afectado

“El tratamiento de los datos de carácter personal requerirá el consentimiento inequívoco del afectado, salvo que la ley disponga otra cosa.”

El presente proyecto cumple dicho artículo, puesto que es el usuario quien introduce toda la información a almacenar, en ningún caso la información que recopila la aplicación carece del consentimiento del mismo.

Además, se informa a los usuarios de todo el tratamiento de datos que se realizará de ellos en la aplicación previamente al inicio de sesión del usuario.

1.4.2. Estándares técnicos

Es necesario indicar que existen distintas normas técnicas referentes al ámbito de la domótica, definidas por diferentes organismos en función de su lugar de aplicación, siendo los siguientes los principales en las distintas localizaciones que nos afectan:

Ámbito de aplicación	General	Eléctrico	Telecomunicaciones
Internacional			
Europeo			
Nacional			

Tabla 1. Organismos del ámbito domótico por localización

Las normas técnicas son de carácter voluntario, y la siguiente es la única que aplica a nuestro proyecto.

1.4.2.1. Normas UNE-EN 50090 para Sistemas Electrónicos de Viviendas y Edificios (HBES)

Las normas UNE-EN 50090 [3] normalizan las aplicaciones de control del sistema de comunicación abierto destinado a viviendas y edificios. Cubren cualquier combinación de dispositivos electrónicos conectados a través de una red de transmisión digital y tienen en cuenta los sistemas de control de automatización, tanto descentralizados como distribuidos.

Esta serie de normas se centra en la Clase I del sistema de comunicación (comunicación de datos a baja velocidad destinada al control) como, por ejemplo: control del alumbrado, calefacción, gestión de energía, alarma de incendios, control de persianas, diferentes formas de control de seguridad, entre otros.

2. Estado del Arte

En esta sección vamos a realizar una breve introducción respecto a los temas tratados en el presente documento, garantizando a cualquier lector del mismo la perfecta comprensión de los futuros apartados.

Esta sección se divide en varios apartados, en función del tipo de aspecto a tratar:

- Tecnológico: se explica el concepto tecnológico a tratar.
- Funcional: se estudia el mercado de aplicaciones domóticas, observando las ventajas y desventajas de cada una.
- Estudio de alternativas de solución: finalmente, se realiza un estudio sobre los aspectos tecnológicos requeridos para el desarrollo de la aplicación, comparando las alternativas existentes en el mercado.

2.1. Tecnológico

2.1.1. Domótica

El término domótica se crea por la unión de dos palabras: domus (casa en latín), y tica (automática en griego). [4]

La Real Academia Española explica el concepto de domótica como “un conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda.” [5] Esta definición la podemos ampliar, puesto que no es simplemente la automatización de las instalaciones, sino la capacidad de gestionar y controlar cada sistema, así como comunicarse entre ellos, ya sea de manera alámbrica o inalámbrica.

Además, es imprescindible que cuente con interacción con el usuario, puesto que es quien debe beneficiarse del uso de estas tecnologías en la misma. Esta interacción se puede realizar de diversas maneras, mediante un Smartphone, un dispositivo embebido en la pared, control por voz, o soluciones más sencillas, como interruptores, etc.

La domótica no hay que confundirla con los términos inmótica: aplicar la domótica en el sector terciario (no residencias), ni urbótica: aplicar la domótica en la ciudad (relacionado con el concepto Smart City).

2.1.1.1. Beneficios proporcionados

Un sistema domótico otorga al usuario diversos beneficios, entre los que se destacan los siguientes:

2.1.1.1.1. *Accesibilidad*

Un sistema domótico, independientemente, facilita el uso de los diversos sistemas que abarque, pudiendo llegar a considerarse una vivienda asistida. Como observamos, es de gran utilidad para cualquier persona, pero más aún si la persona cuenta con discapacidades o dependencias (problemas de movilidad, de visión, etc.).

Es importante, por tanto, que el sistema domótico se preste a que el usuario sienta el control sobre el sistema, contando con los sistemas de interacción adaptados y accesibles. De esta manera un servicio domótico adaptado a las condiciones de cada persona, con tecnologías concienzudamente pensadas para cada tipo de persona puede mejorar la calidad de vida de la misma, así como facilitar la integración de las personas discapacitadas en la sociedad. [6]

2.1.1.1.2. *Comunicación*

La capacidad de un sistema domótico a comunicarse con el usuario es imprescindible para permitir al mismo sentir control sobre el sistema, ya esté dentro o fuera de la vivienda, desde diferentes dispositivos, de una manera rápida y sencilla con la tecnología de comunicaciones actual.

Un ejemplo de comunicación puede ser cómo el propio usuario controla el sistema de manera remota o local (temperatura, alarmas, etc.), detectar anomalías, control parental, etc.

2.1.1.1.3. *Confort*

Además, la domótica tiene claramente funcionalidades destinadas a obtener el bienestar de sus usuarios, haciendo principalmente que dispositivos realicen el trabajo por nosotros, como puede ser encender las luces, subir las persianas, etc.

2.1.1.1.4. *Ahorro*

La alta tecnología de la que disponen los sistemas domóticos influyen directamente al ahorro en las necesidades básicas (agua, gas, electricidad, calefacción, etc.). [7]

Este ahorro viene definido por el control automático que realiza el sistema, previniendo errores humanos, como puede ser dejar la luz encendida, o el grifo del agua abierto, y maximizando el impacto de los recursos para encontrar cada lugar de la casa en su estado óptimo.

Además, hay que tener en cuenta los posibles gastos por intentar cumplir las otras características, como puede ser contar con una asistencia por condiciones de accesibilidad, o vigilantes de seguridad.

2.1.1.1.5. *Seguridad*

Por último, tenemos el concepto de seguridad. Garantizar la seguridad de los habitantes debe ser imprescindible en cualquier vivienda domótica. Para ello, en la actualidad se cuenta con una gran cantidad de dispositivos capaces de parametrizar multitud de peligros que puedan surgir, como pueden ser cámaras de video-vigilancia, sensores de humedad, temperatura, así como estar conectados con los sistemas de terceras empresas destinados exclusivamente a este sector.

2.1.2. *Dispositivos móviles*

La sociedad humana está fuertemente ligada a los avances tecnológicos. Existen multitud de hitos tecnológicos en la historia, hallados por unos genios, los cuales han modificado la forma de vida de toda la sociedad.

Un dispositivo móvil lo podemos definir como un aparato electrónico con capacidad de procesamiento, con dimensiones reducidas pensados para ser utilizados en movilidad. Suelen ser individuales, y cuentan con diversas comunicaciones inalámbricas para cumplir una o varias funciones. [8]

2.1.2.1. *Historia*

En este apartado hablaremos de los dispositivos móviles a lo largo de la historia, observando la evolución realizada hasta llegar a los dispositivos más utilizados actualmente.

2.1.2.1.1. *Inspiración en computadoras*

Este concepto es relativamente reciente, puesto que, al ser dispositivos con capacidad de procesamiento, se inspiraron en las computadoras. Estas surgieron alrededor de 1950, siendo máquinas de varias toneladas y que ocupaban habitaciones enteras. Estas máquinas eran únicas y muy costosas, además de su alta complejidad para programar sobre ellas. Con el paso de los años, debido a la alta aceptabilidad de ellas por parte de compañías privadas, y el avance de la tecnología (transistores, circuitos integrados, microprocesadores...), fueron disminuyendo su tamaño a la vez que aumentaban su velocidad, disminuyendo sus costes y llegando a las masas, siendo un objeto arraigado en la sociedad actual. [9]

2.1.2.1.2. *Inspiración de comunicación telefónica*

Además, los dispositivos móviles cuentan con la capacidad de comunicación inalámbrica, y en su gran mayoría de capacidad telefónica.

Fue en la década de los ochenta cuando se comenzó a explorar los dispositivos móviles. El primero fue el NMT 900, el cual funcionaba con tecnología 1G, la cual no ofrecía transmisión de datos, ya que era analógica, solamente llamadas. [10]

Con el paso del tiempo más empresas comenzaron a crear sus teléfonos móviles, y sobre los noventa se hizo popular el uso del teléfono móvil, gracias principalmente a la inclusión de la conectividad 2G, la cual funcionaba con tecnología digital, abaratando costes, y permitiendo el envío de mensajes cortos, los SMS.

Estos teléfonos móviles fueron mejorando con el avance del tiempo, vía software gracias principalmente a pequeñas interfaces de usuario, las cuales permitían algunas funcionalidades como añadir contactos, pequeños juegos, etc. Además, gozaron de una disminución drástica de dimensiones y peso, al eliminar toda faceta analógica.

Destacar que, en esta década, también surgieron otros dispositivos móviles:

- PDA (personal digital assistant), el cual era un ordenador de bolsillo, en el que se podían realizar algunas tareas que eran impensables en los teléfonos de aquel entonces.
- MP3, pequeños reproductores de música que gozaron de gran popularidad debido a su gran capacidad de canciones y bajo precio. Indicar que ya existían dispositivos portátiles para escuchar música, como el Walkman.

En 2004 surgió el 3G, y con él una mejora en la transmisión de datos, la cual fomentó el uso de contenido multimedia. Comenzaron a incluir cámaras de fotos, pantallas a color con más píxeles, conectividad Bluetooth para transmisión de archivos a corta distancia, permitió la navegación por Internet, sistemas operativos más robustos, con posibilidad de descargar algunas aplicaciones. [11]

En 2007 se presentó el iPhone, dispositivo que cambió la manera de concebir los dispositivos móviles, incorporando una pantalla multitáctil y eliminando el teclado físico, apostando claramente por otorgar al dispositivo más funcionalidades, permitiendo descargar a posteriori aplicaciones nuevas. [12]

Esto cambió el mercado de la telefonía móvil, dando paso a la era del Smartphone, donde los dispositivos móviles se vuelven imprescindibles por la cantidad de funcionalidades que permiten, chatear, acceso a Internet inmediato (gracias a la llegada de tecnologías como el 4G), funciones multimedia de gran calidad...

En estos años también se popularizó el concepto del Tablet, un Smartphone del tamaño de un folio, el cual cuenta con similares capacidades, permitiendo disfrutar del mismo contenido multimedia y aplicaciones, pero en una pantalla bastante mayor.

2.1.2.1.3. *Sistemas operativos*

Los Smartphones a día de hoy se distinguen principalmente por su sistema operativo. Actualmente, son dos los principales sistemas operativos abanderados de los dispositivos móviles, permitiendo la elección al usuario en función de sus preferencias.

En este apartado mostraremos las bondades de cada uno, y posteriormente los compararemos entre sí.

2.1.2.1.3.1. *iOS*

iOS es un sistema operativo de la empresa Apple Inc. exclusivo para sus dispositivos multitáctiles. Fue lanzado en 2007 junto a su primer teléfono móvil, el iPhone. iOS (en aquel entonces denominado iPhone OS) con un sistema muy escueto pero innovador en su momento, donde todos los teléfonos contaban con teclado físico. Apostaba por una interfaz donde los gestos y las animaciones visuales primaban por encima de todo, utilizando principalmente un único botón físico.

Desde ese momento se fue extendiendo al resto de dispositivos móviles que Apple sacó, debido al éxito del mismo, el iPod Touch y el iPad; y se fue actualizando, pasando a contar con una tienda de aplicaciones, la App Store, rediseños, un centro de notificaciones, multitarea, su servicio en la nube iCloud, etc. [13]

2.1.2.1.3.2. *Android*

Android es un sistema operativo basado en Linux para móviles con pantalla táctil. Fue lanzado en 2003 y rápidamente fue comprado por Google en 2005, observando la tendencia del mercado. En 2008 se comercializó el primer dispositivo con sistema operativo Android, el HTC Dream, utilizando la versión 1.0 Apple Pie.

Android se ha caracterizado siempre por ser un sistema operativo open-source, permitiendo a cualquiera su modificación y distribución. Además, utiliza aplicaciones escritas en Java y con facilidades de publicación en Google Play (antes Android Market), la tienda de aplicaciones para Android, lo cual hizo que rápidamente cubriera las necesidades de todos los posibles compradores.

Android fue creciendo exponencialmente a partir de ese momento, y fue puliéndose mediante actualizaciones periódicas que mejoraron la experiencia de usuario, la personalización del dispositivo (característica importante), multitarea, etc. hasta llegar a la última versión a día de hoy, Android 7.1 Nougat. [14]

Destacar que Android no se ha quedado en teléfonos móviles. Son muchos los dispositivos que disfrutan del sistema operativo libre, ya sean tablets, televisores, hasta relojes y consolas.

2.1.2.1.3.3. Comparativa

La comparativa entre estos dos sistemas operativos se resume, en una palabra, libertad. Android se puede utilizar la mayoría de dispositivos, es gratuito crear aplicaciones para él, no tienes limitaciones para introducir contenido en el desde un ordenador...

Por contrario, iOS proporciona un sistema operativo mucho más cerrado, con sus ventajas, todo está más pensado para sus dispositivos: mayor seguridad, aplicaciones de mayor calidad, mejor rendimiento...

También, hay que tener en cuenta que, al ser libre Android, existen más fabricantes y por ende mayor mercado que abarcan. Android cuenta con dispositivos en cualquier franja de precio, mientras que Apple se ha orientado a dispositivos en su mayoría de más calidad, lo cual se refleja en el siguiente gráfico:

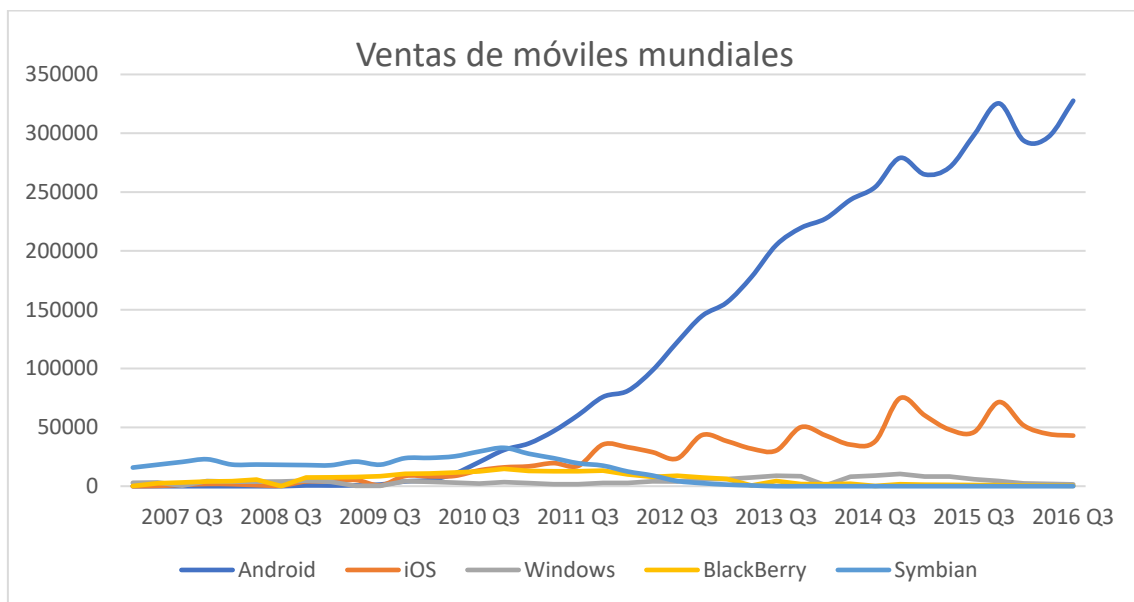


Ilustración 1. Comparativa venta de móviles [15]

Como observamos, Android vende aproximadamente 6 veces más dispositivos que iOS, lo cual desequilibra el porcentaje de usuarios al que se puede acceder hoy en día con una aplicación móvil:

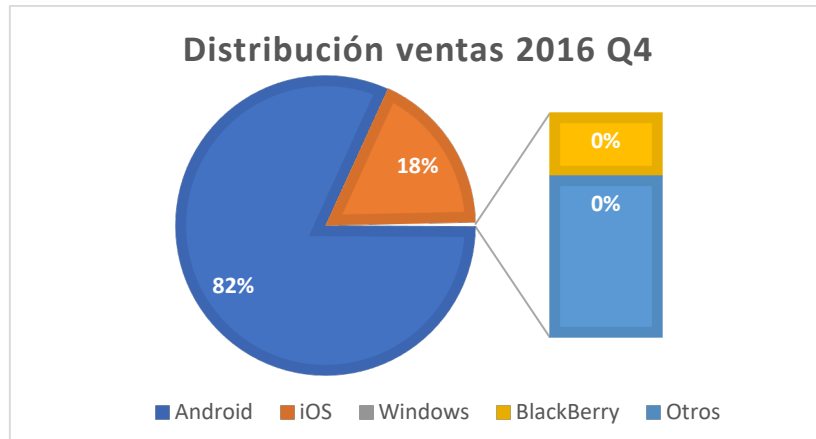


Ilustración 2. Distribución ventas móviles 2016 Q4 [15]

2.1.3. Internet de las cosas

Este término, el IoT, dicen que va a ser el siguiente salto tecnológico que se produzca, del nivel de la Revolución Industrial. Este término fue creado por Kevin Ashton, gurú británico de la tecnología en 1999. Es ahora cuando coge fuerza el término. [16]

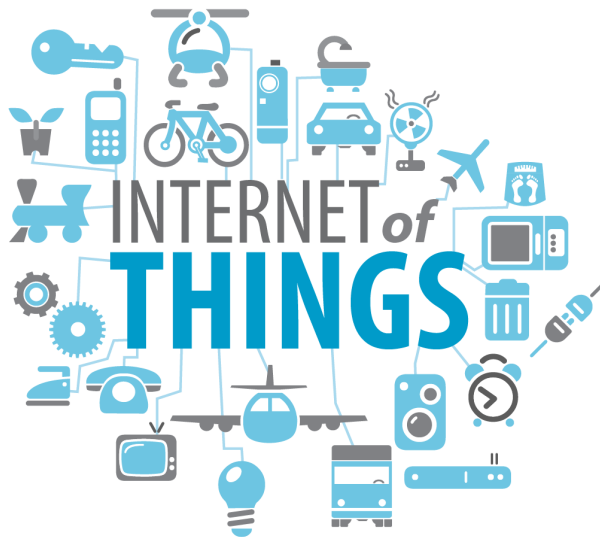


Ilustración 3. Ilustración IoT

2.1.3.1. Definición

El Internet of Things es el concepto que relaciona la interconexión entre un objeto físico cualquiera e Internet. Es decir, cualquier objeto tiene conectividad mediante una dirección IP unívoca y una serie de sensores que le proporcionan utilidades. Cualquier objeto cotidiano puede soportar IoT: un libro, una bicicleta, un depósito... ¿y, para qué? Para cualquier cosa. Recordar en qué página estás, cuán rápido lees; medir el número de pedaladas por minuto, kilómetros recorridos; medir la capacidad utilizada del depósito... y todo esto desde cualquier lugar y al instante. [17]

Gracias a esta nueva forma de concebir el mundo, tenemos acceso a datos que nunca hubiéramos imaginado, y lo más importante, tener la posibilidad de razonarlos, para resolver cualquier tipo de necesidad y aprovecharnos de ellos.

2.1.3.2. Necesidad técnica

Para que el IoT llegue a ser viable, es necesario que la tecnología avance. A día de hoy, se utiliza un estándar de 1983, IPv4, el cual definía una dirección IP como una cadena de 32 bits. Por tanto, da soporte a algo más de 4 mil millones de dispositivos conectados. Esto era más que suficiente por aquella época, pero el auge de Internet ha hecho que desde 2010 se hable de la escasez de direcciones IP (quedaban en aquel entonces menos del último 10% sin asignar). [18]

En la siguiente imagen se observa cuán rápido está aumentando la necesidad de dispositivos conectados.

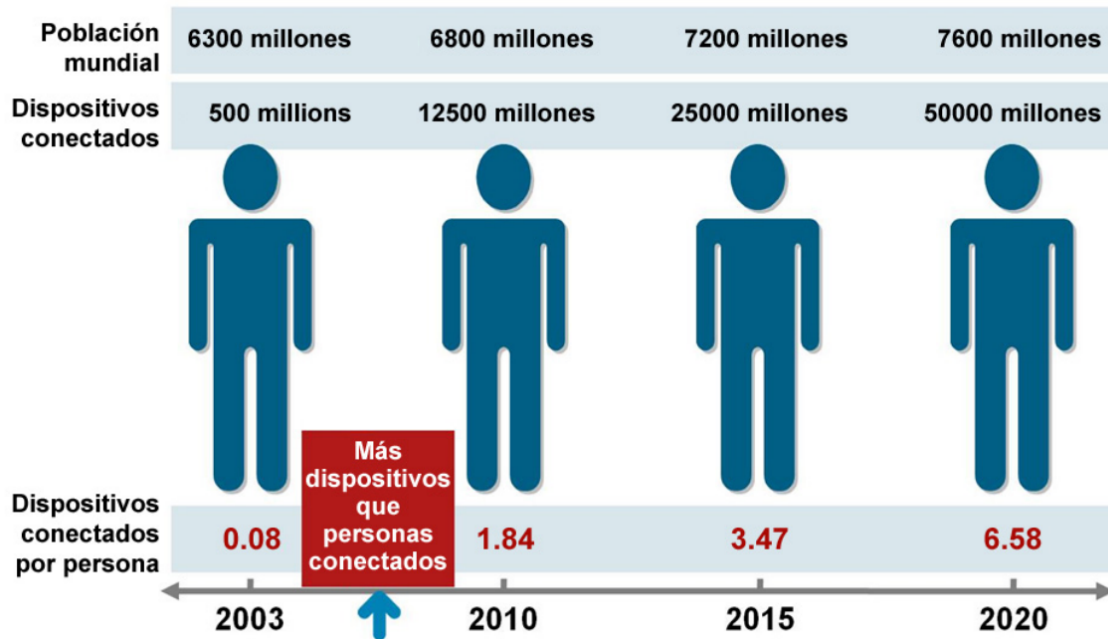


Ilustración 4. Progresión dispositivos conectados por persona [19]

Por tanto, IoT requiere que se aplique el estándar IPv6, el cual da soporte a direcciones IP de 128 bits, más de 340 sextillones de direcciones, una cantidad inmensa para dar soporte a toda la tecnología que acompañe.

Por otro lado, tantos datos de tantos dispositivos requieren almacenamiento rápido y accesible desde cualquier lugar, por lo que el almacenamiento en la nube tiene gran importancia para el desarrollo del IoT.

2.1.4. Arduino

Arduino es una plataforma abierta para la creación de prototipos electrónicos. Consta de dos partes fundamentales: la parte hardware (una placa con un circuito integrado) y la parte software (un entorno de programación para aprovechar las funcionalidades del hardware).

Esta plataforma fue ideada en el instituto Ivrea italiano, en 2005, debido al elevado coste y complejidad para programar los microcontroladores Basic Stamp. Como no todos los alumnos podían permitirse dicha placa, ni tenían tiempo para enseñar electrónica con una placa tan compleja, decidieron construir una placa, cuya finalidad era ser cuanto más barata mejor, y que fuera capaz de funcionar en todos los sistemas posibles, manteniendo un carácter open-source. Debido a estas características, su uso se extendió a lo largo del mundo en multitud de proyectos. [20]

2.1.4.1.1. Hardware

El hardware de la plataforma Arduino se basa en tarjetas electrónicas programables, que utilizan un PCB (Printed Circuit Board) para conectar todos los componentes. El componente principal es un microcontrolador, el cual se encarga de ejecutar el software que se le instale. Otro aspecto importante de las tarjetas Arduino son los pines que tiene en los costados, los cuales son entradas o salidas para la tarjeta, permitiendo ampliar la funcionalidad y customizar el Arduino según las necesidades del proyecto. Todas las placas Arduino cuentan con una interfaz USB para proporcionarle la corriente necesaria para funcionar, y además permite conectarse a una computadora para cargar el software en la placa. [21]

Existen multitud de tarjetas Arduino, en función de las necesidades técnicas de cada proyecto. Los hay con más memoria (de 32 KB a 512 KB), con más entradas físicas, con más pines, con conectividad WiFi... Además, destacar que, aparte de las tarjetas Arduino oficiales, existen multitud de tarjetas similares, compatibles con todos los accesorios y que ejecutan el mismo software que las placas originales. La más conocida, y la que se va a utilizar en el proyecto, es la siguiente:

2.1.4.1.1.1. NodeMCU

NodeMCU es una placa creada en 2013, tras la creación del módulo WiFi ESP8266. Se trata de una placa muy barata, con 128 KB de memoria RAM y 4MB de almacenamiento. Estas características no son novedosas, pero si lo es la inclusión del comentado módulo WiFi directamente integrado en la placa, puesto que le permite acceder a Internet, siendo la herramienta perfecta para desarrollos IoT. Hasta su salida, la única placa con conectividad WiFi era la Arduino Uno junto a un módulo Wifi, cuyo precio ascendía a 100€ Esta placa, cuesta menos de 10€ [22]

Además, cuenta con soporte en el IDE Arduino, por lo que permite cualquier software creado para dicha plataforma open-source, siendo la mejor placa para proyectos IoT actualmente.

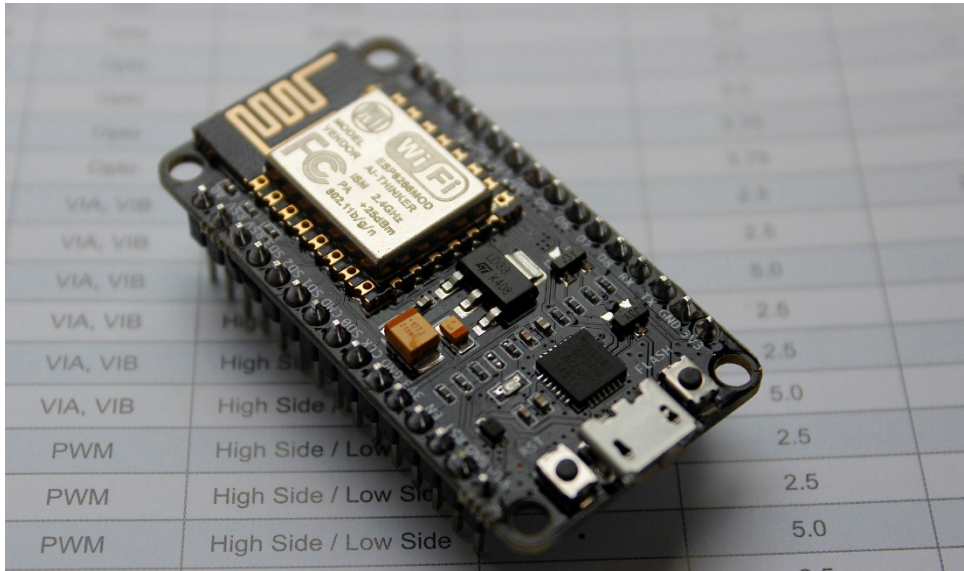


Ilustración 5. NodeMCU

2.1.4.2. Software

Arduino destaca por contar con un entorno de desarrollo IDE (Integration Development Enviroment) desde el cual programar el hardware de la tarjeta utilizada, indicando los pines de entrada / salida a activar, leer / escribir, etc.

La programación del Arduino se realiza bajo el lenguaje de programación Arduino Programming Language, el cual está basado en C, por lo que permite utilizar cualquier construcción de C, y permite añadir multitud de librerías que se ofrecen desde el mismo IDE, facilitando el uso de todos los complementos que se pueden añadir a la placa.

La ejecución de una placa Arduino es muy simple. Consta de dos métodos muy sencillos. El primero de ellos, `setup()`, se ejecuta una única vez, cuando se inicia el dispositivo. Su uso habitual es iniciar los pines a utilizar, así como objetos necesarios próximamente.

El otro método es `loop()`, que, como su propio nombre indica, es un bucle que se está ejecutando constantemente. En él se realiza toda la funcionalidad deseada.

2.2. Funcional

Los sistemas domóticos a día de hoy giran en torno a una única empresa, la cual desarrolla toda una infraestructura para ofrecer su servicio. Es por ello que la mayoría de soluciones de calidad que se encuentran vienen definidas por el pack que contratas.

A continuación, exponemos los principales exponentes en servicios domóticos con aplicaciones móviles, junto al resultado del análisis realizado de cada uno.

2.2.1. Home

La aplicación de Apple [23] busca crear un ecosistema para sus dispositivos iOS, funcionando únicamente con dispositivos HomeKit, el estándar de Apple, que creen otros fabricantes de accesorios domóticos. Esta aplicación está disponible de fábrica en todos los dispositivos con iOS 10.



Ilustración 6. Apple Home

2.2.1.1. Ventajas

- Es una aplicación de Apple, con todo lo que ello conlleva. Calidad, seguridad, integrado perfectamente en su ecosistema de dispositivos, compatibilidad con Siri (su asistente de voz) ...

- Muy configurable. Puedes crear “habitaciones”, conjuntos de dispositivos en cada habitación real de la casa, para controlarlos a la vez. Puedes programar ambientes para cuando estés en casa, cuando salgas, o cuando vayas a dormir.
- Puedes compartir el control de los dispositivos con quien quieras, otorgando determinados permisos a cada invitado.
- Puedes utilizar un dispositivo auxiliar para funcionar como terminal, y controlar la casa desde fuera de ella.

2.2.1.2. Desventajas

Se puede resumir en el precio. Aunque la aplicación es gratuita, viene marcada por:

- La aplicación solo funciona con dispositivos iOS, y como comentamos anteriormente, tienen menos de un 20% de cuota de mercado.
- La aplicación solo funciona con los dispositivos HomeKit, los cuales tienen precios bastante elevados, partiendo de 34.95€ por una bombilla.
- No ofrece todas las funcionalidades si no se cuenta con un dispositivo que funcione como terminal en la casa.

2.2.2. Wink

La siguiente aplicación es Wink [24], promovida por la marca que lleva el mismo nombre. Cuenta con soporte para Android e iOS, y utiliza sus propias centralitas para dar soporte a una gran cantidad de dispositivos, ya sean suyos o de terceras marcas.

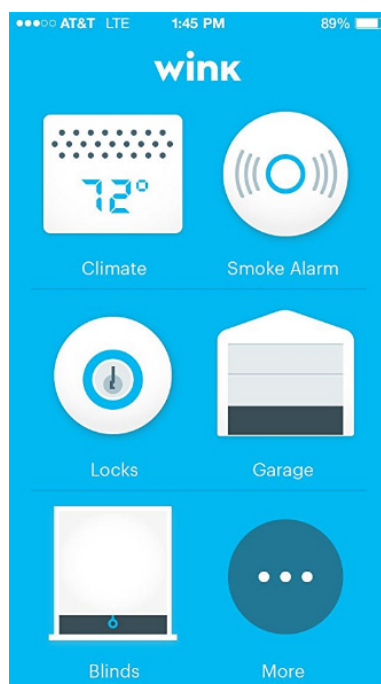


Ilustración 7. Wink

2.2.2.1. Ventajas

- Interfaz intuitiva y minimalista.
- Permite hacer accesos directos, agrupando acciones.
- Permite observar un historial de acciones realizadas.
- Actualizaciones constantes.
- Funciona con bastantes accesorios, de diferentes marcas (incluida la suya).
- Cooperación con Amazon.com, vendiendo sus productos e incorporando su asistente personal 'Alexa' en su sistema, permitiendo así realizar las acciones mediante comandos de voz.

2.2.2.2. Desventajas

- Completamente en inglés.
- No ofrece soporte en España.
- No ofrece la personalización de entornos que otorga la competencia.

2.2.3. Securitas Direct

El sistema de seguridad líder en Europa cuenta con una aplicación para gestionar sus dispositivos de seguridad [25]. No es una aplicación domótica al uso, pero cumple algunas de las condiciones explicadas anteriormente de los hogares domóticos, la seguridad. Esta aplicación permite administrar y controlar los dispositivos insertados en la vivienda.



Ilustración 8. Securitas Direct

2.2.3.1. Ventajas

- Control total sobre el acceso a la vivienda. Permite saber quién ha entrado en la vivienda, a qué hora concreta, etc.
- Administrar la alarma desde la aplicación.
- Utilizar las cámaras de vigilancia instaladas para observar lo que esté ocurriendo en la vivienda.
- Llamada silenciosa para escuchar lo que sucede en la vivienda en caso de emergencia.
- Controlar los cortes de luz, al estar integrada directamente en la vivienda.
- Da soporte a distintas localizaciones con la instalación.

2.2.3.2. Desventajas

- Requiere un contrato mensual con la empresa.
- Únicamente da soporte a dispositivos orientados para la seguridad.

2.2.4. Blynk

Una de las aplicaciones mejor valoradas en ambas plataformas [26], puesto que cuenta con aplicación para Android (100.000 – 500.000 descargas) e iOS.

Esta aplicación está orientada para controlar dispositivos Arduino desde el smartphone.

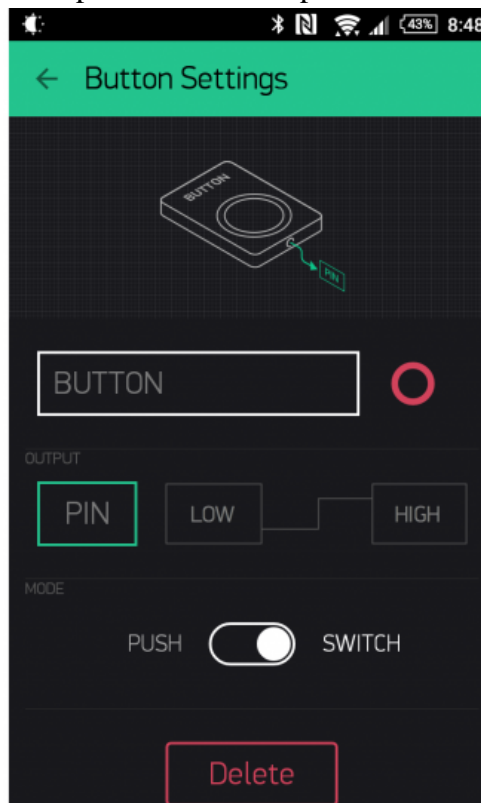


Ilustración 9. Blynk

2.2.4.1. Ventajas

- Gran personalización, permite configurar la interfaz incorporando diferentes widgets al dashboard.
- Configuración muy específica y concreta para dispositivos Arduino, permite fijar a qué pin mandar una determinada señal.
- Posibilidad de compartir el proyecto.
- No requiere de ningún terminal, puesto que cada dispositivo Arduino se comunica directamente con el servidor de la empresa.

2.2.4.2. Desventajas

- No está orientado a un público que busca un producto final, sino más a gente que pretende crear sus proyectos.
- Al no estar orientado a domótica, no cuenta con funcionalidades para ello, como control de varios dispositivos.
- Pagos dentro de la aplicación bastante elevados.

2.2.5. Comparativa respecto a las aplicaciones

Una vez observadas las distintas aplicaciones que tienen un largo recorrido en el mercado, pasamos a compararlas entre ellas, y con la idea de proyecto que tenemos.

Como hemos observado, todas estas aplicaciones están orientadas a controlar dispositivos instalados en una vivienda, pero algunas de ellas tienen conceptos diferentes.

Apple Home y Wink son muy similares, ofreciendo un producto final para controlar dispositivos propios y de terceros, los cuales son de alto coste. Ambas permiten generar conjuntos de estados de varios dispositivos, o programar acciones, mediante una interfaz minimalista y amigable. El “punto débil” de ambas respecto a nuestro proyecto es depender de un dispositivo fijo que funcione como terminal para su control, además de las limitaciones claras, Home no está disponible para dispositivos Android, mientras que Wink no distribuye sus dispositivos terminales en España.

Por otra parte, Securitas Direct ofrece unas funcionalidades muy específicas orientadas a la seguridad. No es comparable a nuestro proyecto.

Finalmente, la última aplicación analizada, Blynk, cuenta con un sistema de gestión de dispositivos similar al requerido para nuestra aplicación, pero como explicamos antes, no está para nada orientada como un producto final, mientras que nuestro proyecto sí.

A continuación, observamos mediante una tabla las ventajas que cumple cada aplicación respecto al proyecto realizado:

<i>Ventajas</i> \ <i>Aplicación</i>	Home	Wink	Securitas Direct	Blynk	Proyecto
<i>Gestión de dispositivos</i>	✓	✓		✓	✓
<i>Bajo coste de dispositivos</i>				✓	✓
<i>Interfaz minimalista</i>	✓	✓	✓		✓
<i>Configuración de dispositivos</i>	✓	✓			✓
<i>Control múltiple de dispositivos</i>	✓				✓
<i>Gestión de localizaciones</i>	✓		✓		✓
<i>Control desde fuera de casa</i>	*		✓	✓	✓

Tabla 2. Comparación aplicaciones domótica

*: requiere un dispositivo en la vivienda

Por tanto, podemos concluir indicando que nuestro proyecto tiene un claro hueco en el mercado, al no tener ninguna alternativa que cumpla las características anteriormente mencionadas.

2.3. Estudio de alternativas de solución

Antes de llevar a cabo el desarrollo de la aplicación, es necesario observar qué estructura montar para soportar la misma, con las condiciones necesarias para su correcto funcionamiento.

En este apartado profundizaremos en las alternativas existentes en cada una de las necesidades que requiramos:

2.3.1. Sistema de gestión y control de dispositivos

La aplicación requiere de un sistema que administre los diferentes dispositivos domóticos que se conecten a él. Para ello, el cliente nos pedía utilizar la solución Thinger, como se explica en el apartado Estudio de la solicitud. A pesar de ello, vamos a estudiar sus ventajas e inconvenientes por si existiera otra solución que se adapte mejor a nuestras necesidades.

2.3.1.1. Crear un sistema propio

Esta solución proporcionaría a la aplicación de un sistema perfectamente adaptado para la aplicación, dando soporte a todos los dispositivos deseados y contar con las características idóneas para la misma.

Ventajas	Desventajas
Infraestructura propia, no depende de terceros.	Alta inversión, tanto económica como temporal.
Facilidad para implementar mejoras.	Dificultades añadidas a la solución, siendo más complejo que el proyecto en sí.
	El cliente no desea esta solución

Tabla 3. Ventajas / desventajas sistema propio

2.3.1.2. Thinger



Ilustración 10. Imagen Thinger

Este servicio [27] permite a cualquier usuario gestionar su dispositivo IoT y controlarlos manualmente mediante su página web, o mediante peticiones API REST. Cuenta con distintas tarifas en función del uso que se haga de sus funciones, con un servicio gratuito.

Además, se puede utilizar para generar gráficas (Dashboards) o como un almacén de datos tomados por los dispositivos, para su posterior procesamiento.

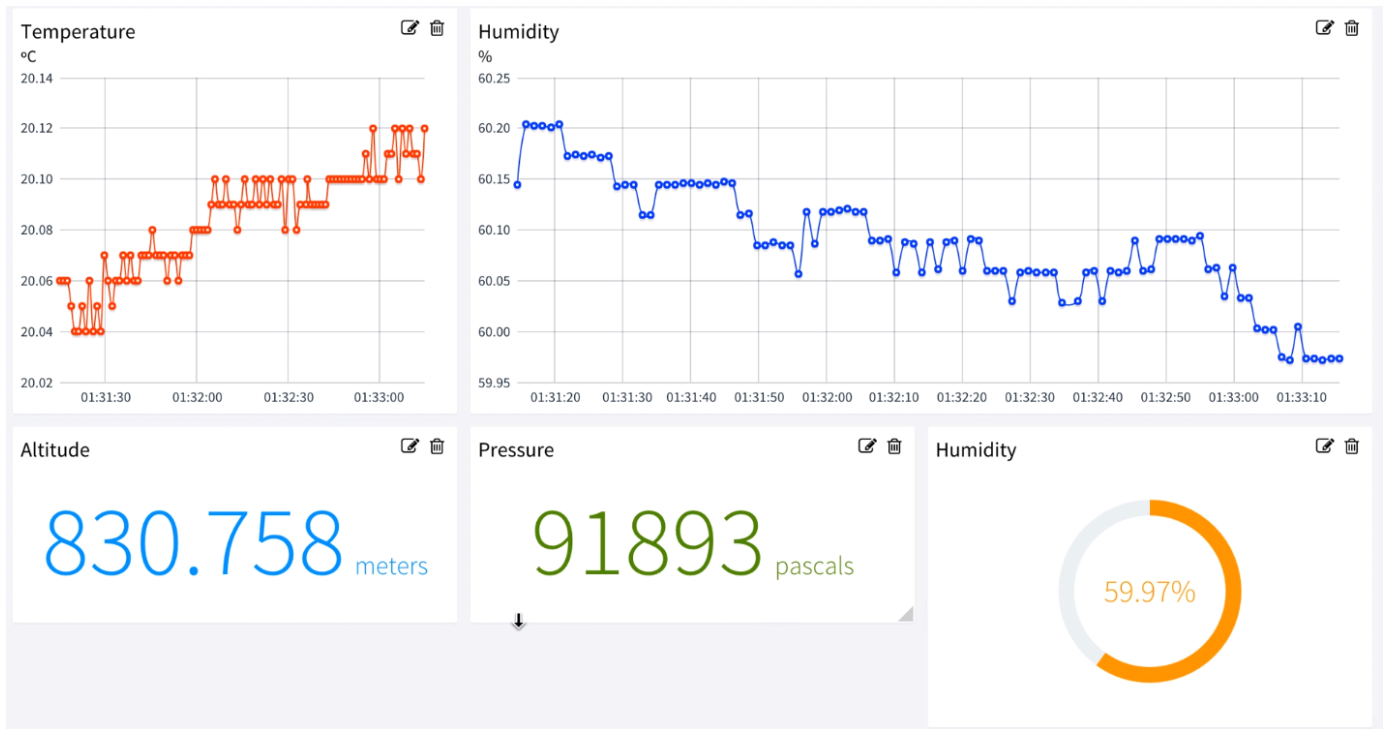


Ilustración 11. Dashboards Thingier

Ventajas	Desventajas
Infraestructura adecuada para el proyecto deseado.	De pago al aumentar el número de dispositivos
Facilidad de uso.	Dificultad para implementar nuevas funcionalidades, al depender de un sistema externo
El cliente desea esta solución	

Tabla 4. Ventajas / desventajas Thingier.

2.3.1.3. Comparativa

Esta comparativa está claramente a favor de utilizar el sistema Thingier para gestionar los dispositivos, por ser un sistema solvente para el cometido del proyecto, y por las exigencias del cliente. Además, es evidente la dificultad de implementar un sistema para ello de cero, ni es el cometido del proyecto.

Por todo lo expuesto, se decide finalmente utilizar Thingier.io como sistema de gestión y control de dispositivos.

2.3.2. Base de datos y servidor

La aplicación requiere de una base de datos disponible las 24 horas del día en la cual almacenar toda la información de los usuarios, y por ello necesitamos un servidor el cual permita el servicio a la base de datos.

2.3.2.1. Base de datos MySQL + Servidor

Esta solución nos permitiría tener un control total sobre la infraestructura utilizada y adaptarla a nuestras necesidades.

Esta base de datos se consultaría constantemente por cada usuario, por lo que habría que utilizar un servidor de garantías para satisfacer a todos los usuarios en tiempo real sin problemas.

Ventajas	Desventajas
Infraestructura propia, no depende de terceros.	Alta inversión inicial, tanto económica como temporal.
	Dificultades añadidas al proyecto: montaje del servidor, creación de la base de datos...

Tabla 5. Ventajas / desventajas base de datos MySQL + Servidor

2.3.2.2. Google Firebase

Este servicio de Google [28] pone a disposición de cualquiera una infraestructura completa sobre la cual apoyar una aplicación.

Además del servidor con una base de datos online, Firebase ofrece una gran cantidad de funcionalidades extra muy útiles para el desarrollo de cualquier aplicación, como un sistema de autenticación fiable, almacenamiento en la nube, etc.

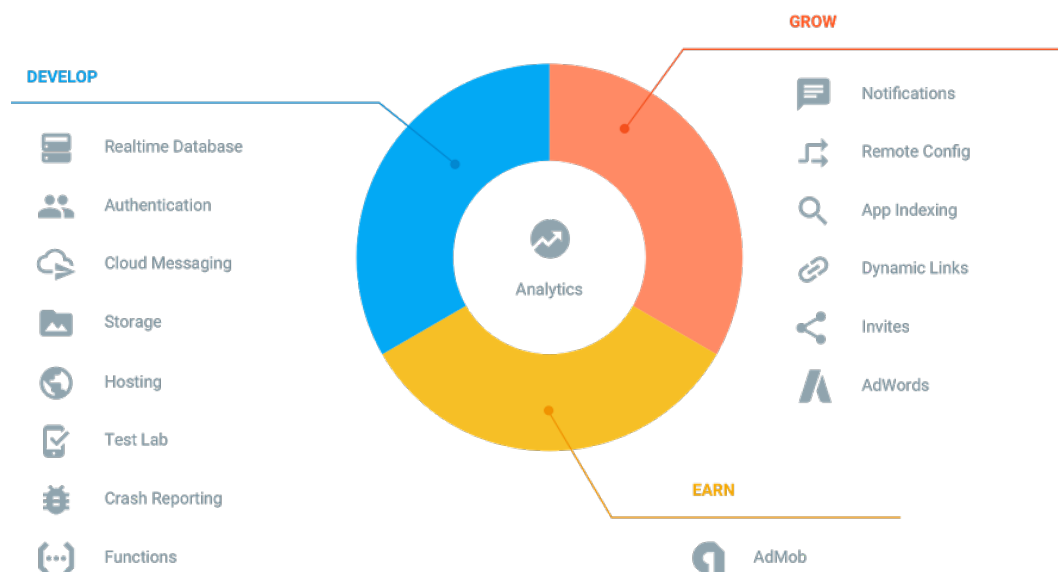


Ilustración 12. Imagen Google Firebase

Todas estas funcionalidades se utilizan con una API completa disponible para dispositivos Android, iOS y servicios web, por lo que permite integrar a todos los usuarios en la misma estructura.

Google Firebase tiene una política de precios adaptable, ofreciendo tres planes diferentes y variables, en función del éxito de la aplicación:

- El gratuito, ofreciendo un máximo de 100 conexiones simultáneas, 1GB de almacenamiento y un máximo de 10GB de transferencia. El paquete idóneo para comenzar y realizar pruebas sobre una aplicación.
- El siguiente paquete ofrece, por 25\$ al mes, ilimitadas conexiones simultáneas y un límite de memoria mayor. Está destinado a aplicaciones en crecimiento.
- Y el último es para aplicaciones asentadas, las cuales pagan en función de su consumo.

Ventajas	Desventajas
Infraestructura de Google, se espera alta calidad.	Bases de datos algo simples.
Gratuito inicialmente, perfecto para este tipo de desarrollo.	Límite de conexiones simultáneas algo bajo (100 clientes).
Posibilidad de expansión rápida. Otras plataformas, herramientas funcionales disponibles para su uso...	
Alta integración con Android y las cuentas de Google.	

Tabla 6. Ventajas / desventajas Google Firebase

2.3.2.3. Amazon Web Services (AWS)

Otro servicio de infraestructura preparada para aplicaciones. Este servicio de Amazon [29] proporciona multitud de herramientas útiles para la aplicación. En nuestro caso, nos centraríamos en la base de datos DynamoDB, así como en el servicio de cuentas, Cognito.

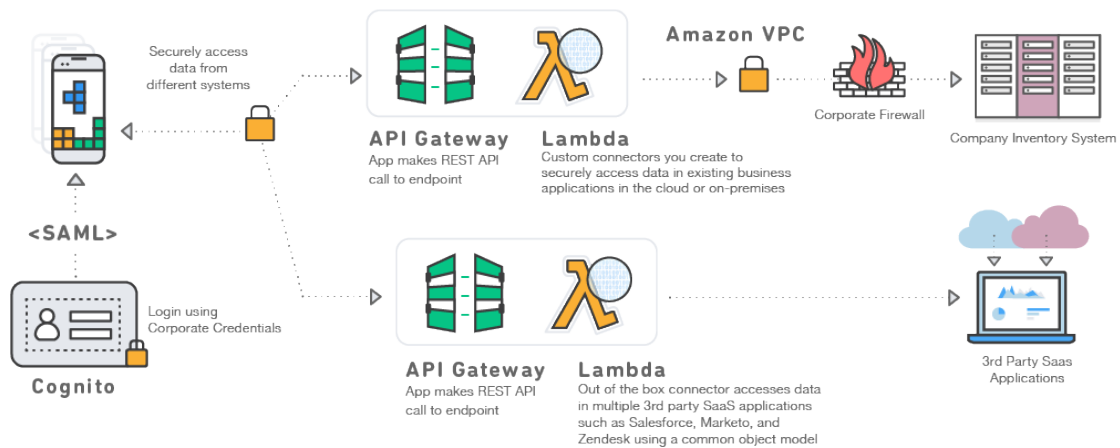


Ilustración 13. Imagen Amazon Web Services (AWS)

AWS permite integrar todos los sistemas perfectamente, y adaptarse a diferentes entornos (iOS, web, aplicaciones de terceros...), con total control sobre el sistema.

Además, cuenta con un SDK para desarrollar aplicaciones con acceso a todos los servicios de AWS.

AWS ofrece una facturación muy personalizada, en función de los recursos que se empleen, otorgando precios muy competitivos a casi cualquier aplicación. Además, cuentan con un entorno gratuito, en el que ofrecen cada uno de sus servicios con su limitación, el que nos interesaría a nosotros, serían 25GB en su base de datos.

Ventajas	Desventajas
Infraestructura de Amazon, se espera alta calidad.	Inexperiencia con esta plataforma
Gratuito inicialmente, perfecto para este tipo de desarrollo.	Mayor complejidad a la hora de utilizar Android Studio
Posibilidad de expansión rápida. Otras plataformas, herramientas funcionales disponibles para su uso...	

Tabla 7. Ventajas / desventajas Amazon Web Services (AWS)

2.3.2.4. Comparativa

Es importante la decisión a tomar para el desarrollo del proyecto, por lo que se debe realizar un análisis extensivo.

En primer lugar, nos encontramos que tanto la solución de Google como la de Amazon nos eliminan costes económicos, y además nos proporcionan tiempo respecto a la otra solución. Por ello, decidimos desechar la opción de montar nosotros el sistema.

Ahora decidimos entre las dos infraestructuras. Cualquiera que elijamos será correcta, ya que estamos hablando de sistemas con años de experiencia y soportando aplicaciones de nombre mundial.

Según el estudio realizado, Amazon Web Services está más dedicado e implantado en soluciones para complejas que Firebase, contando con muchas soluciones que no se utilizarán. Además, Firebase tiene una ventaja agregada y es su alta integración con Android, recordamos, sistema operativo de la propia Google.

Además, es reseñable el hecho de la experiencia previa que se tiene sobre Firebase para esta decisión, lo cual disminuirá tiempos de comprensión con la infraestructura y permite producir más rápido.

Por todo lo expuesto, se decide finalmente utilizar Google Firebase como base de datos e infraestructura del proyecto.

3. Gestión y organización del proyecto

En este apartado se va a explicar el funcionamiento del equipo de desarrollo de este proyecto. Para ello, se va a detallar la metodología utilizada y los roles ocupados por miembros del equipo.

3.1. Metodología de desarrollo software utilizada

Una metodología bien aplicada es clave para cumplir con la calidad y el tiempo exigidos por un cliente. En este caso, el grupo, ha decidido apostar por el método de Craig-Larman, debido a su conocimiento adquirido durante el grado. Destacar que esta metodología se adaptará a las necesidades de este proyecto, simplificando algunas de sus tareas.

3.1.1. Explicación Craig-Larman

Craig-Larman es un método basado en una versión reducida del Racional Unified Process (RUP). Este cuenta con diversos caminos a seguir para construir software. Lo que hace el método de Larman es explicar detenidamente uno de esos caminos, haciendo más sencilla la aplicación del método.

El método de Larman se basa en un único ciclo de vida, cuyas características son: [30]

- Es iterativo, puesto que se divide el proyecto en N trozos, que se van abordando por separado. Por tanto, realizamos análisis, codificación y pruebas por separado. Esta separación se realiza según casos de uso.
- Es incremental: cada uno de los trozos del proyecto se realiza sobre los anteriores, obteniendo cada vez un producto más funcional. Gracias a ello podemos enseñarle funcionalidad al cliente de manera rápida, y este es capaz de percibir el avance.
- Está dirigido por los casos de uso. Trabajamos a partir los casos de uso, los cuales se van transformando para finalmente llegar a funcionalidad en el código.

El proceso de Craig-Larman cuenta con 3 grandes fases:

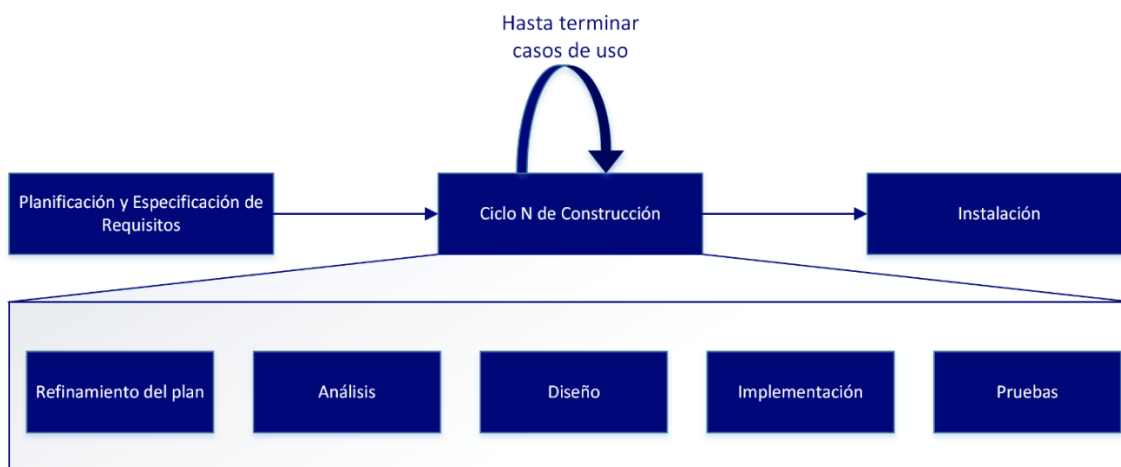


Ilustración 14. Fases Craig-Larman

3.1.2. Explicación del procedimiento a seguir

A continuación, se explica el procedimiento a seguir, siguiendo las pautas de la metodología seleccionada, y adaptándola para realizar el proyecto con viabilidad por un único miembro.

3.1.2.1. Fase 1. Planificación y especificación de requisitos

Se comienza realizando la planificación y especificación de requisitos. Esta fase se realiza una única vez, en donde se realizan las siguientes sub-tareas:

- Estudio de viabilidad del sistema. Observar si es viable el proyecto. Se realiza una primera planificación estimada, así como un aproximado coste del proyecto, para que el cliente lo valide.
 - En este proyecto, no es necesario, puesto que el proyecto está validado desde un primer momento. Pese a ello, se realizarán los apartados de planificación estimada y coste aproximado, para comparar finalmente si se cumplieron los plazos estimados.
- Definir los requisitos del sistema. Un requisito es descripción atómica de necesidad respecto a un producto, donde se describe cierto aspecto de funcionalidad del producto esperado. En resumen, qué hará el sistema. Para especificarlos, se utilizará el estándar IEEE 830.
- Implementar prototipos, para mostrárselo al cliente y obtener su feedback. No es obligatorio si se conoce perfectamente el objetivo.
- Definir casos de uso: un caso de uso refleja un conjunto de requisitos, describiendo la interacción que crea el usuario con el sistema.
- Realizar el diagrama de casos de uso, para comprender con un vistazo la funcionalidad del sistema.
- Priorizar: una vez se tienen todos los casos de uso, se crea un orden de preferencia para ser desarrollados, siguiendo una serie de factores relativos al proyecto. Además, hay que tener en cuenta si existen casos de uso que dependen de otros, para desarrollar primero aquellas funcionalidades que no requieren de otros casos de uso.

3.1.2.2. Fase 2. Construcción

Una vez las ideas están claras, y se cuenta con todos los grupos de casos de uso definidos, se comienza la fase de construcción. Esta se realiza N veces, tantas como grupos de casos de uso haya en el proyecto (cada mini-proyecto). Cada ciclo se compone, a su vez de:

- Refinamiento del plan: sirve para revisar si algún elemento anterior no cuadra con los cambios realizados en cada iteración, como por ejemplo la planificación, se puede ver modificada tras realizar un ciclo.
- Análisis: su finalidad es obtener una mejor comprensión del problema centrado en los casos de uso de esta iteración. Para ello, se realizarán los siguientes pasos:
 1. Creación de casos de uso en formato expandido.

2. Refinar casos de uso, así como el diagrama de casos de uso.
 3. Definir diagramas de actividad del sistema.
- Implementación por parte de los programadores.
 - Pruebas a realizar por los programadores. A cada caso de uso se le realizan sus pruebas hasta tener el nivel deseable. De no alcanzarse, se continúa refinando hasta conseguirlo.

3.1.2.3. Fase 3. Entrega

Finalmente, se entrega del producto al cliente. Para ello se generan los diversos documentos necesarios para su utilización, en este caso, el manual de uso y guía de instalación.

3.2. Organización del proyecto

En este apartado vamos a definir los recursos humanos necesarios para el desarrollo del proyecto, así como los roles generados dentro del equipo. Debido a ser un proyecto unipersonal, un miembro podrá desempeñar más de un rol en el mismo.

3.2.1. Roles



Ilustración 15. Roles del proyecto

El nivel más alto en la jerarquía establecida en este proyecto es el cliente. Este rol será representado por el tutor Miguel Ángel Patricio. Se encargará de transmitir lo mejor posible su idea original al jefe de proyecto, y comprobará la evolución del proyecto en

cada una de las reuniones a realizar tras terminar los grupos de casos de uso, observando como aumenta la funcionalidad hasta conseguir el producto final.

A continuación, se encuentra el jefe de proyecto, representado por Daniel Amigo. Será el encargado de reunirse con el cliente y obtener la información que contemple la solicitud del software del cliente para que el equipo comience a desarrollar. Además, es el responsable de que los plazos del proyecto se cumplan a tiempo, dirigiendo al grupo en todo momento.

Una vez se tiene toda la información del cliente, los analistas la recogen y realizan la especificación de requisitos, utilizando distintas técnicas para ello. Posteriormente, realizan todos los diagramas necesarios, así como la planificación del proyecto.

Una vez los analistas tienen priorizados los casos de uso, comienza el programador a desarrollar la aplicación. Para ello, utilizará los recursos explicados posteriormente, y finalmente, realizará las pruebas de cada conjunto de casos de uso.

3.2.2. Recursos humanos

A continuación, se adjunta una versión compacta del currículum del único miembro del equipo, el cual realiza los roles de jefe de proyecto, analista y programador.

Daniel Amigo Herrero	
Fecha de nacimiento:	15/09/1995
País de nacimiento:	España
Estudios	
	<ul style="list-style-type: none"> • Actualmente cursando Ingeniería Informática por la Universidad Carlos III de Madrid (2013 – 2017) • Bachillerato en Ciencias y Tecnología. IES Valmayor (2011 – 2013) • Educación Secundaria Obligatoria. IES Valmayor (2007 – 2011)
Conocimientos	
	<ul style="list-style-type: none"> • Mente abierta y organizada, involucrándose al 110% en los desarrollos. • Atento a los pequeños detalles.

Tabla 8. Currículo Daniel Amigo

3.2.3. Recursos hardware

El desarrollo de este proyecto se ha realizado íntegramente en la siguiente computadora:

Nombre	Lenovo Y50-70
Procesador	Intel Core I7-4710HQ 2.5GHz. 4 núcleos, 8 hilos
Memoria RAM	16 GB
Procesador gráfico	Nvidia 860M
Sistema Operativo	Windows 10 64 bits
Pantalla	15.6'' 1920 · 1080 píxeles

Tabla 9. Recurso hardware. Portátil

En cuanto al dispositivo móvil en el que se realizarán todas las pruebas, estas son sus características:

Nombre	OnePlus One
Procesador	Snapdragon 801
Memoria RAM	3 GB
Procesador gráfico	Adreno 330
Sistema Operativo	Android 6.1 Marshmallow
Pantalla	5.5'' 1920 · 1080 píxeles

Tabla 10. Recurso hardware. Teléfono Móvil

Por último, el dispositivo Arduino a utilizar en las pruebas:

Nombre	NodeMCU
Procesador	32 bits a 80MHz
Memoria RAM	35K
Memoria Flash	1Mb
Conectividad	WiFi, Micro-USB

Tabla 11. Recurso hardware. Arduino

3.2.4. Recursos software

En este apartado se detallará el software utilizado para el desarrollo de este proyecto, realizando una pequeña introducción a los programas más específicos:

3.2.4.1. Microsoft Office

Para el desarrollo de toda la documentación se ha utilizado la suite de Microsoft. Se ha optado por ella debido al hábito del equipo a estas herramientas, tras utilizarlas desde la universidad. A continuación, se especifica el uso de cada herramienta:

- Microsoft Word 2016: para el desarrollo de la presente memoria. Redacción del documento, estilo, formato, elaboración de índices, etc.
- Microsoft PowerPoint 2016: para el desarrollo de la presentación de la defensa del proyecto.
- Microsoft Visio 2016: para el desarrollo de ilustraciones y diagramas presentes en la memoria y en la presentación.
- Microsoft Project 2016: para la realización de distintos diagramas, como la planificación.

3.2.4.2. Android Studio

La herramienta más importante para la elaboración del proyecto. Android Studio el entorno de desarrollo integrado (IDE) oficial de Google para el desarrollo de aplicaciones para Android. Cuenta con todas las funcionalidades necesarias para cualquier desarrollador de aplicaciones Android, con características como:

- Acceso directo al SDK Manager, para descargar la versión deseada del SDK de Android (Software Development Kit).
- Acceso a emuladores para probar las aplicaciones.
- Estructura de proyecto simplificada para acceder a los distintos tipos de código creados.
- Compleción de código inteligente.
- Soporte para Gradle, una automatización del proceso de compilación, que facilita la inclusión de librerías y la construcción de la aplicación.
- Soporte para el desarrollo de todo el código necesario, ya sean clases JAVA, layouts en XML, ficheros JSON, etc.

3.2.4.2.1. Estructura de proyecto

Android Studio permite personalizar la vista del proyecto, para que nos muestre los archivos necesarios en cada momento. La vista principal es la denominada Android, que se muestra a continuación:

El proyecto se compone de tres carpetas principales:

- Manifests: Contiene el archivo AndroidManifest.xml, el cual indica la información esencial de la aplicación, como el nombre, las distintas vistas que tiene, los permisos, etc.
- Java: En esta carpeta se contiene todo el código fuente que realiza la funcionalidad deseada. Además, se incluyen aquí el código de prueba en JUnit.
- Res: Todos los recursos de la aplicación. Las layouts de la interfaz, los colores, los textos con sus traducciones, las listas de elementos, todo el contenido se almacena aquí.

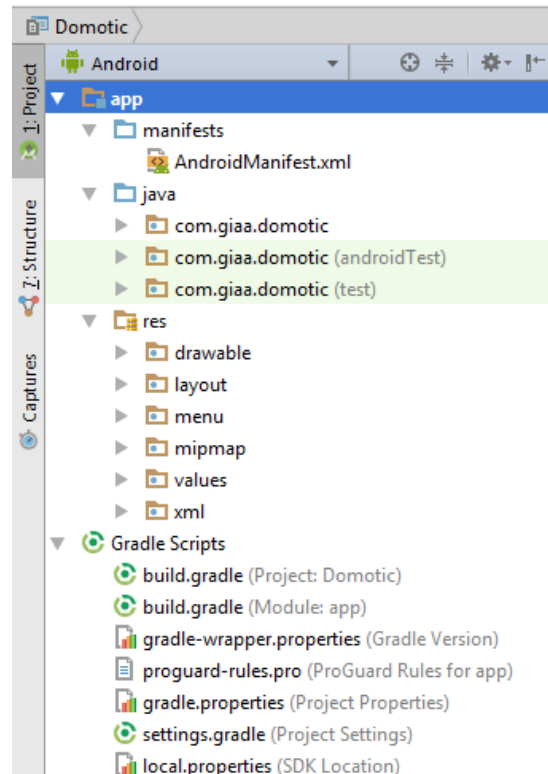


Ilustración 16. Estructura de proyecto en Android Studio

Finalmente, observamos, fuera del proyecto, tenemos Gradle Scripts, donde destacan los dos archivos build.gradle:

- Project: define todas las configuraciones de compilación a aplicar en todos los módulos del proyecto.
- Module: define las configuraciones de compilación a aplicar en este módulo concreto, en este caso, la aplicación. En este archivo se incluyen los enlaces de las librerías a utilizar en el proyecto.

3.2.4.3. Arduino

Para la configuración y pruebas de dispositivos de domótica, se ha utilizado el IDE Arduino para el desarrollo del software a utilizar en la placa NodeMCU.

3.2.4.4. Google Drive

Con el fin de tener una copia de seguridad de garantías, para almacenar todos los ficheros creados, las reuniones con el cliente, documentación relevante sobre el trabajo, distintas versiones del código realizado para la aplicación y el NodeMCU, se ha decidido utilizar

Google Drive, por su conocimiento sobre la aplicación obtenido durante los años en el grado y el almacenamiento ilimitado que ofrece a los alumnos del mismo.

3.2.4.5. TimeCamp

Se va a utilizar TimeCamp para la medida del tiempo trabajado en cada tarea del proyecto. Este recurso es necesario pues, al final del proyecto, es necesario realizar el presupuesto final del proyecto, y para ello calcular las horas hombre realizadas por los distintos roles del equipo.

Esta herramienta, de pago para grupos de más de una persona, permite registrar la actividad que está realizando cada miembro del grupo, utilizando una aplicación instalada en el dispositivo de uso. De esta manera, cada aplicación que se utilice se almacena, junto a su correspondiente tiempo de uso, en la cuenta del usuario.

Al ser una aplicación menos conocida en este ámbito, pasamos a explicar el uso que se ha realizado de la misma.

3.2.4.5.1. Uso realizado

Una vez termina el día, el usuario se mete en la web de TimeCamp, para crear un nuevo proyecto, el cual es sub-proyecto del proyecto TFG 2017. Este sub-proyecto tiene como título la fecha y la tarea realizada, y almacenando manualmente las distintas tareas realizadas de labor productiva en el nuevo proyecto.

Al terminar un día, el resultado será una entrada como la siguiente:

Activity	Duration	Start Time	End Time
studio64 - Domotica - [C:\Users\Daniel\AndroidStudioProjects\Domotica] - [app] - ...\app\src\main\java\com\tfg\daniel\domotica\LogInActivity.java - A...	2h 25m	00:00	21:24
console.thinger.io (Thinger.io - Cloud Console - Google Chrome, Android app variable management issue - Cloud Integration - The Internet of Thinger - ...	24m 26s	13:46	20:31
Java (- sketch_mar19a Arduino 1.6.12, Moviendo, Error, Alerta, Salvar carpeta de programas como..., - v0_Conecto_mi_arduino_a_Thinger Arduino 1.6.12,...	16m 35s	13:53	19:25
console.firebase.google.com (Firebase Console - Google Chrome, datos usuario firebase - Buscar con Google - Google Chrome, User Authentication - Fi...	16m 28s	00:07	12:12
firebase.google.com (Autenticación mediante el uso de Google Sign-In en Android Firebase - Google Chrome, Manage Users Firebase - Google Chr...	13m 14s	00:00	11:54
docs.thinger.io (Thinger.io Documentation - Open Source Internet of Things - Google Chrome, Thinger.io Server Deployment Documentation - Open So...	10m 26s	13:49	17:31
stackoverflow.com (How do I link each user to their data in Firebase? - Stack Overflow - Google Chrome, swift - How do I check if a firebase database val...	7m 06s	00:08	18:16
youtube.com (firebase user data - YouTube - Google Chrome, Firebase Data Modeling + User Management - YouTube - Google Chrome, Android Studio...	5m 49s	09:03	17:42
firebase.com (Understanding Data - Firebase - Google Chrome, Saving Data - Firebase - Google Chrome, Android Platform - Firebase - Google Chrome, ...	5m 48s	09:26	09:50
developer.android.com (request get android - Buscar con Google - Google Chrome, Sending a Simple Request Android Developers - Google Chrome, ...	5m 37s	17:25	18:02
hermosaprogramacion.com (Volley: Librería De Android Para Realizar Peticiones Http - Google Chrome, HttpURLConnection En Android Como Cliente Htt...	5m 24s	17:31	20:00
github.com (quickstart-android/auth at master · firebase/quickstart-android · GitHub - Google Chrome, Docs/README.md at gh-pages · thinger-io/Docs...	4m 36s	00:17	18:24
gist.github.com (empty database firebase - Buscar con Google - Google Chrome, Firebase: Detecting if data exists. This snippet detects if a user ID is alre...	3m 04s	10:05	18:09
thinger.io (Thinger.io - Open Source IoT Platform - Google Chrome, https://thinger.io - Google Chrome, Products - Thinger.io - Google Chrome, https://...	2m 34s	13:46	20:36
(unassigned computer activities)	6h 27m	00:01	23:08

Ilustración 17. Entrada TimeCamp

Como podemos observar, ese día producimos 4 con 26 horas de TFG, destinando 2 horas 25 minutos a Android Studio, 24 minutos a la consola de Thinger, etc.

Finalmente, contamos con una lista en el proyecto TFG 2017 con gran cantidad de datos, y todos ellos se pueden exportar a formato Excel (característica de pago), para el posterior uso de los datos.

NAME	HOURS WITHOUT SUBTASKS	HOURS WITH SUBTASKS
TFG 2017	-	51h 26m
03-21-2017 TFG Desarrollo	4h 49m	4h 49m
03-22-2017 TFG FAB y rehago interfaz	4h 46m	4h 46m
03-19-2017 Conectar Arduino LED	4h 26m	4h 26m
03-17-2017 TFG Log in	4h 13m	4h 13m
03-29-2017 Hacer fragment device	3h 59m	3h 59m
03-25-2017 TFG Nuevo dispositivo	3h 21m	3h 21m
03-31-2017 TFG Un poco de todo	3h 8m	3h 8m
03-20-2017 TFG Desarrollo	3h 6m	3h 6m
01-25-2017 Haciendo Memoria en Biblioteca	3h 6m	3h 6m
03-18-2017 Log in y Database	2h 39m	2h 39m
01-22-2017 Definiendo Memoria TFG	2h 31m	2h 31m

Ilustración 18. Lista tiempos TimeCamp

3.2.4.6. Axure

Axure es una herramienta de prototipado y wireframing profesional, muy sencilla de utilizar y dirigida a aplicaciones web, pero cuenta con librerías para ser utilizada para prototipos móviles. Permite realizar prototipos arrastrando los objetos (cuadros, textos, botones...) en cada pantalla, construyendo la estructura de la aplicación. Además, permite generar pequeños cambios cuando se producen eventos simples, permitiendo generar una pequeña funcionalidad.

3.2.4.7. Adobe Acrobat Pro DC

Adobe Acrobat es la aplicación profesional más conocida para la creación de documentos en formato PDF. Se utilizará en el proyecto para la maquetación final de la memoria, para dotarla de hipervínculos internos y externos, así como proteger el documento de ediciones.

3.2.4.8. Visual Paradigm

Visual Paradigm es una de las aplicaciones líder en desarrollo de diagramas UML. En nuestro caso, lo utilizaremos para el desarrollo de los diagramas de actividad relacionados con la funcionalidad de la aplicación.

4. Planificación y especificación de requisitos

4.1. Identificación de requisitos

En este punto se van a especificar los requisitos del sistema. Se ha seguido el estándar IEEE 830 para su especificación. Todos los requisitos seguirán la siguiente plantilla:

Identificador del requisito	Identifica de manera unívoca a cada requisito, mediante un acrónimo y un número X. Los acrónimos son: <ul style="list-style-type: none"> • RUC.X: Requisito de usuario de capacidad. • RUR.X: Requisito de usuario de restricción. • RF.X: Requisito funcional. • RNF.X: Requisito no funcional.
Nombre del requisito	Indica con dos o tres palabras el contenido del requisito.
Tipo	Indica si es un requisito o una restricción.
Fuente del requisito	Indica quién ha definido dicho requisito, quien manda realizar dicha funcionalidad. Cliente / Analista.
Prioridad del requisito	Establece la prioridad de llevar a cabo dicho requisito dentro del proyecto. Toma los siguientes valores: Esencial / Deseado / Opcional
Descripción del requisito	Contiene una descripción detallada sobre el requisito.

Tabla 12. Estructura requisitos

4.1.1. Procedimiento a seguir

Estos requisitos son los encargados de delimitar qué debe hacer el sistema. Suelen definirse como una descripción de acciones que puede realizar el usuario dentro del sistema. Dentro de ellos se distinguen dos tipos:

- Requisitos de capacidad (RUC): definen la capacidad de la aplicación, siempre desde la perspectiva de la funcionalidad a realizar por el usuario.
- Requisitos de restricción (RUR): este tipo de requisitos especifican las limitaciones del sistema para realizar las funciones. Suelen responder a la pregunta ¿cómo lo hace?

Una vez con la lista de requisitos de usuario terminada y validada, se define la lista de requisitos software. Esta es una versión refinada de los requisitos de usuario, realizada con lenguaje técnico y orientada desde el punto de vista del sistema, de modo que los programadores entiendan y plasmen perfectamente el objetivo del requisito en el sistema.

Los requisitos de software se dividen en:

- Funcionales (RSF): definen el comportamiento del software a realizar y sus funcionalidades.

- No funcionales (RSNF): se enfocan al diseño o la implementación del sistema.

Por último, destacar que no tiene por qué haber el mismo número de requisitos de software que en la lista de requisitos de usuario, puesto que puede ser necesario otros requisitos para garantizar que el sistema cumpla los criterios definidos.

4.1.2. Requisitos de usuario

4.1.2.1. Requisitos de capacidad

Número del requisito	RU.01
Nombre del requisito	Registro de usuario
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz de registrarse en la aplicación utilizando su cuenta de Google.

Tabla 13. Requisito de usuario 01

Número del requisito	RU.02
Nombre del requisito	Cerrar sesión de usuario
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz de cerrar su sesión en la aplicación.

Tabla 14. Requisito de usuario 02

Número del requisito	RU.03
Nombre del requisito	Revocar acceso usuario
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz de eliminar toda información registrada en su cuenta por la aplicación.

Tabla 15. Requisito de usuario 03

Número del requisito	RU.04
Nombre del requisito	Añadir dispositivo
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz de añadir un dispositivo a su cuenta.

Tabla 16. Requisito de usuario 04

Número del requisito	RU.05
Nombre del requisito	Gestionar dispositivo
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz de modificar o eliminar un dispositivo de su cuenta.

Tabla 17. Requisito de usuario 05

Número del requisito	RU.06
Nombre del requisito	Utilizar dispositivo
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz de regular los parámetros de su dispositivo, así como observar sus sensores.

Tabla 18. Requisito de usuario 06

Número del requisito	RU.07
Nombre del requisito	Añadir escenas
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz de crear escenas a partir de los dispositivos añadidos a su cuenta.

Tabla 19. Requisito de usuario 07

Número del requisito	RU.08
Nombre del requisito	Gestionar escena
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz de modificar o eliminar una escena.

Tabla 20. Requisito de usuario 08

Número del requisito	RU.09
Nombre del requisito	Utilizar escena
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz de accionar una escena que haya definido.

Tabla 21. Requisito de usuario 09

Número del requisito	RU.10
Nombre del requisito	Visualizar dispositivos
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz de visualizar el estado de los dispositivos que haya vinculado en dicha localización y acceder a cada uno de ellos desde la pantalla principal.

Tabla 22. Requisito de usuario 10

Número del requisito	RU.11
Nombre del requisito	Visualizar escenas
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz de visualizar el estado de las escenas que haya definido en dicha localización y acceder a cada una de ellas desde la pantalla principal.

Tabla 23. Requisito de usuario 11

Número del requisito	RU.12
Nombre del requisito	Acceso rápido dispositivos
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz filtrar rápidamente la lista de dispositivos que tiene vinculados a una localización para encontrarlo rápidamente.

Tabla 24. Requisito de usuario 12

Número del requisito	RU.13
Nombre del requisito	Acceso rápido escenas
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz filtrar rápidamente la lista de escenas que tiene vinculadas a una localización para encontrarlo rápidamente.

Tabla 25. Requisito de usuario 13

Número del requisito	RU.14
Nombre del requisito	Crear nueva localización
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz de crear una nueva localización en la que incluir sus dispositivos.

Tabla 26. Requisito de usuario 14

Número del requisito	RU.15
Nombre del requisito	Cambiar de localización
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz de cambiar de una localización a otra definida por el mismo usuario.

Tabla 27. Requisito de usuario 15

Número del requisito	RU.16
Nombre del requisito	Gestionar localización
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe ser capaz de eliminar una localización previamente definida por el usuario, con toda su información, así como modificar su nombre.

Tabla 28. Requisito de usuario 16

Número del requisito	RU.17
Nombre del requisito	Conocer localización
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El usuario debe tener constancia de en qué localización se encuentra.

Tabla 29. Requisito de usuario 17

4.1.2.2. Requisitos de restricción

Número del requisito	RUR.01
Nombre del requisito	Versión Android
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	El dispositivo del usuario en el que se ejecute la aplicación debe tener mínimo Android 4.4 KitKat (versión 19 SDK).

Tabla 30. Requisito de restricción 01

Número del requisito	RUR.02
Nombre del requisito	Permisos necesarios
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación debe contar con los siguientes permisos del dispositivo del usuario: <ul style="list-style-type: none"> • Acceso a Internet.

Tabla 31. Requisito de restricción 02

Número del requisito	RUR.03
Nombre del requisito	Acceso a Internet
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación requiere una conexión a Internet para utilizar todas sus funciones. Para ello se puede utilizar cualquier red, ya sea WiFi, Ethernet o conectividad móvil.

Tabla 32. Requisito de restricción 03

Número del requisito	RUR.04
Nombre del requisito	Idioma
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación cuenta con todo el contenido mostrado en inglés, salvo que el usuario tenga su dispositivo configurado en idioma español, que se mostrará el contenido en español.

Tabla 33. Requisito de restricción 04

Número del requisito	RUR.05
Nombre del requisito	Relación de pantalla
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación debe adaptarse a cualquier dispositivo Android, tenga la relación de aspecto que tenga su pantalla (16:9, 4:3, ...).

Tabla 34. Requisito de restricción 05

Número del requisito	RUR.06
Nombre del requisito	Unidades sensores
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación debe mostrar los valores numéricos de los sensores en unidades del Sistema Internacional (SI).

Tabla 35. Requisito de restricción 06

4.1.3. Requisitos software

4.1.3.1. Requisitos funcionales

4.1.3.1.1. Objetos a utilizar

Número del requisito	RSF.01
Nombre del requisito	Tipos de datos
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	<p>La aplicación deberá almacenar en la base de datos los siguientes tipos de datos:</p> <ul style="list-style-type: none"> • Usuarios. • Localizaciones. • Dispositivos. • Acciones.

Tabla 36. Requisito funcional 01

Número del requisito	RSF.02
Nombre del requisito	Objeto usuario
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	<p>La aplicación deberá almacenar en la base de datos, por cada usuario, los siguientes parámetros:</p> <ul style="list-style-type: none"> • Un identificador unívoco para la lista de usuarios. • El nombre del usuario. • El email del usuario. • Una lista de localizaciones.

Tabla 37. Requisito funcional 02

Número del requisito	RSF.03
Nombre del requisito	Objeto localización
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	<p>La aplicación deberá almacenar en la base de datos, dentro del correspondiente usuario, por cada localización, los siguientes parámetros:</p> <ul style="list-style-type: none"> • El nombre (identificador) de la localización. • Una lista de dispositivos. • Una lista de escenas.

Tabla 38. Requisito funcional 03

Número del requisito	RSF.04
Nombre del requisito	Objeto dispositivo
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	<p>La aplicación deberá almacenar en la base de datos, dentro de la localización seleccionada, por cada dispositivo, los siguientes parámetros:</p> <ul style="list-style-type: none"> • Un identificador para la lista de dispositivos. • Una descripción respecto a su funcionamiento. • Sus credenciales para Thinger. • La habitación en la que se encuentra. • Una lista de acciones. • La acción por defecto del mismo.

Tabla 39. Requisito funcional 04

Número del requisito	RSF.05
Nombre del requisito	Objeto acción
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	<p>La aplicación deberá almacenar en la base de datos, dentro del dispositivo seleccionado, por cada acción, los siguientes parámetros:</p> <ul style="list-style-type: none"> • un identificador para la lista de acciones. • El tipo de acción. • El valor actual de la acción.

Tabla 40. Requisito funcional 05

Número del requisito	RSF.06
Nombre del requisito	Objeto escena
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	<p>La aplicación deberá almacenar en la base de datos, en su correspondiente localización, por cada escena, los siguientes parámetros:</p> <ul style="list-style-type: none"> • Un identificador unívoco. • Una descripción respecto a su funcionamiento. • Una lista de dispositivos personalizada, que contengan las acciones seleccionadas con su estado concreto.

Tabla 41. Requisito funcional 06

4.1.3.1.2. *Thinger*

Número del requisito	RSF.07
Nombre del requisito	Peticiones API REST
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación deberá ser capaz de comunicarse con el sistema Thinger mediante peticiones API REST, para la gestión de los dispositivos y acciones.

Tabla 42. Requisito funcional 07

4.1.3.1.3. *Firebase*

Número del requisito	RSF.08
Nombre del requisito	Gestión de base de datos
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación deberá ser capaz de comunicarse con el sistema Firebase, para gestionar la base de datos, así como dar de alta a los usuarios.

Tabla 43. Requisito funcional 08

4.1.3.1.4. *Acceso a la aplicación*

Número del requisito	RSF.09
Nombre del requisito	Iniciar sesión
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación deberá permitir al usuario iniciar sesión / registrarse en la misma utilizando su cuenta de Google. Con esto registrará en la base de datos dicho usuario, como se especifica en RSF.01.

Tabla 44. Requisito funcional 09

Número del requisito	RSF.10
Nombre del requisito	Localización inicial
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación generará una localización inicial para el usuario, denominada “Localizacion1”. El usuario la podrá modificar como se explica en RSF.23.

Tabla 45. Requisito funcional 10

Número del requisito	RSF.11
Nombre del requisito	Cerrar sesión
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación deberá permitir al usuario, desde el menú desplegable, cerrar su sesión en la aplicación. Una vez cerrada deberá volver a la pantalla de inicio de sesión.

Tabla 46. Requisito funcional 11

Número del requisito	RSF.12
Nombre del requisito	Revocar acceso
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación deberá permitir al usuario, desde el menú desplegable, revocar acceso a la misma utilizando su cuenta de Google. Con esto se borrará de la base de datos su usuario, y con ello todos sus datos, especificados en RSF.01. Además, deberá realizar una segunda confirmación para evitar errores.

Tabla 47. Requisito funcional 12

4.1.3.1.5. *Menú principal*

Número del requisito	RSF.13
Nombre del requisito	Estructura pantalla principal
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación deberá contar con dos pestañas en la pantalla principal, en las cuales se mostrarán los dispositivos y las escenas del usuario.

Tabla 48. Requisito funcional 13

Número del requisito	RSF.14
Nombre del requisito	Estructura pantalla principal 2
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación deberá contar con un botón de búsqueda accesible, desde el cual filtrar los objetos de dicha pestaña mediante su identificador. Además, debe mostrar la localización actual al usuario.

Tabla 49. Requisito funcional 14

Número del requisito	RSF.15
Nombre del requisito	Manejo de datos. Cargando
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación deberá mostrar siempre una ventana flotante cuando esté cargando la información de la base de datos, la cual se cerrará una vez cargada.

Tabla 50. Requisito funcional 15

Número del requisito	RSF.16
Nombre del requisito	Pestaña dispositivos
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación deberá mostrar los dispositivos que tiene registrados en dicha localización, ordenados por habitaciones.

Tabla 51. Requisito funcional 16

Número del requisito	RSF.17
Nombre del requisito	Pestaña escenas
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación deberá mostrar las escenas que tiene creadas el usuario en dicha localización.

Tabla 52. Requisito funcional 17

4.1.3.1.6. *Crear objetos*

Número del requisito	RSF.18
Nombre del requisito	Botón desplegable
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación deberá mostrar un botón destacado en la pantalla principal en el que poder añadir: <ul style="list-style-type: none"> • Un nuevo dispositivo. • Una nueva escena.

Tabla 53. Requisito funcional 18

Número del requisito	RSF.19
Nombre del requisito	Nuevo dispositivo
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	<p>La aplicación deberá permitir al usuario introducir los siguientes parámetros para un nuevo dispositivo:</p> <ul style="list-style-type: none"> • el identificador propio de la aplicación. • una descripción respecto a su funcionamiento. • La habitación en la que se encuentra. En el caso de no tener disponible la habitación deseada, se le permitirá crear una nueva habitación. • La acción más relevante del dispositivo, a partir de la lista de acciones se obtendrá del sistema de Thinger.

Tabla 54. Requisito funcional 19

Número del requisito	RSF.20
Nombre del requisito	Nuevo dispositivo 2
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	<p>La aplicación deberá registrar el nuevo dispositivo en el sistema Thinger, a través de una petición API REST, enviando los siguientes parámetros:</p> <ul style="list-style-type: none"> • el identificador propio de la aplicación. • una descripción respecto a su funcionamiento. • Sus credenciales.

Tabla 55. Requisito funcional 20

Número del requisito	RSF.21
Nombre del requisito	Nueva escena
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	<p>La aplicación deberá permitir al usuario introducir los siguientes parámetros para una nueva escena:</p> <ul style="list-style-type: none"> • el identificador propio de la aplicación. • una descripción respecto a su funcionamiento. • Las acciones a gestionar, así como su estado.

Tabla 56. Requisito funcional 21

4.1.3.1.7. *Vistas objetos*

Número del requisito	RSF.22
Nombre del requisito	Vista dispositivo
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	Al pulsar sobre un dispositivo en la pestaña de dispositivos, la aplicación deberá mostrar los datos de dicho dispositivo, así como el estado actual de las acciones por defecto.

Tabla 57. Requisito funcional 22

Número del requisito	RSF.23
Nombre del requisito	Vista dispositivo 2
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	En la vista de dispositivo, la aplicación deberá permitir modificar el estado de cada acción, observar el estado de los sensores, así como permitir modificar los datos variables del mismo, o borrar el dispositivo.

Tabla 58. Requisito funcional 23

Número del requisito	RSF.24
Nombre del requisito	Vista escena
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	Al pulsar sobre una escena en la pestaña de escenas, la aplicación deberá mostrar los dispositivos de dicha escena, así como su estado actual.

Tabla 59. Requisito funcional 24

Número del requisito	RSF.25
Nombre del requisito	Vista escena 2
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	En la vista de escena, la aplicación deberá permitir conmutar su estado mediante un botón, así como permitir modificar los datos variables de la misma, o borrar la escena por completo.

Tabla 60. Requisito funcional 25

4.1.3.1.8. *Localización*

Número del requisito	RSF.26
Nombre del requisito	Vista localizaciones
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación deberá tener una vista en la que el usuario gestione sus localizaciones, cambiando entre ellas, modificando o borrando las mismas de la base de datos.

Tabla 61. Requisito funcional 26

Número del requisito	RSF.27
Nombre del requisito	Nueva localización
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación deberá permitir al usuario crear una localización nueva en caso que lo requiera.

Tabla 62. Requisito funcional 27

4.1.3.1. *Requisitos no funcionales*

Número del requisito	RSNF.01
Nombre del requisito	Manejo de datos
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación deberá utilizar siempre los valores escritos en la base de datos, y en caso de modificación, ser esta almacenada en la misma inmediatamente.

Tabla 63. Requisito no funcional 01

Número del requisito	RSNF.02
Nombre del requisito	Método de control de interfaces
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	Las interfaces que utilizan los usuarios se controlarán mediante una pantalla táctil.

Tabla 64. Requisito no funcional 02

Número del requisito	RSNF.03
Nombre del requisito	Tipo interfaz
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	Las interfaces deberán ser lo más minimalistas posible en las primeras capas de la aplicación.

Tabla 65. Requisito no funcional 03

Número del requisito	RSNF.04
Nombre del requisito	Estilo aplicación
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	Las interfaces de la aplicación deberán tener unos colores contrastados y ser clara para el usuario.

Tabla 66. Requisito no funcional 04

Número del requisito	RSNF.05
Nombre del requisito	Conectividad necesaria
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	<p>Para que el usuario pueda utilizar la aplicación y pueda recibir la información del sistema su terminal (dispositivo móvil con Android), necesitará conexión a Internet. Para este fin, necesitará algún hardware como:</p> <ul style="list-style-type: none"> • Tarjeta de red. • Hardware que permita la conectividad WiFi. • Hardware que permita la conectividad móvil (3G, 4G).

Tabla 67. Requisito no funcional 05

Número del requisito	RSNF.06
Nombre del requisito	Versión Android
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación requiere que el dispositivo del usuario en el que se ejecute la aplicación debe tener mínimo Android 4.4 KitKat (versión 19 SDK).

Tabla 68. Requisito no funcional 06

Número del requisito	RSNF.07
Nombre del requisito	Permisos necesarios
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación debe contar con los siguientes permisos del dispositivo del usuario: <ul style="list-style-type: none"> • Acceso a Internet.

Tabla 69. Requisito no funcional 07

Número del requisito	RSNF.08
Nombre del requisito	Idioma
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación cuenta con todo el contenido mostrado en inglés, salvo que el usuario tenga su dispositivo configurado en idioma español, que se mostrará el contenido en español. Este se podrá configurar en los ajustes.

Tabla 70. Requisito no funcional 08

Número del requisito	RSNF.09
Nombre del requisito	Relación de pantalla
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación debe adaptarse a cualquier dispositivo Android, tenga la relación de aspecto que tenga su pantalla (16:9, 4:3, ...).

Tabla 71. Requisito no funcional 09

Número del requisito	RSNF.10
Nombre del requisito	Unidades sensores
Tipo	Requisito
Fuente del requisito	Analista
Prioridad del requisito	Esencial
Descripción del requisito	La aplicación debe mostrar los valores numéricos de los sensores en unidades del Sistema Internacional, y permitir al usuario cambiar a otras unidades.

Tabla 72. Requisito no funcional 10

4.2. Prototipado

Una vez tuvo el analista la información obtenida durante la primera reunión por el jefe de proyecto con el cliente, llegó a la conclusión de realizar otra técnica de elicitación, realizar wireframes.

Un wireframe es un pequeño prototipo mediante el cual poder explicar las ideas de un proyecto, la ubicación de los botones, de las imágenes, etc. No concreta los colores de la aplicación, todo tiene que ser en colores planos, excepto botones importantes, que puedan tener un color llamativo. Es una solución rápida y barata para explicar una interfaz.

El analista desarrollará unos prototipos, con la herramienta Axure, de cada una de las vistas que debería tener la aplicación. Estos prototipos se crean con el fin de someter a diversos usuarios a distintas sesiones de observación en un entorno controlado, donde los usuarios interactúan con los prototipos. Gracias a estas sesiones, los analistas son capaces de captar la naturaleza de las tareas a realizar por los usuarios, y ajustar la lista de requisitos de usuario.

4.2.1. Explicación de wireframes

A continuación, se dejan los prototipos explicando las ideas del analista sobre cada uno de ellos, al igual que se hizo a los usuarios en las sesiones de observación:

4.2.1.1. Inicio de sesión

En este primer prototipo se muestra una simple pantalla inicial, donde se observa un único botón centrado, en el cual permitimos al usuario utilizar su cuenta de Google para acceder a la aplicación.

De tener varias cuentas disponibles para iniciar sesión, se abrirá un panel permitiendo al usuario elegir la cuenta deseada.

Una vez el usuario ha entendido la funcionalidad de esta interfaz, pasamos a la siguiente.

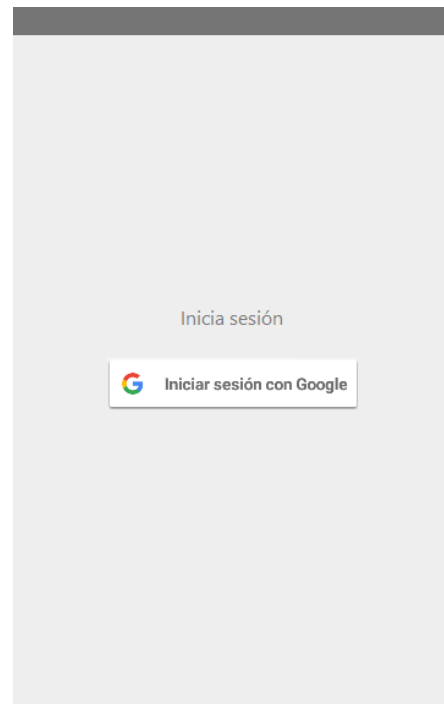


Ilustración 19. Wireframe inicio de sesión

4.2.1.2. Menú principal

Una vez hemos iniciado sesión, observamos la pantalla principal de la aplicación. Esta contiene tres pestañas:

- Dispositivos: donde se mostrarán los dispositivos añadidos por el usuario.
- Escenas: donde se mostrarán las escenas que cree el usuario.
- Programa: donde se mostrarán los programas creados por el usuario.

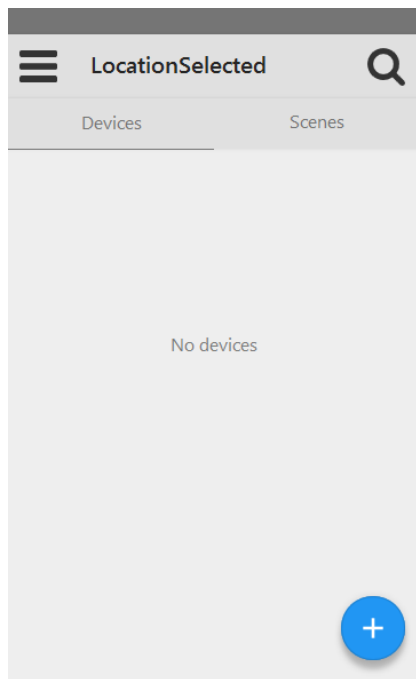


Ilustración 21. Wireframe menú principal 2

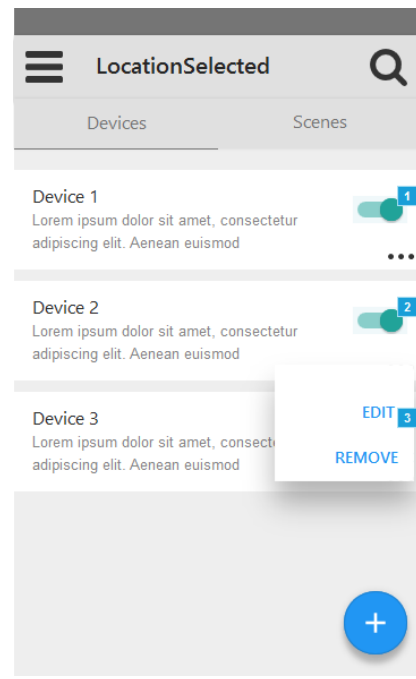


Ilustración 20. Wireframe menú principal 1

Como podemos observar, los dispositivos se muestran en formato tarjeta, con el nombre que los identifica, y su descripción. En el lado derecho, se añade un interruptor, el cual indica el estado del dispositivo y permite cambiarlo directamente. Además, la tarjeta cuenta con un menú oculto, que al desplegarse desvela unos accesos rápidos, permitiendo editar algunos parámetros del dispositivo, o eliminarlo, con un mensaje de doble confirmación, para evitar imprevistos.

Indicar, por último, que, si se pulsa en el dispositivo, se lleva a la pantalla de vista de dispositivo, que se explicará posteriormente.

4.2.1.3. Botón flotante

En la esquina de abajo derecha, se encuentra un llamativo botón azul, con un más en su interior. Este botón sirve para añadir los distintos elementos que se muestran en la pantalla principal.

Una vez se pulsa el botón flotante, se despliegan sobre el tres botones más, uno para añadir cada elemento, como se explica en la etiqueta que tiene cada botón a su lado: dispositivo o escena.

Cada uno de ellos lleva a una nueva pantalla, que se explicará posteriormente.

El botón original gira el más de su interior, pasando a ser una equis, dando a entender al usuario que, si se pulsa, se vuelve a la vista anterior.

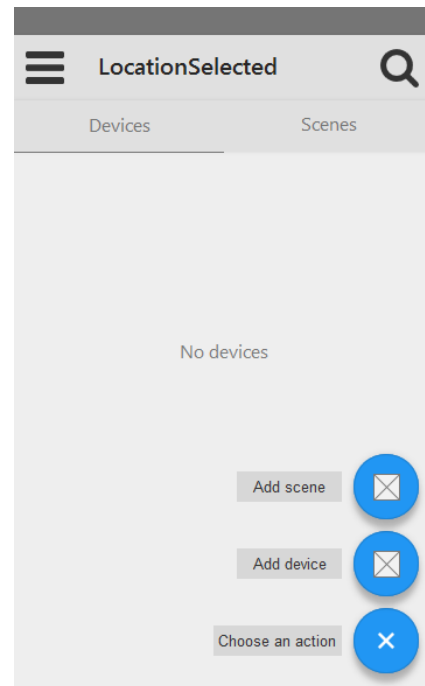


Ilustración 22. Wireframe botón flotante

4.2.1.4. Nuevo dispositivo

Cuando se pulsa el botón de añadir dispositivo, se lleva a esta pantalla. En ella, se pide al usuario que introduzca los parámetros que definamos para un nuevo dispositivo. Por ahora, que introduzca un identificador único, una descripción para el mismo, y las credenciales del mismo. Finalmente, requerimos que se escoja la habitación en la que está colocado el dispositivo, permitiendo crear una nueva en el caso de no existir en la lista desplegable.

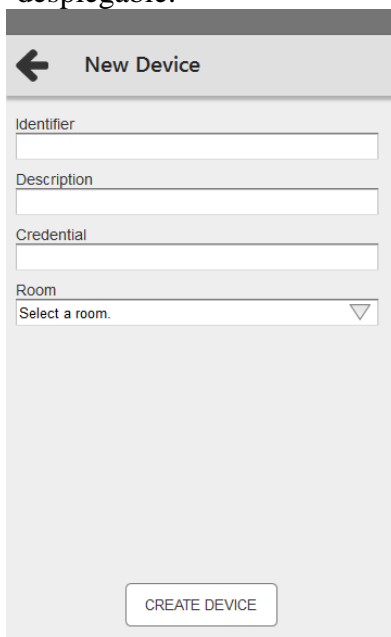


Ilustración 24. Wireframe nuevo dispositivo 2

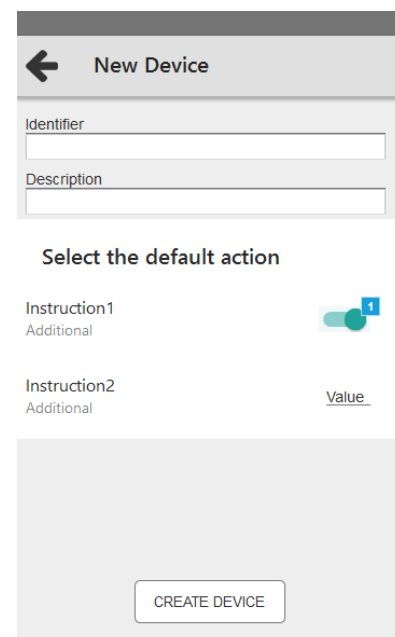
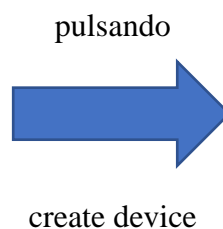


Ilustración 23. Wireframe nuevo dispositivo 1

Finalmente, al crear el dispositivo se pregunta al usuario qué acción de las disponibles en dicho dispositivo quiere como acción por defecto, a mostrar en el menú principal. Una vez se elige se vuelve al menú principal, donde se podrá observar el dispositivo en la pestaña de dispositivos.

4.2.1.5. Vista dispositivo

Al pulsar un dispositivo, pasamos a esta pantalla. En ella, podemos observar el contenido del dispositivo. Primero, encontraremos las instrucciones del mismo, permitiendo al usuario modificar el valor de manera sencilla. Debajo, contamos con una lista con los parámetros que reciben los sensores incorporados por el dispositivo.

Además, contamos con dos botones en la esquina superior derecha, para eliminar el dispositivo, con una doble confirmación, o editar los campos permitidos, aún por definir.

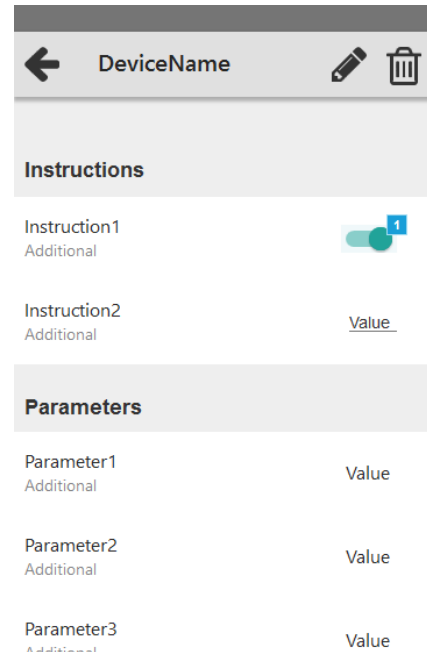


Ilustración 25. Wireframe vista dispositivo

4.2.1.6. Nueva escena

Al pulsar en el botón de crear nueva escena, se lleva a una vista nueva.

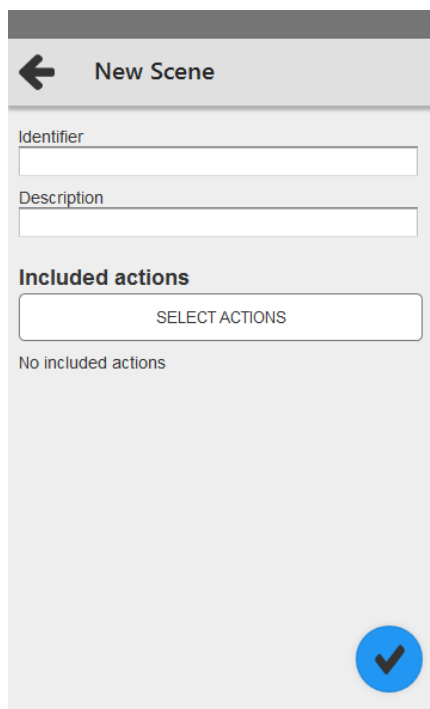


Ilustración 27. Wireframe nueva escena 2

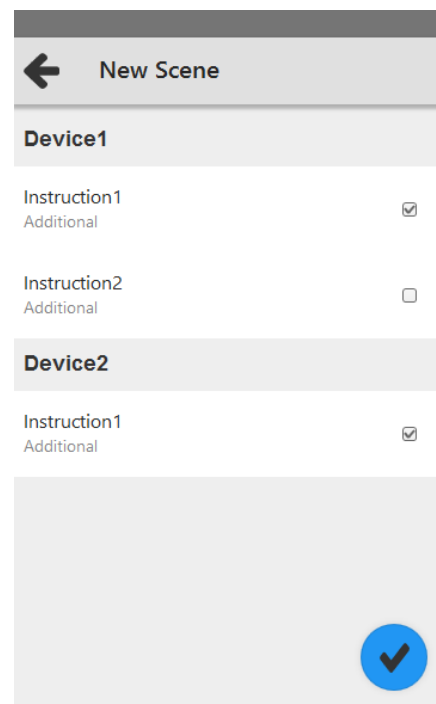
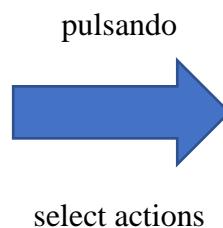


Ilustración 26. Wireframe nueva escena 1

En esta vista, se rellena un breve formulario relativo a la escena, y existe un botón para incluir las acciones que activará la escena. Tras pulsarlo vamos a la segunda vista, donde el usuario seleccionará las acciones que quiere modificar con dicha escena.

Una vez seleccionadas, tras pulsar el botón flotante, volverá a la vista anterior, pero aparecerán las acciones anteriores, permitiendo al usuario ajustar el estado deseado para cada acción seleccionada.

Una vez terminados los ajustes, se pulsa el botón flotante y se guarda la escena creada en la base de datos.

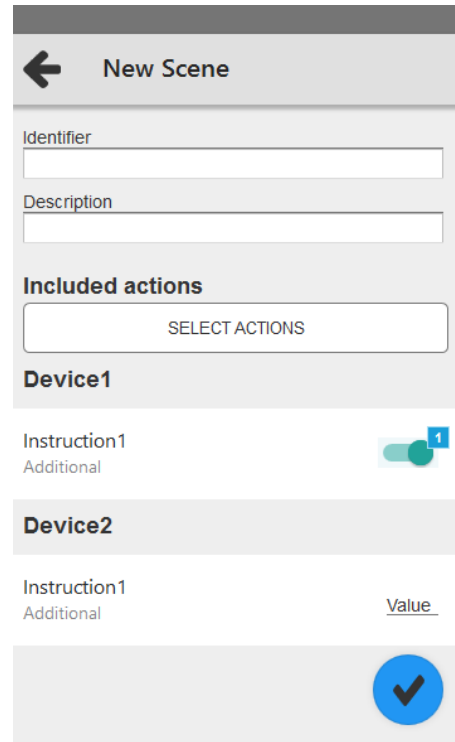


Ilustración 28. Wireframe nueva escena 3

4.2.1.7. Vista escena

Al seleccionar una escena, se pasa a la presente vista. En esta, podemos observar si la escena está activada (o no), y el estado actual de las acciones que componen la escena.

Además, contamos con dos botones en la esquina superior derecha, para eliminar la escena, con una doble confirmación, o editar los campos permitidos, aún por definir.

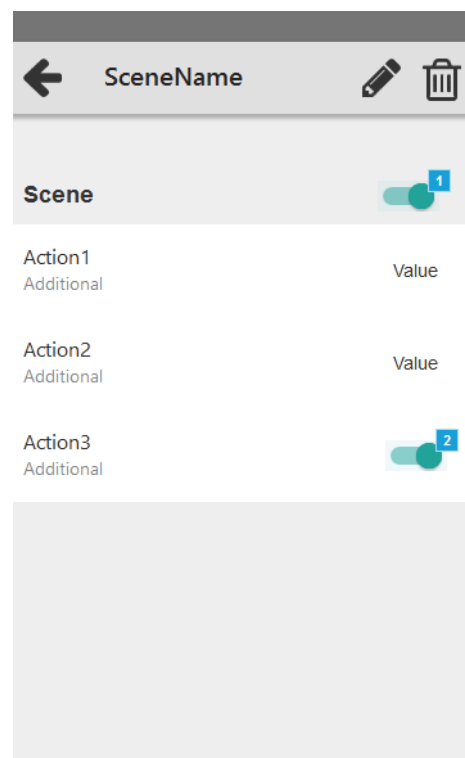
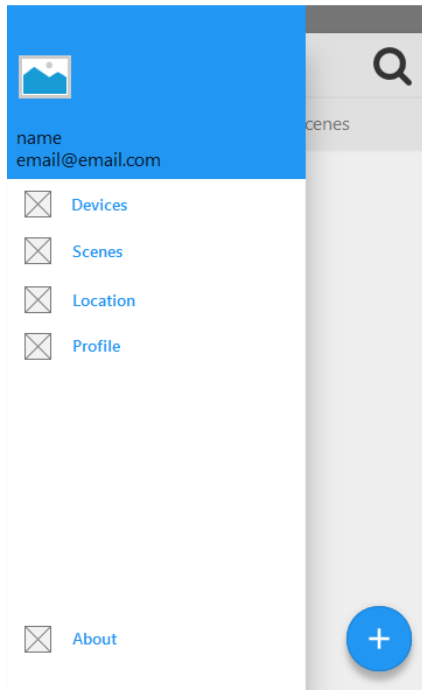


Ilustración 29. Wireframe vista escena

4.2.1.8. Menú de navegación

En la parte superior observamos, a la derecha, un botón de búsqueda, el cual permitirá filtrar rápidamente los objetos mostrados en cada pestaña. A la izquierda, nos encontramos con un menú desplegable, el cual contiene la siguiente funcionalidad:



- Devices, scenes: se desplaza a la pestaña correspondiente del menú principal.
- Location: lleva a otra pantalla, que se explicará a continuación.
- Profile: el cuadro azul, que muestra el nombre del usuario, junto a su correo electrónico y su imagen de perfil, es un acceso al perfil, el cual nos lleva a otra pantalla donde poder cerrar sesión, o revocar acceso a la aplicación.
- About: muestra información relativa a la aplicación.

Ilustración 30. Wireframe menú de navegación

4.2.1.9. Seleccionar localización

Al seleccionar Location en el menú de navegación, se abre la siguiente pantalla. En ella, el usuario puede seleccionar la localización que quiere gestionar.

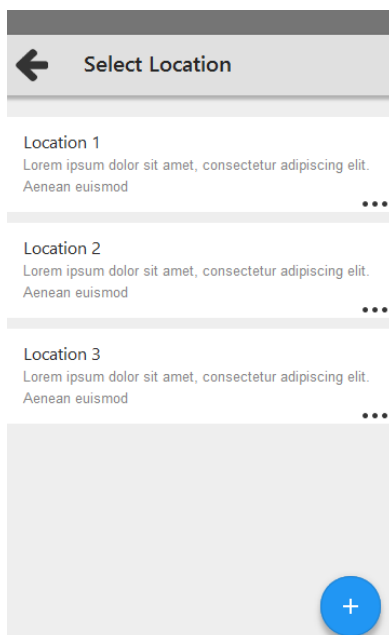



Ilustración 31. Wireframe seleccionar localización 1

pulsando

 botón flotante

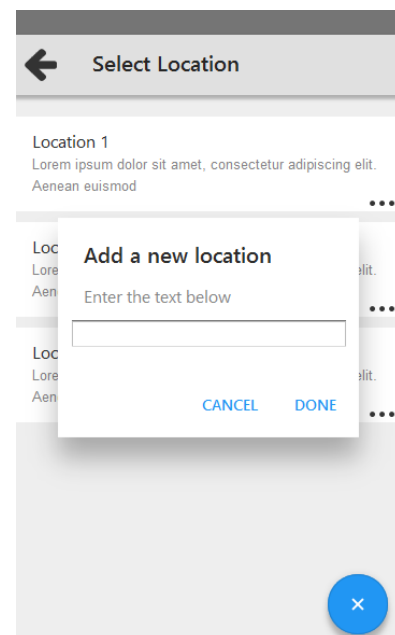


Ilustración 32. Wireframe seleccionar localización 2

Además, tiene la posibilidad de crear nuevas localizaciones, pulsando el botón flotante. Este desplegará un cuadro en el cual incluir los parámetros de la nueva localización

Por último, al seleccionar una localización cargará el menú principal con los datos de dicha localización.

4.2.2. Conclusiones sacadas

Una vez terminadas las sesiones de observación, el analista pasa a explicar los resultados más relevantes, con su opinión a continuación.

- La mayoría de usuarios no intuyó el perfil en la zona azul. Sería interesante hacer un botón más claro.
- Algunos usuarios pidieron un botón de ayuda disponible en el menú principal, para escuchar la explicación proporcionada por el analista de nuevo. La idea se descarta puesto que se cuenta con el manual del usuario.
- Los usuarios entendieron de manera correcta el funcionamiento de la vista de dispositivo.
- A la mayoría de los usuarios le pareció un gran diseño la interfaz principal de la aplicación.
- Algunos usuarios pensaron que sería buena idea compartir con otros usuarios utilizar los dispositivos, como la familia o compañeros de piso. Esta idea no se ve factible debido al escaso tiempo de desarrollo de la aplicación.
- Algunos usuarios pensaron que, si el dispositivo no se encontrase disponible, o perdiera la conexión, se debería oscurecer su tarjeta. Esta es una gran idea, que se intentará implementar en la aplicación.
- Los usuarios entendieron perfectamente el funcionamiento de las escenas.

4.3. Casos de uso

A continuación, vamos a definir los casos de uso de la aplicación. Un caso de uso define las actividades a realizar para realizar una determinada tarea, definiendo los actores que participan para su realización.

Para una mejor comprensión, debido a la cantidad de sistemas definidos (cada sistema es una vista de la aplicación), expondremos cada sistema por separado, seguido de sus casos de uso en formato expandido, indicando el curso típico de eventos, y sus posibles cursos alternativos.

Los casos de uso en formato expandido seguirán la siguiente estructura:

Identificador	UC_XX, siendo XX el número del caso de uso.	
Caso de uso	Nombre del caso de uso.	
Actores	Actores que intervienen en dicho caso de uso.	
Prioridad	Prioridad del caso de uso.	
Fuente	Analista o cliente.	
Pre-condiciones	Pre-condiciones requeridas por el sistema.	
Post-condiciones	Post-condiciones sufridas por el sistema.	
Descripción	Descripción del caso de uso.	
Curso típico de eventos		
	Actor	Sistema
	Acción realizada por el usuario	Acción realizada por el sistema
Están numeradas siguiendo el curso de eventos		
Curso alternativo de eventos		
Se especifica cada acción alternativa.		

Tabla 73. Estructura casos de uso

4.3.1. Actores involucrados

- Usuario: persona que utiliza la aplicación móvil.
- Firebase: API utilizada para gestión de la base de datos online.
- Thinger: API utilizada para la gestión de dispositivos.

Por simplicidad a la hora de observar el diagrama, y para no repetir casos de uso, los relacionados con las APIs se introducen aparte, en vez de extender del caso de uso realizado por el usuario. Destacar que serán referenciados en cada curso típico o alternativo de eventos.

4.3.2. Casos de uso

4.3.2.1. Inicio de sesión

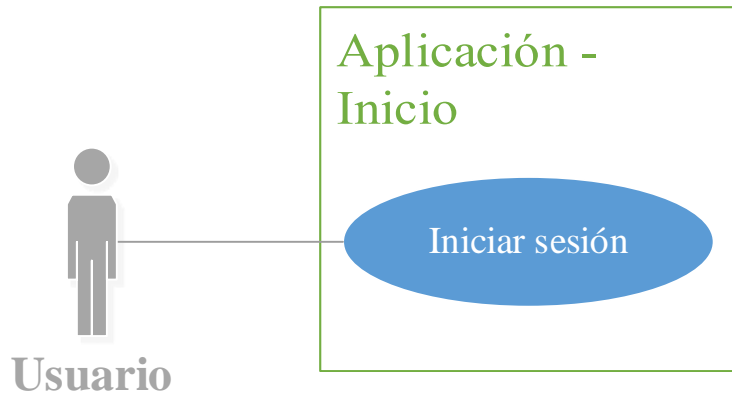


Ilustración 33. Casos de uso Inicio

Identificador	UC_01
Caso de uso	Inicio de sesión con cuenta de Google
Actores	Usuario, Firebase
Prioridad	Alta
Referencia	Analista
Pre-condiciones	Tener acceso a Internet
Post-condiciones	Se genera una entrada en la base de datos para esa cuenta de Google, y se accede al menú principal.
Descripción	Permite al usuario iniciar sesión en la aplicación utilizando su cuenta de Google.

Curso típico de eventos

Actor	Sistema
1. Iniciar la aplicación	
2. Pulsar sobre el botón “iniciar sesión con Google”	3. El sistema observa las cuentas de Google del usuario y se las muestra.
4. Escoger la cuenta deseada.	5. La API Firebase registra el usuario.
	6. Una vez registrado, se pasa al menú principal.

Curso alternativo de eventos

3.1. En el caso no tener ninguna cuenta de Google, el sistema permite al usuario crear uno nuevo.
5.1. En caso de error, el sistema mostrará un mensaje indicando el motivo.

Tabla 74. UC_01 Inicio de sesión con cuenta de Google

4.3.2.2. Menú principal

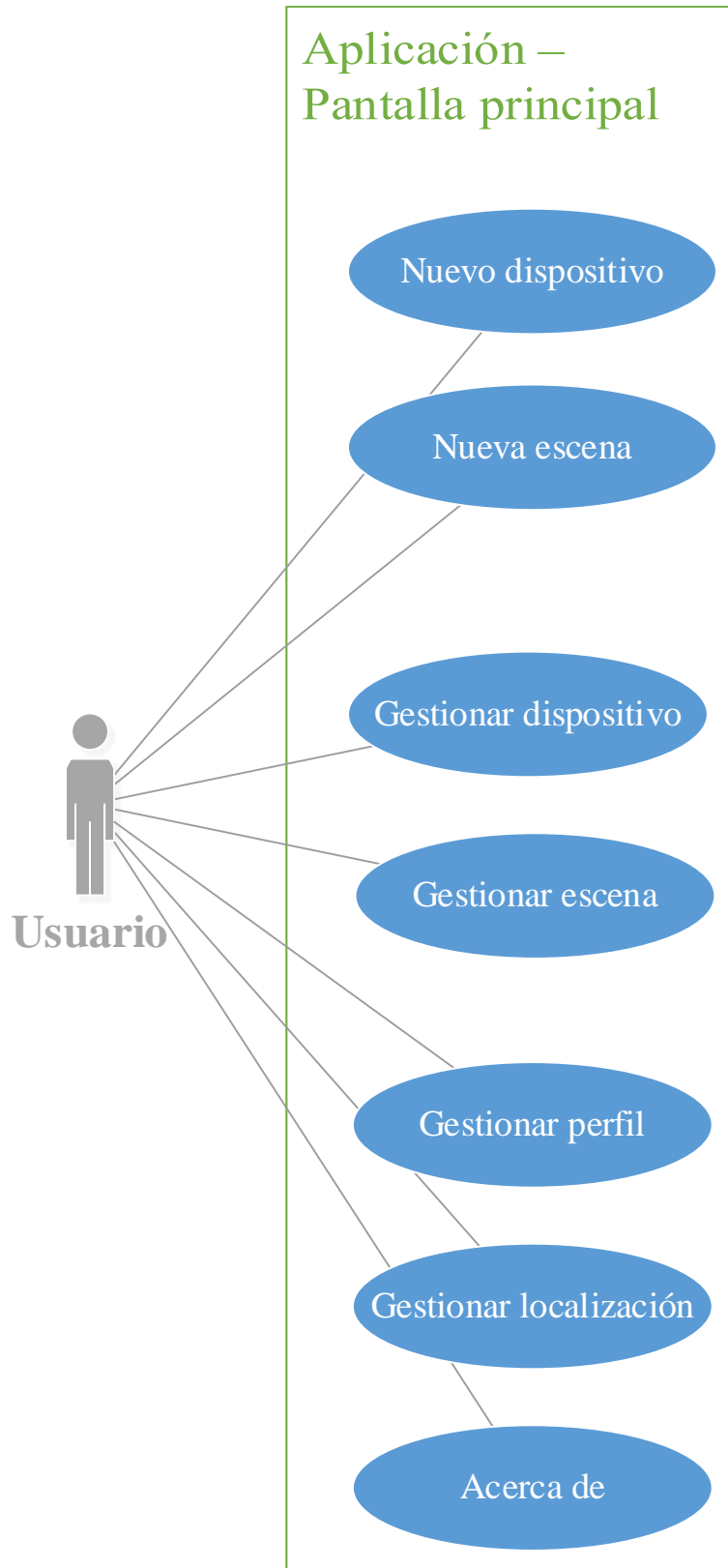


Ilustración 34. Casos de uso Pantalla principal

Identificador	UC_02
Caso de uso	Nuevo dispositivo
Actores	Usuario, Thinger, Firebase
Prioridad	Alta
Referencia	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación
Post-condiciones	Se genera una entrada en la base de datos de esa cuenta, almacenando el dispositivo creado.
Descripción	Permite al usuario crear un dispositivo en la base de datos, el cual se vincule con el dispositivo domótico físico.
Curso típico de eventos	
Actor	Sistema
1. Pulsar el botón flotante.	2. El sistema desplegará tres botones sobre el botón flotante.
3. Pulsar el botón “Nuevo dispositivo”.	4. El sistema pasará a la vista de creación de dispositivos.
5. Rellenar el formulario.	
6. Seleccionar una habitación.	
7. Pulsar el botón flotante para validar los datos.	8. El sistema valida los datos.
	9. La API de Thinger registra el dispositivo, y muestra al usuario las acciones disponibles para seleccionar por defecto.
10. El usuario escoge la acción por defecto.	11. La API de Firebase escribe en la base de datos el dispositivo generado
Curso alternativo de eventos	
6.1. Si el usuario no encuentra la habitación deseada, se le permite crear una nueva, mediante una ventana emergente que crea el sistema.	
8.1. Si los datos introducidos son erróneos, el sistema informará al usuario del error realizado, para que solvante el mismo.	
9.1. Si se produce un error al registrar el dispositivo, se detendrá el registro y se notificará al usuario del error.	
9.2. Si el dispositivo cuenta con una única acción, dicha acción será la acción por defecto y se realizará el paso 11.	

Tabla 75. UC_02 Nuevo dispositivo

Identificador	UC_03	
Caso de uso	Nueva escena	
Actores	Usuario, Firebase	
Prioridad	Alta	
Fuente	Analista	
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación	
Post-condiciones	Se genera una entrada en la base de datos de esa cuenta, almacenando la escena creada.	
Descripción	Permite al usuario crear una escena en la base de datos, la cual active unos determinados dispositivos domóticos.	
Curso típico de eventos		
	Actor	Sistema
	1. Pulsar el botón flotante.	2. El sistema desplegará tres botones sobre el botón flotante.
	3. Pulsar el botón “Nueva Escena”.	4. El sistema pasará a la vista de creación de escenas.
	5. Rellenar el formulario.	
	6. Pulsar el botón “Añadir acciones”	7. El sistema pasará a la vista de selección de acciones.
	8. Seleccionar las acciones deseadas para configurar la escena.	
	9. Pulsar el botón flotante añadir dichas acciones a la escena.	10. El sistema valida las acciones seleccionadas y vuelve a la anterior pantalla, añadiendo las acciones a la misma.
	11. Seleccionar el estado de cada una de las acciones.	
	12. Pulsar el botón flotante para añadir la escena.	13. El sistema valida los datos.
		14. La API de Firebase escribe en la base de datos la escena generada
Curso alternativo de eventos		
	6.1. Si el usuario no cuenta con dispositivos, se le indicará al usuario, cancelando la operación.	
	11.1. Si el usuario desea volver a añadir más dispositivos para configurar la escena, se le permite, manteniendo los dispositivos ya seleccionados.	
	13.1. Si los datos introducidos son erróneos, el sistema informará al usuario del error realizado, para que solviente el mismo.	

Tabla 76. UC_03 Nueva escena

Identificador	UC_04
Caso de uso	Gestionar dispositivo
Actores	Usuario
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación Tener dispositivos añadidos
Post-condiciones	Se accede a la vista del determinado dispositivo.
Descripción	Permite al usuario pasar a la vista de gestión de un dispositivo, donde controlar sus acciones, observar los valores de sus sensores, o modificar o borrar el dispositivo.
Curso típico de eventos	
Actor	Sistema
1. Colocarse en la pestaña “Dispositivos”	2. El sistema cargará en ella todos los dispositivos añadidos por el usuario
3. Seleccionar el dispositivo deseado, encontrándolo deslizando hacia abajo o utilizando la búsqueda	4. El sistema cambiará a la vista de dispositivo.
Curso alternativo de eventos	
3.1.El usuario puede pulsar el botón de tres puntos para desplegar un menú, en el cual seleccionar “Modificar”, para acceder a la vista de dispositivo en modificación, explicada en UC_15; o “Borrar” para eliminar el dispositivo, explicado en UC_16.	
3.2.Además, puede seleccionar el interruptor para activar / desactivar el dispositivo, explicado en UC_17.	

Tabla 77. UC_04 Gestionar dispositivo

Identificador	UC_05
Caso de uso	Gestionar escena
Actores	Usuario
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación Tener escenas añadidas
Post-condiciones	Se accede a la vista de la determinada escena.
Descripción	Permite al usuario gestionar una determinada escena, configurar sus dispositivos y su estado, añadir o eliminar acciones, o modificar o borrar la misma.

Curso típico de eventos	
Actor	Sistema
1. Colocarse en la pestaña “Escenas”	2. El sistema cargará en ella todas las escenas añadidas por el usuario
3. Seleccionar la escena deseada, encontrándola deslizando hacia abajo o utilizando la búsqueda	4. El sistema cambiará a la vista de escena.

Curso alternativo de eventos
3.1.El usuario puede pulsar el botón de tres puntos para desplegar un menú, en el cual seleccionar “Modificar”, para acceder a la vista de escena en modificación; o “Borrar” para eliminar la escena.
3.2.Además, puede seleccionar el interruptor para activar / desactivar la escena.

Tabla 78. UC_05 Gestionar escena

Identificador	UC_06
Caso de uso	Gestionar localización
Actores	Usuario
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación
Post-condiciones	Acceder a la vista de localización.
Descripción	Permite al usuario acceder a la pantalla en la que gestionar sus localizaciones.

Curso típico de eventos	
Actor	Sistema
1. Abrir el menú de navegación	
2. Seleccionar “Localización”	3. El sistema cambiará a la vista de localización

Curso alternativo de eventos

Tabla 79. UC_06 Gestionar localización

Identificador	UC_07
Caso de uso	Gestionar perfil
Actores	Usuario
Prioridad	Alta
Fuente	
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación
Post-condiciones	Acceder a la vista de gestión de perfil.
Descripción	Permite al usuario acceder a la pantalla en la que gestionar su perfil

Curso típico de eventos	
Actor	Sistema
1. Abrir el menú de navegación	
2. Seleccionar “Perfil”	3. El sistema cambiará a la vista de gestión de perfil
Curso alternativo de eventos	

Tabla 80. UC_07 Gestionar perfil

Identificador	UC_08
Caso de uso	Acerca de la aplicación
Actores	Usuario
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación
Post-condiciones	Abrir una ventana con la información de la aplicación
Descripción	Permite al usuario ver la información referente a la aplicación
Curso típico de eventos	
Actor	Sistema
1. Abrir el menú de navegación	
2. Seleccionar “Acerca de”	3. El sistema abrirá una ventana emergente mostrando la información al usuario
Curso alternativo de eventos	

Tabla 81. UC_08 Acerca de la aplicación

4.3.2.3. Pantalla perfil

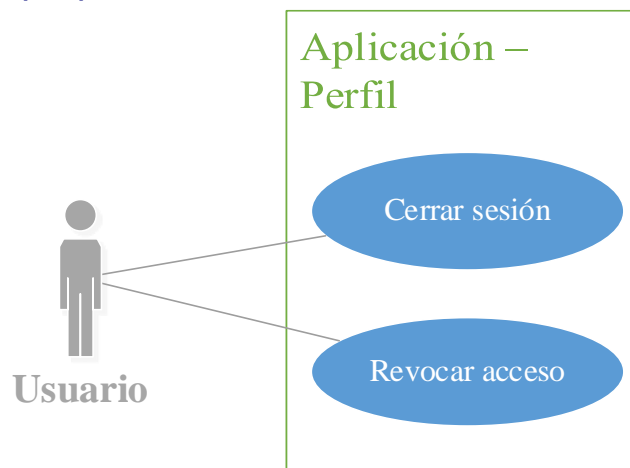


Ilustración 35. Casos de uso Perfil

Identificador	UC_09
Caso de uso	Cerrar sesión
Actores	Usuario
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación Haber accedido a la vista de perfil (UC_11)
Post-condiciones	Se vuelve a la pantalla de inicio
Descripción	Permite al usuario cerrar la sesión de su perfil
Curso típico de eventos	
Actor	Sistema
1. Pulsar el botón “Cerrar sesión”	2. El sistema cerrará la sesión del usuario.
	3. El sistema sale de la pantalla de perfil y vuelve a la pantalla de inicio de sesión.
Curso alternativo de eventos	
1.1.Si se produjese algún error cerrando la sesión, se saldría al menú principal igualmente, pero se notificaría al usuario.	

Tabla 82. UC_09 Cerrar sesión

Identificador	UC_10
Caso de uso	Revocar acceso
Actores	Usuario, Firebase
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación Haber accedido a la vista de perfil (UC_11)
Post-condiciones	Se vuelve a la pantalla de inicio y se eliminan los datos del usuario de la base de datos
Descripción	Permite al usuario eliminar toda su información almacenada en la base de datos de la aplicación
Curso típico de eventos	
Actor	Sistema
1. Pulsar el botón “Revocar acceso”	2. El sistema indicará al usuario que se borrará todo su contenido, y pide doble confirmación.
3. El usuario pulsa “Aceptar”	4. El sistema cerrará la sesión del usuario.

	5. La API de Firebase eliminará todas las entradas del usuario de la base de datos.
	6. El sistema sale de la pantalla de perfil y vuelve a la pantalla de inicio de sesión.
Curso alternativo de eventos	
3.1.	Si el usuario pulsa “Cancelar”, el sistema cancelará toda la operación.
1.1.	Si se produjese algún error cerrando la sesión, se saldría al menú principal igualmente, pero se notificaría al usuario.

Tabla 83. UC_10 Revocar acceso

4.3.2.4. Localización

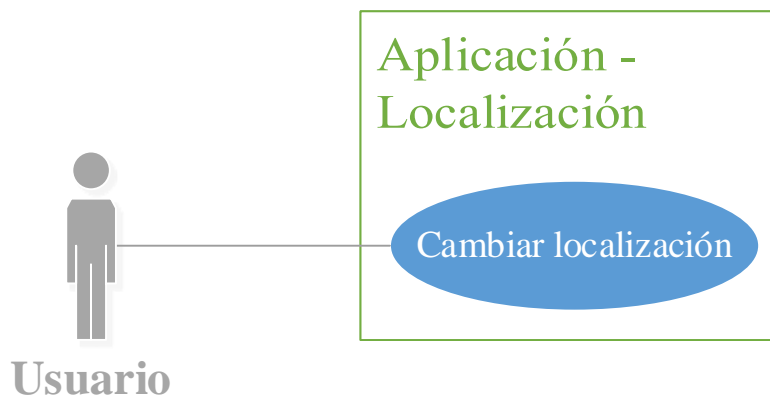


Ilustración 36. Casos de uso Localización

Identificador	UC_11
Caso de uso	Cambiar localización
Actores	Usuario, Firebase
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación Haber accedido a la vista de localización (UC_08)
Post-condiciones	Se vuelve a la pantalla principal con los datos de dicha localización
Descripción	Permite al usuario cambiar a la localización deseada, para gestionar sus dispositivos

Curso típico de eventos	
Actor	Sistema
	1. El sistema generará una lista con todas las localizaciones del usuario, cargadas mediante los datos recibidos de la API de Firebase.
2. Escoge la localización deseada.	3. El sistema vuelve a la pantalla principal, notificando utilizar la localización seleccionada.
Curso alternativo de eventos	
2.1. Si no encontrase la localización deseada, deberá crearla seleccionando el botón flotante, el cual creará una ventana para añadir el nombre de la nueva localización.	

Tabla 84. UC_11 Cambiar localización

4.3.2.5. Dispositivo

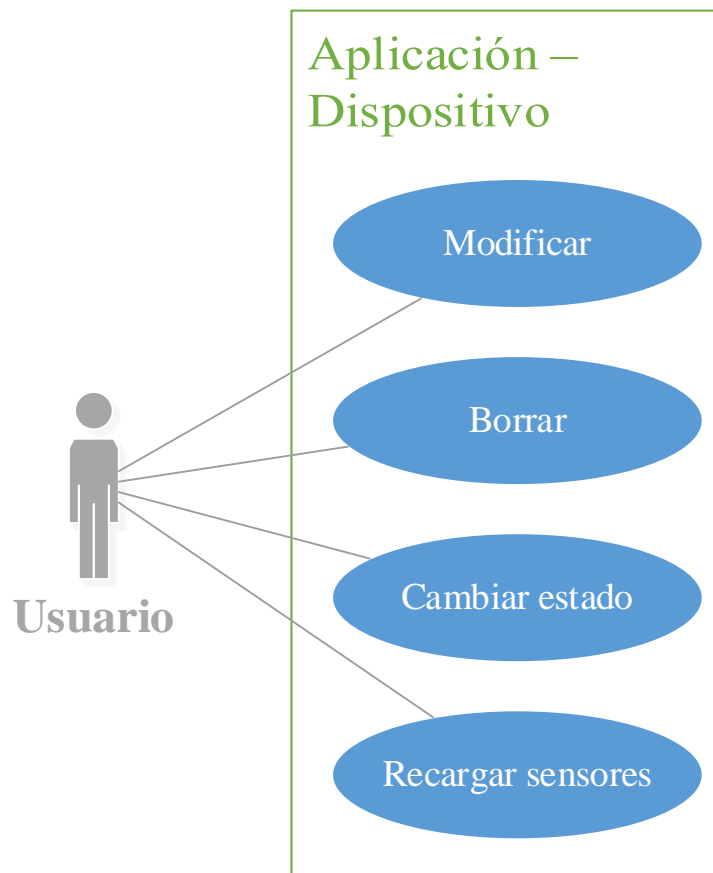


Ilustración 37. Casos de uso Dispositivo

Identificador	UC_12
Caso de uso	Modificar dispositivo
Actores	Usuario, Thinger, Firebase
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación Haber accedido a la vista de dispositivo, o haber accedido desde el caso de uso UC_08 3.1
Post-condiciones	Se editan los parámetros modificados del dispositivo en la base de datos
Descripción	Permite al usuario cambiar algunos parámetros del dispositivo, mediante un formulario
Curso típico de eventos	
Actor	Sistema
1. Pulsa el botón de modificar	2. El sistema accede a la pantalla de modificar dispositivo, cargando el formulario con los valores actuales del dispositivo, gracias a la API de Thinger.
3. Modifica el formulario	
4. Pulsa el botón flotante para guardar la modificación	5. El sistema valida los datos introducidos en el formulario
	6. La API de Firebase escribe los datos introducidos por el usuario en la base de datos, eliminando previamente los ya existentes.
	7. El sistema vuelve a la vista anterior
Curso alternativo de eventos	
3.2.Si el usuario no encuentra la habitación deseada, se le permite crear una nueva, mediante una ventana emergente que crea el sistema.	
5.1.Si los datos introducidos son erróneos, el sistema informará al usuario del error realizado, para que solvante el mismo.	

Tabla 85. UC_12 Modificar dispositivo

Identificador	UC_13
Caso de uso	Borrar dispositivo
Actores	Usuario, Firebase, Thinger
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación Haber accedido a la vista de dispositivo, o haber accedido desde el caso de uso UC_08 3.1
Post-condiciones	Se elimina todo rastro de dicho dispositivo de la base de datos. Se vuelve al menú principal
Descripción	Permite al usuario eliminar un dispositivo incluido previamente desde la aplicación
Curso típico de eventos	
Actor	Sistema
1. Pulsa el botón de borrar	2. El sistema crea una ventana emergente que solicita la confirmación del usuario
3. Pulsa el botón “Eliminar”	4. La API de Firebase elimina el dispositivo de la base de datos
	5. La API de Thinger elimina el dispositivo
	6. El sistema vuelve a la vista anterior
Curso alternativo de eventos	
3.1.Si el usuario decide no eliminar el dispositivo, pulsa el botón “Cancelar”, y se cerraría la ventana emergente, cancelando la operación	
3.1.Si se produce algún error eliminando el dispositivo, se informaría al usuario, y se cancelaría la operación	
4.1.Si dicho dispositivo estuviera en alguna escena, se eliminará también de la misma.	

Tabla 86. UC_13 Borrar dispositivo

Identificador	UC_14
Caso de uso	Cambiar estado acción
Actores	Usuario, Thinger
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación Haber accedido a la vista de dispositivo, o haber accedido desde el caso de uso UC_08 3.2
Post-condiciones	Se realiza la correspondiente acción del dispositivo

Descripción	Permite al usuario realizar modificar el estado de una acción del dispositivo. Existen tres cursos, en función del tipo de dispositivo	
Curso tipo booleano		
Actor	Sistema	
1. Pulsa el interruptor de la acción	2. El sistema conmuta el estado del interruptor	
	3. La API de Thinger realiza una petición PUT para modificar el estado de dicha acción.	
Curso tipo numérico		
Actor	Sistema	
1. Pulsa sobre el cuadro de texto de la acción.	2. El sistema permitirá al usuario escribir números.	
3. El usuario introduce el valor numérico.	4. El sistema valida el valor numérico.	
	5. La API de Thinger realiza una petición PUT para modificar el estado de dicha acción.	
Curso tipo texto		
Actor	Sistema	
1. Pulsa sobre el cuadro de texto de la acción.	2. El sistema permitirá al usuario escribir.	
3. El usuario introduce la cadena deseada.	4. El sistema valida el texto introducido.	
	5. La API de Thinger realiza una petición PUT para modificar el estado de dicha acción.	

Tabla 87. UC_14 Cambiar estado acción

Identificador	UC_15
Caso de uso	Recargar sensores
Actores	Usuario, Thinger
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación Haber accedido a la vista de dispositivo, o haber accedido desde el caso de uso UC_08 3.1
Post-condiciones	Se actualizan los valores de las acciones del dispositivo.

Descripción	Permite al usuario refrescar los valores obtenidos por los sensores del dispositivo
Curso típico de eventos	
Actor	Sistema
1. Pulsa el botón de recargar sensores.	2. Se envía una petición mediante la API de Thinger para cada acción de tipo sensor, y modifica los valores de la acción.
Curso alternativo de eventos	

Tabla 88. UC_15 Recargar sensores

4.3.2.6. Escena

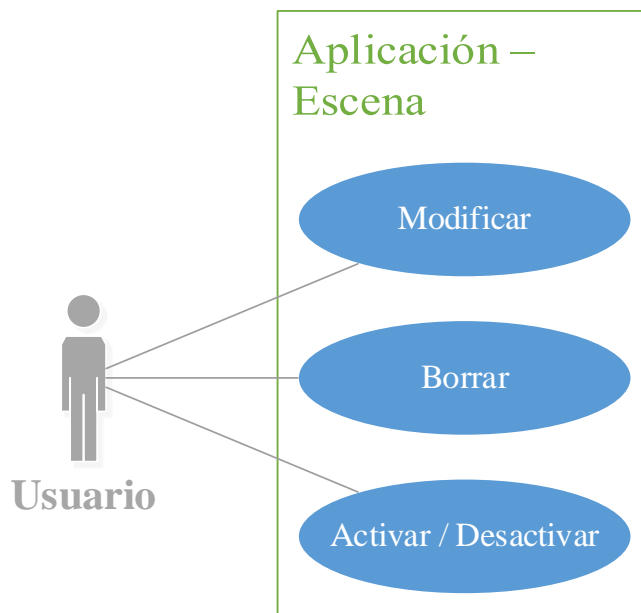


Ilustración 38. Casos de uso Escena

Identificador	UC_16
Caso de uso	Modificar escena
Actores	Usuario, Thinger, Firebase
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación Haber accedido a la vista de escena, o haber accedido desde el caso de uso UC_08 3.1

Post-condiciones	Se editan los parámetros modificados de la escena en la base de datos	
Descripción	Permite al usuario cambiar algunos parámetros d la escena, mediante un formulario	
Curso típico de eventos		
Actor	Sistema	
1. Pulsa el botón de modificar	2. El sistema accede a la pantalla de modificar dispositivo, cargando el formulario con los valores actuales del dispositivo, gracias a Thinger.	
3. Modifica el formulario		
4. El usuario puede añadir o eliminar acciones, de igual forma a cuando crea una escena (UC_03).		
5. Pulsa el botón flotante para guardar la modificación	6. El sistema valida los datos introducidos en el formulario	
	7. La API de Firebase escribe los datos introducidos por el usuario en la base de datos, eliminando previamente los ya existentes.	
	8. El sistema vuelve a la vista anterior	
Curso alternativo de eventos		
6.1.Si los datos introducidos son erróneos, el sistema informará al usuario del error realizado, para que solvente el mismo.		

Tabla 89. UC_16 Modificar escena

Identificador	UC_17
Caso de uso	Borrar escena
Actores	Usuario, Firebase
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación Haber accedido a la vista de escena, o haber accedido desde el caso de uso UC_08 3.1
Post-condiciones	Se elimina todo rastro de dicho dispositivo de la base de datos. Se vuelve al menú principal
Descripción	Permite al usuario eliminar un dispositivo incluido previamente desde la aplicación

Curso típico de eventos	
Actor	Sistema
1. Pulsa el botón de borrar	2. El sistema crea una ventana emergente que solicita la confirmación del usuario
3. Pulsa el botón “Eliminar”	4. La API de Firebase elimina el dispositivo de la base de datos
	5. El sistema vuelve a la vista anterior
Curso alternativo de eventos	
3.2.Si el usuario decide no eliminar el dispositivo, pulsa el botón “Cancelar”, y se cerraría la ventana emergente, cancelando la operación	
3.2.Si se produce algún error eliminando el dispositivo, se informaría al usuario, y se cancelaría la operación	
4.2.Si dicho dispositivo estuviera en alguna escena, se eliminará en ella.	

Tabla 90. UC_17 Borrar escena

Identificador	UC_18
Caso de uso	Activar / desactivar escena
Actores	Usuario, Thingier
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación Haber accedido a la vista de escena, o haber accedido desde el caso de uso UC_08 3.1
Post-condiciones	Se modifican las acciones que componen la escena al estado que definió el usuario.
Descripción	Permite al usuario eliminar un dispositivo incluido previamente desde la aplicación
Curso típico de eventos	
Actor	Sistema
1. Pulsa el interruptor de la escena	2. El sistema conmuta el interruptor
	3. La API de Thingier realiza varias peticiones PUT para modificar el estado de todas las acciones.
Curso alternativo de eventos	
3.1.Si se produjese algún error modificando el estado de alguna acción, se notificaría al usuario.	

Tabla 91. UC_18 Activar / desactivar escena

4.3.2.7. Thinger

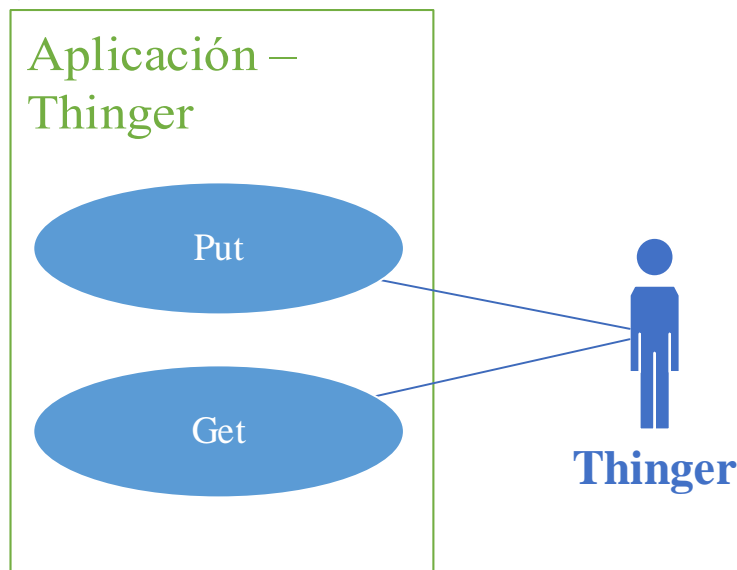


Ilustración 39. Casos de uso Thinger

Identificador	UC_19
Caso de uso	Petición Put
Actores	Thinger
Prioridad	Alta
Fuente	Cliente
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación
Post-condiciones	Se devuelve al sistema si la petición se ha completado con éxito.
Descripción	Envía mediante una petición API REST el valor deseado al servidor Thinger, para establecer un valor concreto.
Curso típico de eventos	
Actor	Sistema
	1. Se genera una petición con los valores deseados.
2. Se envía la petición.	
3. Una vez devueltos los resultados, se pasan al sistema.	
Curso alternativo de eventos	

Tabla 92. UC_19 Petición Put

Identificador	UC_20
Caso de uso	Petición Get
Actores	Thinger
Prioridad	Alta
Fuente	Cliente
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación
Post-condiciones	Se devuelve al sistema si la petición se ha completado con éxito.
Descripción	Envía mediante una petición API REST el valor deseado al servidor Thinger, para recibir un valor concreto.
Curso típico de eventos	
Actor	Sistema
	1. Se genera una petición con los valores deseados.
2. Se envía la petición.	
3. Una vez devueltos los resultados, se pasan al sistema.	
Curso alternativo de eventos	

Tabla 93. UC_20 Petición Get

4.3.2.8. *Firestore*

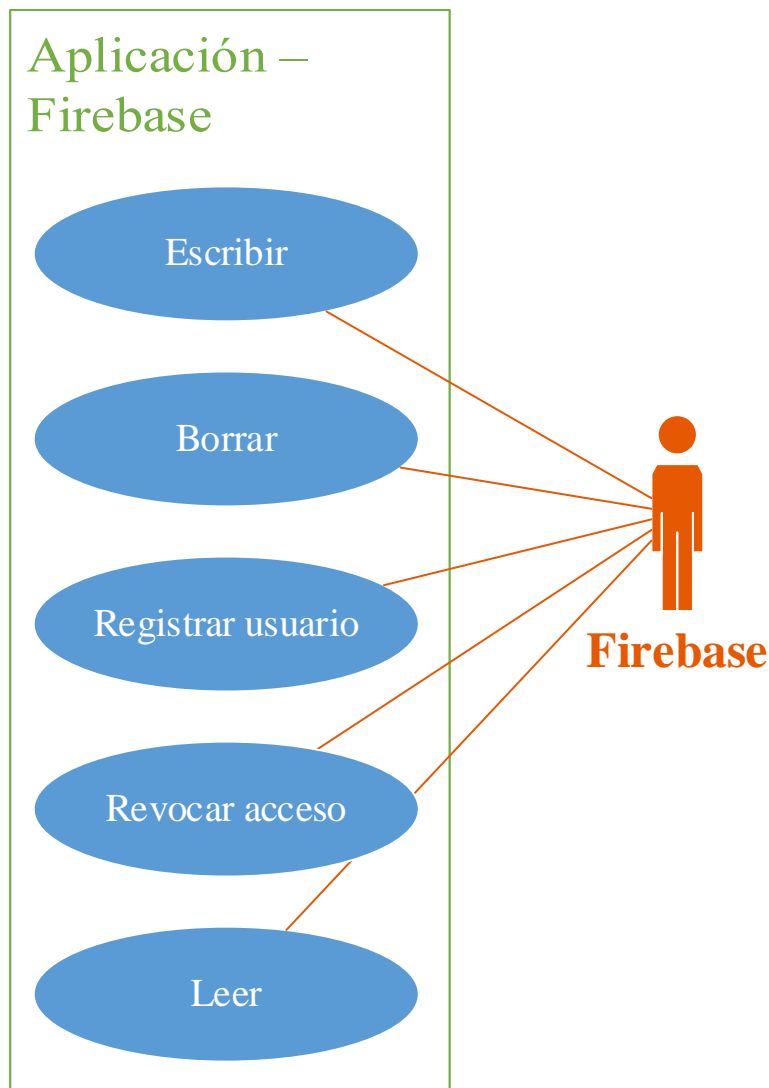


Ilustración 40. Casos de uso Firestore

Identificador	UC_21
Caso de uso	Escribir en la Base de Datos
Actores	Firestore
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación
Post-condiciones	Se almacena la información en la base de datos, y se devuelve al sistema si se ha realizado con éxito.
Descripción	Envía mediante una API el objeto deseado, en una posición concreta, a la base de datos.

Curso típico de eventos	
Actor	Sistema
	1. Se genera el objeto a almacenar y la posición dentro de la base de datos.
2. Se envía el objeto.	
3. Una vez recibido el resultado, se le devuelve al sistema.	
Curso alternativo de eventos	

Tabla 94. UC_21 Escribir en la Base de Datos

Identificador	UC_22
Caso de uso	Borrar de la Base de Datos
Actores	Firebase
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación
Post-condiciones	Se elimina la información de la base de datos, y se devuelve al sistema si se ha realizado con éxito.
Descripción	Envía mediante una API la posición concreta de la base de datos a eliminar.
Curso típico de eventos	
Actor	Sistema
	1. Se pasa al actor la posición dentro de la base de datos a eliminar.
2. Se envía la petición.	
3. Una vez recibido el resultado, se le devuelve al sistema.	
Curso alternativo de eventos	

Tabla 95. UC_22 Borrar de la Base de Datos

Identificador	UC_23
Caso de uso	Revocar acceso de la Base de Datos
Actores	Firebase
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación
Post-condiciones	Se elimina la información del usuario de la base de datos, y se devuelve al sistema si se ha realizado con éxito.

Descripción	Envía mediante una API la petición de eliminar dicho usuario de la base de datos.
Curso típico de eventos	
Actor	Sistema
	1. Se pasa al actor el identificador del usuario de la base de datos a eliminar.
2. Se envía la petición.	
3. Una vez recibido el resultado, se le devuelve al sistema.	
Curso alternativo de eventos	

Tabla 96. UC_23 Revocar acceso de la Base de Datos

Identificador	UC_24
Caso de uso	Leer de la Base de Datos
Actores	Firestore
Prioridad	Alta
Fuente	Analista
Pre-condiciones	Tener acceso a Internet Haber iniciado sesión en la aplicación
Post-condiciones	Se obtiene la información requerida de la base de datos.
Descripción	Envía mediante una API la petición para obtener dicha información de la base de datos.
Curso típico de eventos	
Actor	Sistema
	1. Se pasa al actor la posición de la base de datos a obtener.
2. Se envía la petición.	
3. Una vez recibido el resultado, se le devuelve al sistema.	
Curso alternativo de eventos	

Tabla 97. UC_24 Leer de la Base de Datos

4.3.3. Matriz de trazabilidad

Por último, se realiza una matriz de trazabilidad, para observar de manera directa qué requisitos funcionales afectan a cada uno de los casos de uso.

4.3.3.1. Lista de casos de uso

- UC_01: Inicio de sesión con cuenta de Google
- UC_02: Nuevo dispositivo
- UC_03: Nueva escena
- UC_04: Gestionar dispositivo
- UC_05: Gestionar escena
- UC_06: Gestionar localización
- UC_07: Gestionar perfil
- UC_08: Acerca de la aplicación
- UC_09: Cerrar sesión
- UC_10: Revocar acceso
- UC_11: Cambiar localización
- UC_12: Modificar dispositivo
- UC_13: Borrar dispositivo
- UC_14: Cambiar estado acción
- UC_15: Recargar sensores
- UC_16: Modificar escena
- UC_17: Borrar escena
- UC_18: Activar / desactivar escena
- UC_19: Petición Put
- UC_20: Petición Get
- UC_21: Escribir en la Base de Datos
- UC_22: Borrar de la Base de Datos
- UC_23: Revocar acceso de la Base de Datos
- UC_24: Leer de la Base de Datos

4.3.3.2. Matriz de trazabilidad

RFS \ UC	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
01	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓			✓	✓				✓	✓	✓	✓	
02	✓						✓		✓	✓											✓	✓	✓	✓	
03	✓					✓					✓										✓	✓		✓	
04		✓		✓								✓	✓	✓	✓				✓	✓	✓	✓		✓	
05		✓	✓	✓	✓							✓		✓		✓		✓	✓	✓	✓	✓		✓	
06			✓		✓								✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	
07		✓		✓								✓	✓	✓	✓			✓	✓	✓	✓	✓		✓	
08	✓	✓	✓							✓	✓	✓	✓		✓	✓	✓	✓			✓	✓	✓	✓	
09	✓																				✓			✓	
10	✓					✓					✓										✓			✓	
11									✓																
12										✓													✓	✓	
13		✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓		✓	✓	✓							
14				✓	✓	✓						✓	✓	✓		✓	✓	✓							
15				✓	✓	✓					✓											✓	✓	✓	✓
16				✓								✓	✓	✓								✓	✓		✓
17					✓											✓	✓	✓				✓	✓		✓
18		✓	✓																			✓	✓		✓
19		✓																		✓		✓			✓
20		✓																			✓	✓			✓
21			✓																				✓		✓
22				✓								✓	✓	✓	✓				✓	✓	✓	✓			✓
23				✓								✓	✓	✓	✓				✓	✓	✓	✓			✓
24					✓											✓	✓	✓	✓		✓	✓			✓
25					✓											✓	✓	✓	✓		✓	✓			✓
26						✓					✓										✓	✓			✓
27						✓					✓										✓				✓
TOTAL	6	9	7	10	9	8	2	1	3	4	8	10	10	9	5	9	8	9	9	7	22	19	5	23	

Tabla 98. Matriz de Trazabilidad

4.4. Priorización

En este apartado se definen los casos de uso a realizar en cada uno de los ciclos de desarrollo del proyecto. Estos se determinarán mediante una ponderación de características significativas del proyecto.

4.4.1. Características

Las características a evaluar por cada caso de uso son:

4.4.1.1. *Al realizar el caso de uso, tendré una mejor comprensión del diseño.*

Esta característica pondera cuán importante es si el caso de uso ayuda a tener una mejor comprensión del diseño del sistema final, con lo que ello supone para futuros casos de uso.

Para esta característica se ha decidido otorgar un 20% de valor de ponderación, puesto que existen algunos casos de uso que permitirán definir mejor la aplicación final.

4.4.1.2. *Uso de tecnología no conocida*

Esta característica observa si se tiene conocimiento sobre la tecnología a utilizar en dicho caso de uso, pudiendo costar así un mayor tiempo de desarrollo en acostumbrarse a ella.

Se ha decidido otorgar un 15% puesto que diversos casos de uso hacen uso de varios sistemas y la unión de ellos puede conllevar un tiempo de adaptación hasta producir correctamente.

4.4.1.3. *Dificultad de desarrollo.*

Esta característica pondera el nivel de complejidad, y por tanto de tiempo invertido en desarrollar las funcionalidades de un caso de uso.

Como las funciones del sistema, a priori, no destacan por una alta complejidad, se decide otorgarle un 15% de valor de ponderación.

4.4.1.4. *Importancia en el proyecto.*

Esta característica pondera la importancia de cada caso de uso desde el punto de vista del cliente, pudiendo así ofrecerle en cada etapa un producto con la máxima funcionalidad posible. De esta manera, el cliente verá claramente como su producto aumenta con cada una de las iteraciones que realicemos.

Debido a la importancia de esta característica se decide otorgarle la máxima ponderación, un 50%.

4.4.2. Ponderación

Una vez concretadas las características a valorar, y su ponderación, se elabora la ponderación. Para ello, valoramos la influencia del caso de uso con cada una de las características, y obtenemos la suma del total.

Las valoraciones respecto a cada caso de uso abarcan los números naturales del 1 al 5, siendo un 0 la nota más baja y un 5 la más alta.

Las características se enumeran con letras, siendo “a” la característica “Al realizar el caso de uso, tendré una mejor comprensión del diseño.”.

PONDERACIÓN		0,2	0,15	0,15	0,5
Caso de uso / Característica	a	b	c	d	PONDERADO
UC_01: Inicio de sesión con cuenta de Google	4	3	2	5	4,05
UC_02: Nuevo dispositivo	4	5	2	5	4,35
UC_03: Nueva escena	3	4	3	4	3,65
UC_04: Gestionar dispositivo	4	3	3	5	4,2
UC_05: Gestionar escena	3	3	3	4	3,5
UC_06: Gestionar localización	3	3	2	4	3,35
UC_07: Gestionar perfil	4	2	1	3	2,75
UC_08: Acerca de la aplicación	1	2	1	1	1,15
UC_09: Cerrar sesión	4	1	0	3	2,45
UC_10: Revocar acceso	4	1	1	3	2,6
UC_11: Cambiar localización	2	2	1	2	1,85
UC_12: Modificar dispositivo	4	2	3	3	3,05
UC_13: Borrar dispositivo	4	2	1	2	2,25
UC_14: Cambiar estado acción	5	4	3	5	4,55
UC_15: Recargar sensores	2	2	3	2	2,15
UC_16: Modificar escena	3	2	4	3	3
UC_17: Borrar escena	3	2	3	2	2,35
UC_18: Activar / desactivar escena	4	4	2	2	2,7
UC_19: Petición Get	4	4	3	5	4,35
UC_20: Petición Put	4	4	3	5	4,35
UC_21: Escribir en la Base de Datos	5	5	4	5	4,85
UC_22: Borrar de la Base de Datos	5	5	4	5	4,85
UC_23: Revocar acceso de la Base de Datos	5	5	2	5	4,55
UC_24: Leer de la Base de Datos	5	5	3	5	4,7
TOTAL	89	75	57	88	81,6

Tabla 99. Ponderación casos de uso

Como podemos observar, el caso de uso UC_21 y UC_22 es el que mayor ponderación ha resultado tener, quedando patente su importancia en este proyecto, mientras que UC_08 es el que menos. A continuación, y tratando de realizar iteraciones que garanticen a su final un producto mínimo viable que enseñar al cliente, se obtienen los siguientes ciclos de desarrollo:

4.4.3. Iteraciones

4.4.3.1. Iteración 1

Esta primera iteración se compone de los casos de uso: UC_01, UC_07, UC_09, UC_10, UC_21, UC_22, UC_23 y UC_24, puesto que han sacado la nota más alta de la tabla de priorización. Cabe destacar que los casos de uso UC_07, UC_09 y UC_10 no cuentan con alta ponderación, pero son necesarios para completar el producto mínimo viable, y permitir al usuario iniciar y salir de la aplicación.

Por tanto, este primer ciclo de desarrollo abarca la gestión de los usuarios y el control de la plataforma Firebase, para gestionar la base de datos. Además, cabe destacar que para todo este desarrollo está implícito la creación de la estructura del menú principal, por lo que conlleva un esfuerzo mayor de lo aquí enseñado.

Finalmente, se muestra el resumen de esta primera iteración:

Caso de uso / Característica	a	b	c	d	PONDERADO
UC_01: Inicio de sesión con cuenta de Google	4	3	2	5	4,05
UC_07: Gestionar perfil	4	2	1	3	2,75
UC_09: Cerrar sesión	4	1	0	3	2,45
UC_10: Revocar acceso	4	1	1	3	2,6
UC_21: Escribir en la Base de Datos	5	5	4	5	4,85
UC_22: Borrar de la Base de Datos	5	5	4	5	4,85
UC_23: Revocar acceso de la Base de Datos	5	5	2	5	4,55
UC_24: Leer de la Base de Datos	5	5	3	5	4,7
				TOTAL	30,8

Tabla 100. Priorización iteración 1

4.4.3.2. Iteración 2

En la segunda iteración de desarrollo se abarcarán los siguientes casos de uso:

Caso de uso / Característica	a	b	c	d	PONDERADO
UC_02: Nuevo dispositivo	4	5	2	5	4,35
UC_04: Gestionar dispositivo	4	3	3	5	4,2
UC_12: Modificar dispositivo	4	2	3	3	3,05
UC_13: Borrar dispositivo	4	2	1	2	2,25
UC_14: Cambiar estado acción	5	4	3	5	4,55
UC_15: Recargar sensores	2	2	3	2	2,15
UC_19: Petición Get	4	4	3	5	4,35
UC_20: Petición Put	4	4	3	5	4,35
TOTAL					29,25

Tabla 101. Priorización iteración 2

En esta fase se hará hincapié en crear la funcionalidad relacionada con los dispositivos y su control, mediante la plataforma Firebase y creando la funcionalidad necesaria para que el cliente observe un sencillo ejemplo con un dispositivo.

Como podemos observar, la suma de ponderación de esta iteración desciende respecto a la anterior. Esto deja claro que la fase inicial es más importante que esta.

4.4.3.3. Iteración 3

La última iteración del proyecto contendrá los casos de uso restantes, que son:

Caso de uso / Característica	a	b	c	d	PONDERADO
UC_03: Nueva escena	3	4	3	4	3,65
UC_05: Gestionar escena	3	3	3	4	3,5
UC_06: Gestionar localización	3	3	2	4	3,35
UC_08: Acerca de la aplicación	1	2	1	1	1,15
UC_11: Cambiar localización	2	2	1	2	1,85
UC_16: Modificar escena	3	2	4	3	3
UC_17: Borrar escena	3	2	3	2	2,35
UC_18: Activar / desactivar escena	4	4	2	2	2,7
TOTAL					21,55

Tabla 102. Priorización iteración 3

Se realizarán las escenas que puede generar el usuario, así como las funciones de gestionar las localizaciones.

4.5. Planificación y presupuesto

En este apartado se adjunta la planificación y el presupuesto final del proyecto, con el fin de detallar de la manera más exacta posible los recursos, tanto económicos como temporales, utilizados por el equipo de desarrollo para la realización de este proyecto.

4.5.1. Planificación

La planificación utilizada se rige por la priorización de casos de uso realizada anteriormente, y tiene en cuenta los distintos apartados de la presente memoria.

Recordar que los tiempos mostrados a continuación son lo más precisos posibles al haber sido parametrizados con la aplicación TimeCamp, la cual, como se explicó anteriormente, contabiliza de manera precisa el número de horas destinadas a cada herramienta del proyecto.

Este proyecto comenzó el día 23 de enero de 2017 y finalizó el 21 de junio del mismo año. Se han trabajado un total de 260 horas en el proyecto, repartidas en 108 días de trabajo, de los cuales 80 de ellos superaban las tres horas dedicadas al proyecto. El promedio de horas por día fue de aproximadamente tres horas y media al día.

Como hemos explicado, se va a seguir una metodología Craig-Larman modificada, por lo que tenemos tres grandes fases:

1. Planificación y especificación de requisitos: se realizan los apartados de la memoria que no requieren del producto software final.
2. Construcción: se realizan las tres iteraciones de desarrollo que se detallaron en el apartado anterior, así como las correspondientes pruebas de cada caso de uso.
3. Finalización: se finaliza la memoria con los apartados que requieren del producto software final, así como la maquetación para la entrega de la memoria.

A continuación, se adjunta el diagrama Gantt, en el que observar cada una de las tareas el tiempo real que han requerido.

4.5.1.1. Diagrama Gantt

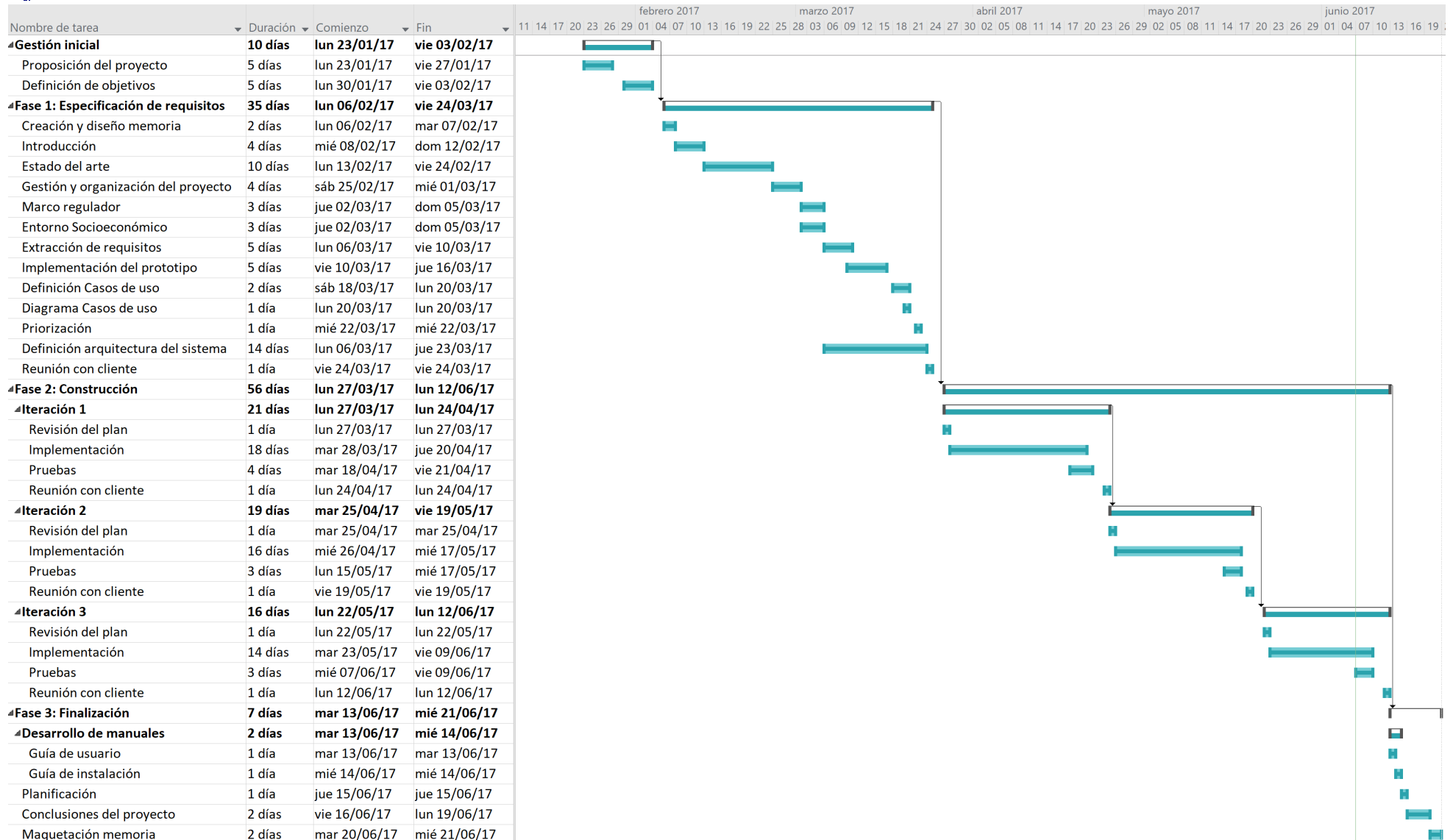


Ilustración 41. Diagrama Gantt

4.5.2. Presupuesto

A continuación, con los recursos utilizados (indicados en el apartado Organización del proyecto), y la planificación, mostramos desglosa el cálculo del presupuesto final del proyecto.

4.5.2.1. Recursos humanos

Antes de proceder al cálculo del importe de cada rol del desarrollo, es necesario fijar el salario de cada uno de los mismos. Para ello, utilizamos como base los datos proporcionados por el “XVI Convenio Colectivo Estatal de empresas consultoras, de planificación, organización de empresa y contable, empresas de servicios de informática y de estudios de mercado y de la opinión pública” [31], al ser el último convenio aprobado por el BOE para este tipo de empresa.

A pesar de ser antiguo, puesto que es de 2009, queda claro que es aplicable para este tipo de empresa, al ser prorrogable anualmente. A continuación, se deja el artículo 1 del mismo para observar la aplicabilidad del mismo:

Artículo 1. *Ámbito funcional.*

1. El presente Convenio será de obligada observancia en todas las empresas consultoras de planificación, organización de empresas y contable, cuyas actividades de servicios de consultoría en selección y formación de recursos humanos, técnicas de organización y dirección de empresas, auditoría, y cualesquiera otras de orden similar, vinieran rigiéndose por el XV Convenio de Empresas Consultoras de Planificación, Organización de Empresas y Contable, al que sustituye íntegramente en dicho ámbito.

2. También están incluidas en el ámbito funcional del presente Convenio, y obligadas por él, las empresas de servicios de informática, así como las de investigación de mercados y de la opinión pública que vinieran rigiéndose por el antes citado Convenio Colectivo.

A continuación, se dejan las profesiones, por categoría y función, que se requieren en este caso, detalladas en el artículo 15 del mismo:

Personal titulado: Es el que se halla en posesión de un título o diploma oficial de grado superior o medio, que está unido a la empresa por un vínculo de relación laboral concertado en razón del título que posee, para ejercer funciones específicas para las que el mismo le habilita y siempre que preste sus servicios en la empresa con carácter exclusivo o preferente por un sueldo o tanto alzado, sin sujeción a la escala habitual de honorarios de su profesión.

Jefe/a de Operación: Es el trabajador que, con personal a su cargo, tiene bajo su responsabilidad:

- La planificación del trabajo a realizar en cada uno de los ordenadores, turnos y particiones.
- La asignación de recursos humanos en cada puesto de trabajo y en cada momento.
- La coordinación con Análisis y Programación para la puesta a punto de las cadenas y para las pruebas que haya que realizar.
- La coordinación con Técnica de Sistemas para el mejor aprovechamiento del sistema.
- El conocimiento, en todo momento, del sistema operativo y del trabajo para tomar decisiones en las dudas que se presenten en operación.
- Llevar estadísticas de averías y control de mantenimiento preventivo en todas las máquinas.
- La conservación de los manuales de operación.

Analista de Sistemas: Le corresponde el diseño, puesta a punto y mantenimiento de los sistemas operativos a utilizar en los procesos de mecanización.

- Formarse e informarse de todo lo concerniente al proceso de datos, componentes físicos («hardware»), componentes lógicos («software»), aplicaciones, organización.
- Asesorar y coordinar con todo el personal de la empresa sobre las posibilidades de proceso de datos.

Programador/a Senior: Es el trabajador que debe tener un conocimiento profundo de las técnicas y recursos que maneja, enfocado principalmente a los lenguajes de programación existentes en el ordenador que utiliza así como de las facilidades y ayuda que le presta al «software» para la puesta a punto de programas, correspondiéndole estudiar los problemas complejos definidos por los Analistas, confeccionando organigramas detallados de tratamiento.

Le corresponde redactar programas en el lenguaje de programación que le sea indicado. Asimismo, confecciona juegos de ensayo, pone a punto los programas y completa los expedientes técnicos de los mismos.

Como se indica en el artículo 20, la jornada laboral máxima será de 1800 horas anuales, por lo que, con este dato y las tablas salariales mínimas, somos capaces de calcular el sueldo / hora.

A continuación, se presenta la tabla que reúne el importe a percibir por cada trabajador:

Categoría	Salario Mes (x14) (Euros)	Salario Año (Euros)	Plus Mes (x14) (Euros)	Plus Año (Euros)	Total Año (Euros)
Titulado/a de grado superior	1569.25	21969.50	109.73	1536.22	23505.72
Analista y Analista de sistemas	1569.25	21969.50	109.73	1536.22	23505.72
Programador/a senior	103.04	15442.56	77.80	1089.20	16531.76

Tabla 103. Importe salario mínimo rol

A estos salarios mínimos, debemos añadir el importe destinado a la Seguridad Social por parte de la empresa. Estos datos, encontrados en la web de la Seguridad Social [32], se resumen en el siguiente desglose:

Contingencias	Empresa
Comunes	23.60%
Desempleo	5.50%
FOGASA	0.20%
Formación Profesional	0.60%
IT (Incapacidad Temporal)	0.65%
IMS (Incapacidad Permanente, Muerte y Supervivencia)	0.45%
TOTAL	31%

Tabla 104. Desglose Seguridad Social

Sabemos también que este año laboral, 2017, en la comunidad de Madrid, está compuesto por 245 días laborables [33], teniendo derecho a mínimo 30 días naturales de vacaciones [34]. Aproximadamente, 215 días laborables este año.

Suponiendo jornada laboral de 8 horas / día, podemos calcular que el sueldo por hora mínimo para cada uno de los roles implicados es:

Categoría	Total Salario Año (Euros)	Seguridad Social 31% (Euros)	Coste final Anual (Euros)	Coste · hora (Euros)
Titulado/a de grado superior	23505,72	7286,7732	30792,4932	17,90261233
Analista y Analista de sistemas	23505,72	7286,7732	30792,4932	17,90261233
Programador/a senior	16531,76	5124,8456	21656,6056	12,59104977

Tabla 105. Coste por hora final rol

Estos sueldos mínimos no se ajustan al mercado laboral actual. La oferta del cliente elevó estos salarios en 10 euros más por coste · hora, al ser el precio habitual al que contrató en este tipo de proyectos.

Finalmente, tras analizar el salario de cada uno de los roles, mostramos la siguiente tabla el importe pagado por la empresa por cada rol que interviene en el desarrollo, y el total de horas de trabajo invertidas, calculando finalmente el coste total invertido en cada rol.

Personal	Tarea	Dedicación (horas)	Sueldo/hora (euros)	Coste (euros)
Amigo Herrero, Daniel	Jefe de proyecto	66,5	27,9	1855,35
	Analista	136,5	27,9	3808,35
	Programador	175	22,59	3953,25
	Total	378	78,39	9616,95

Tabla 106. Presupuesto recursos humanos

4.5.2.2. Recursos hardware

En la siguiente tabla se listan los recursos hardware empleados, con su correspondiente amortización:

Hardware	Coste (euros)	Tiempo de uso (meses)	Periodo deprecación (meses)	% de uso al proyecto	Amortización
Ordenador portátil	900	5	60	85	63,75
Dispositivo Móvil	300	5	24	65	40,625
NodeMCU	9	5	24	100	1,875
				TOTAL	106,25

Tabla 107. Presupuesto recursos hardware

4.5.2.3. Recursos software

En la siguiente tabla se listan los recursos hardware empleados. Indicar que no se realiza amortización de los mismos al ser suscripciones mensuales.

Software	Coste	Tiempo de uso (meses)	Coste final (euros)
Windows 10	0 €	5	0
Microsoft Office 365 Empresa	8,8€/ mes	5	44
Microsoft Visio 2016	11€/ mes	5	55
Microsoft Project 2016	25€/ mes	1	125
Android Studio	0 €	5	0
Google Drive	1,99€/ mes	5	9,95
TimeCamp	7€/ mes	5	35
Axure	29€/ mes	1	145
Adobe Acrobat Pro DC	18,14€/ mes	5	90,7
Visual Paradigm Standard	19€/ mes	1	95
		TOTAL	599,65

Tabla 108. Presupuesto recursos software

4.5.2.4. Costes indirectos

Por último, en la siguiente tabla se recogen los gastos no contemplados anteriormente.

Concepto	Coste	Tiempo de uso (meses)	Coste final (euros)
Luz	40€/ mes	5	200
Agua	15€/ mes	5	75
Internet	40€/ mes	5	200
Transporte	20€/ mes	5	100
		TOTAL	575

Tabla 109. Presupuesto costes indirectos

4.5.2.5. Cálculo total de costes

Una vez desglosados los costes, mostramos un resumen con los costes finales del proyecto.

Concepto	Coste (euros)
Recursos humanos	9616,95
Recursos hardware	106,25
Recursos software	599,65
Costes indirectos	575
TOTAL SIN IVA	10897,85
TOTAL CON IVA	13186,40

Tabla 110. Presupuesto cálculo total de costes

El coste total del proyecto es de **trece mil ciento ochenta y seis euros con cuarenta céntimos**.

5. *Diseño de la aplicación*

En este apartado se describirá de manera detallada el diseño de la aplicación y cómo interactúan los distintos sub-sistemas entre sí. Además, se definirá de una manera simple el diseño y funcionamiento de la base de datos utilizada.

5.1. *Diseño de la arquitectura*

A continuación, procedemos a analizar la estructura a implementar para el funcionamiento de la aplicación.

La aplicación sigue una arquitectura de diseño basada en el Modelo-Vista-Controlador (MVC) [35]. Esta separa los distintos aspectos de la aplicación en tres capas. Esta arquitectura es la más extendida en desarrollo Android, y permite estructurar de manera simple y organizada los datos de la aplicación, separando los datos de cada usuario de la interfaz, y a su vez de la lógica de la aplicación.

En nuestra aplicación el modelo interactuará de la siguiente manera:

- **Modelo:** se refiere a todo dato que es manejado por la aplicación, el cual está disponible para el resto de capas. En este caso, definidos por el sistema Firebase, cuyo diseño se explicará posteriormente.
- **Vista:** esta capa agrupa toda la interfaz de usuario UI. Por tanto, se compone de todos los ficheros XML que definen el estilo de la aplicación (layouts, menus, etc.).
- **Controlador:** la última capa del sistema es la más importante, puesto que es la que relaciona las otras dos capas, mediante la lógica de la aplicación. Esta lógica responde de una manera determinada a los eventos que realiza el usuario en la interfaz, con los datos correspondientes. Por ejemplo, al añadir un dispositivo, la capa “controlador” obtiene los datos introducidos por el usuario en la capa vista, y se entrega de manera que la capa modelo por almacene. Esta capa se compone de todas las clases JAVA realizadas durante el proyecto. Indicar que el sub-sistema Thinger pertenece a esta capa, puesto que es accionado mediante peticiones que se realizan desde las clases JAVA.

En este caso, las distintas capas se reparten por la aplicación de la siguiente manera:

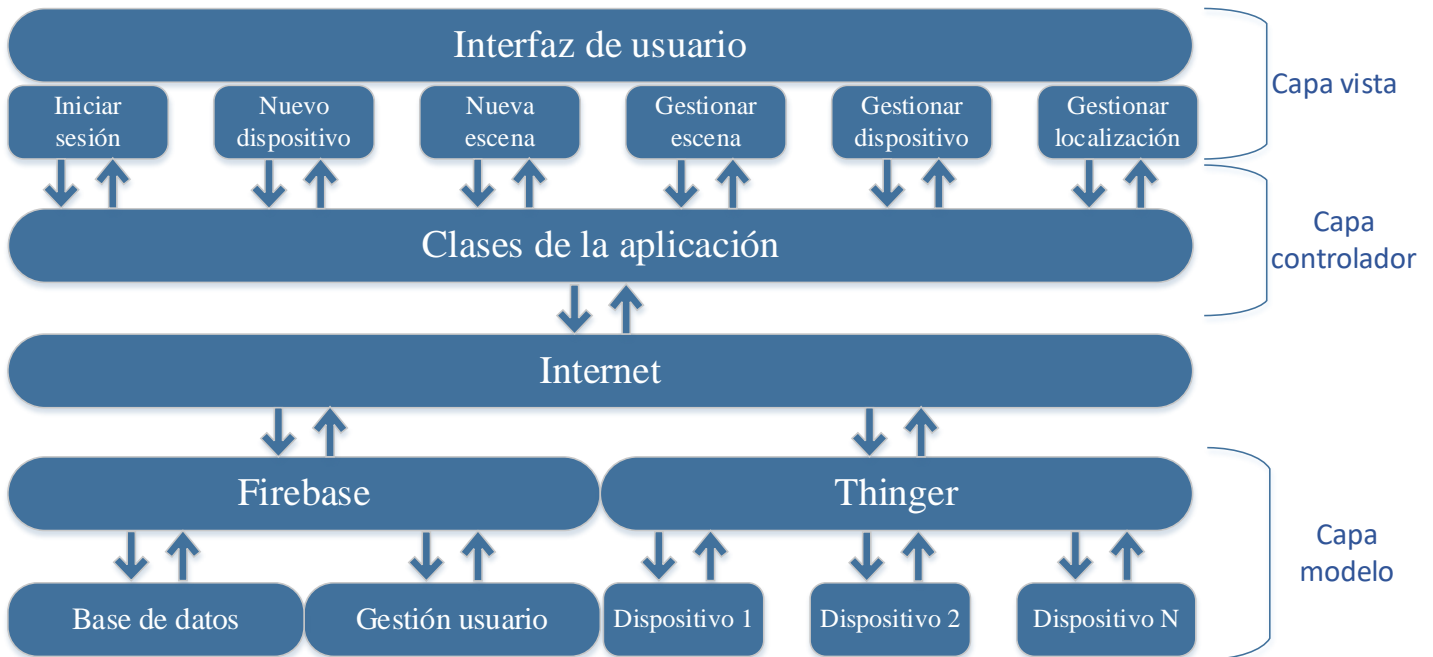


Ilustración 42. Diseño de la arquitectura

La capa vista agrupa toda la interfaz de usuario creada, que se explica detalladamente en el manual de uso.

La capa controlador está formada por todas las clases JAVA que conforman la aplicación, además de la parte de Thinger que activa y desactiva los dispositivos.

Por último, la capa modelo está compuesta por la base de datos de Firebase, su sistema de usuarios, y los datos obtenidos de Thinger referente a cada acción de los dispositivos.

5.2. Diagramas de actividad

A continuación, se van a presentar los diagramas de actividad relativos a los flujos de cada acción a realizar por el usuario. De esta manera se entiende perfectamente, de manera gráfica, el flujo realizado por la aplicación. Estos diagramas están basados en los casos de uso realizados anteriormente, realizando únicamente los que cuentan con funcionalidad por parte del usuario.

Para realizar los siguientes diagramas se ha utilizado el programa Visual Paradigm, utilizando las componentes del estándar UML, que son:

Nombre	Dibujo	Descripción
Nodo inicial		Representa el comienzo del flujo.
Nodo final		Representa el final del flujo.
Nodo de decisión		Representa una partición en el flujo debido a una decisión.
Acción		Representa una acción a realizar.
Partición		Divide el flujo en función de qué sistema sea el responsable de cada parte.

Tabla 111. Componentes diagrama de actividad

5.2.1. Iniciar sesión

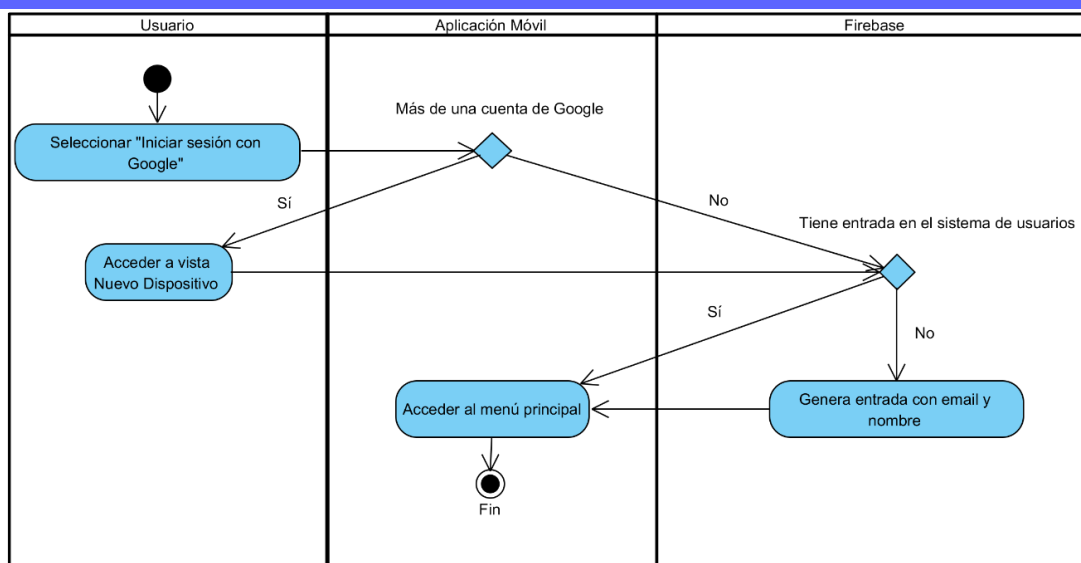


Ilustración 43. Diagrama de actividad Iniciar sesión

5.2.2. Nuevo dispositivo

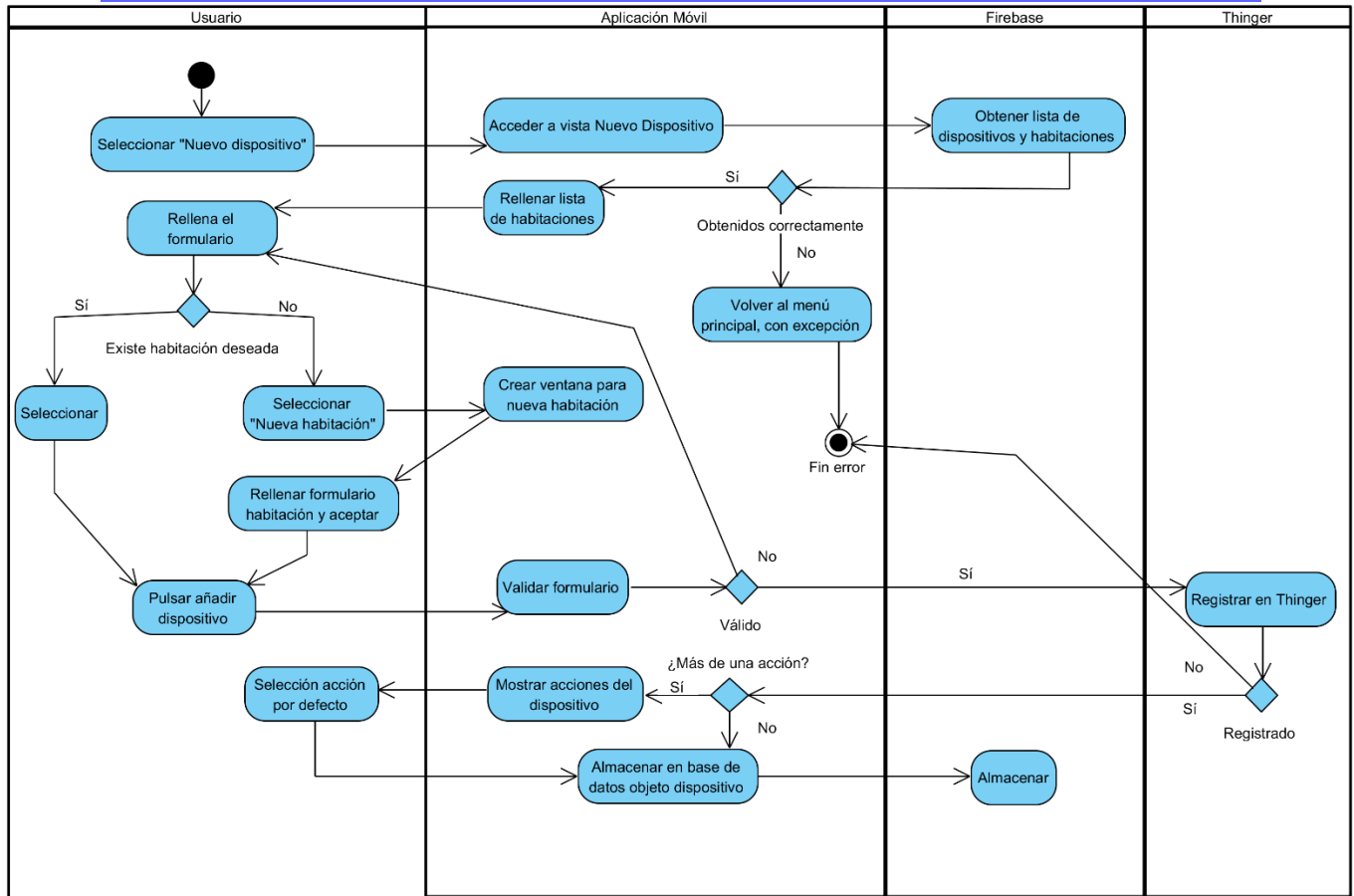


Ilustración 44. Diagrama de actividad Nuevo dispositivo

5.2.3. Nueva escena

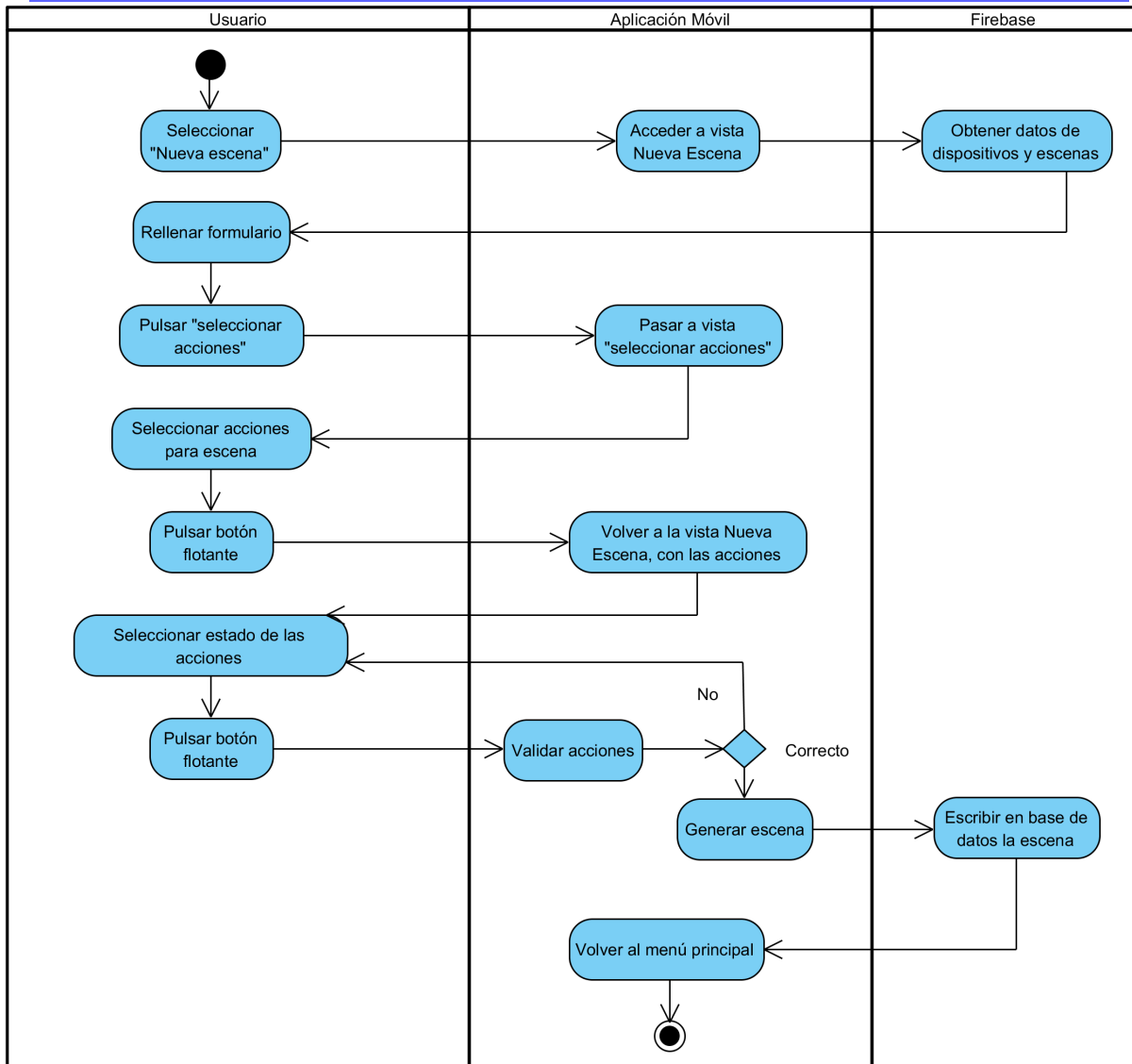


Ilustración 45. Diagrama de actividad Nueva escena

5.2.4. Gestionar dispositivo

5.2.4.1. Modificar dispositivo

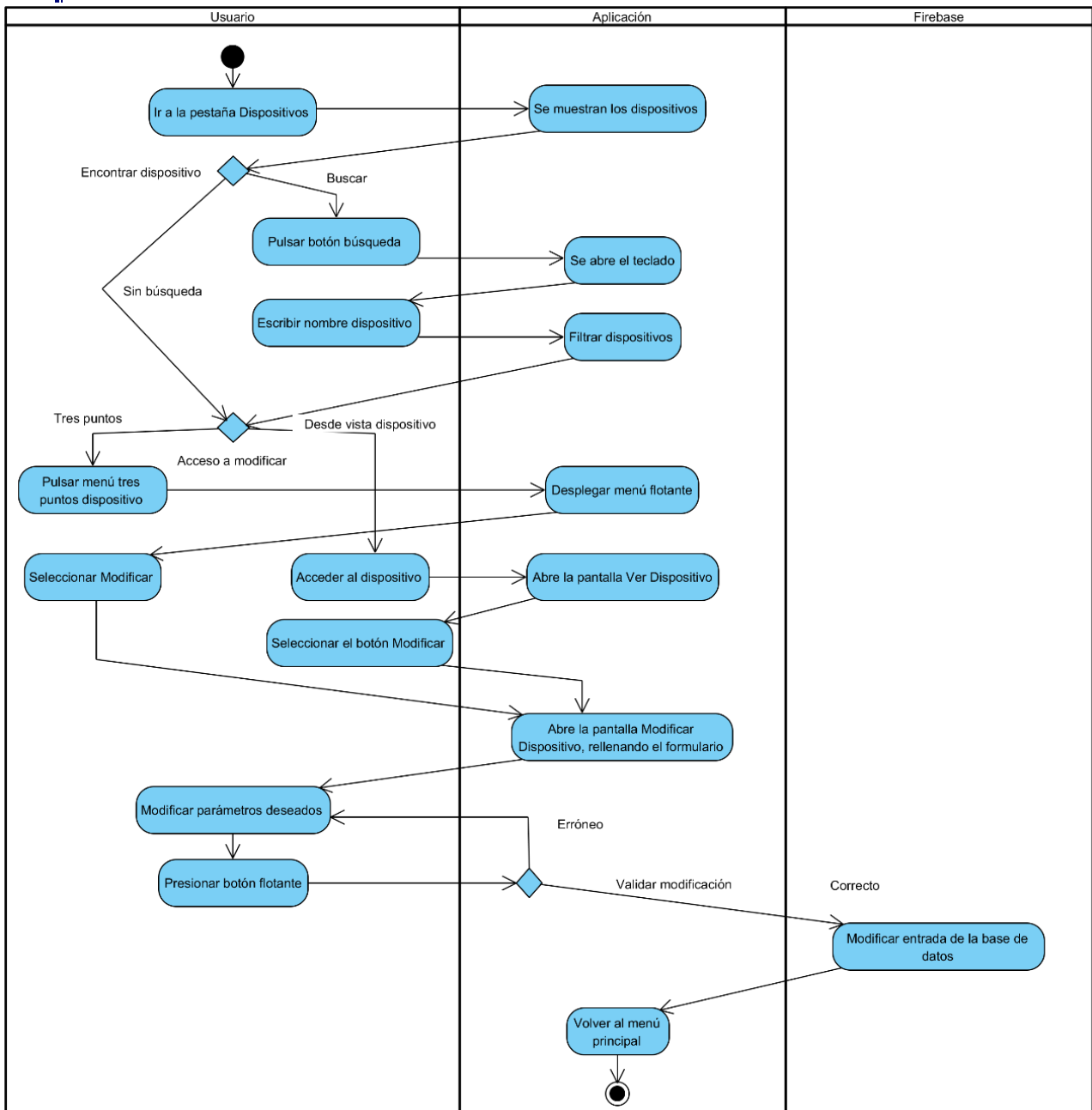


Ilustración 46. Diagrama de actividad Modificar dispositivo

5.2.4.2. Borrar dispositivo

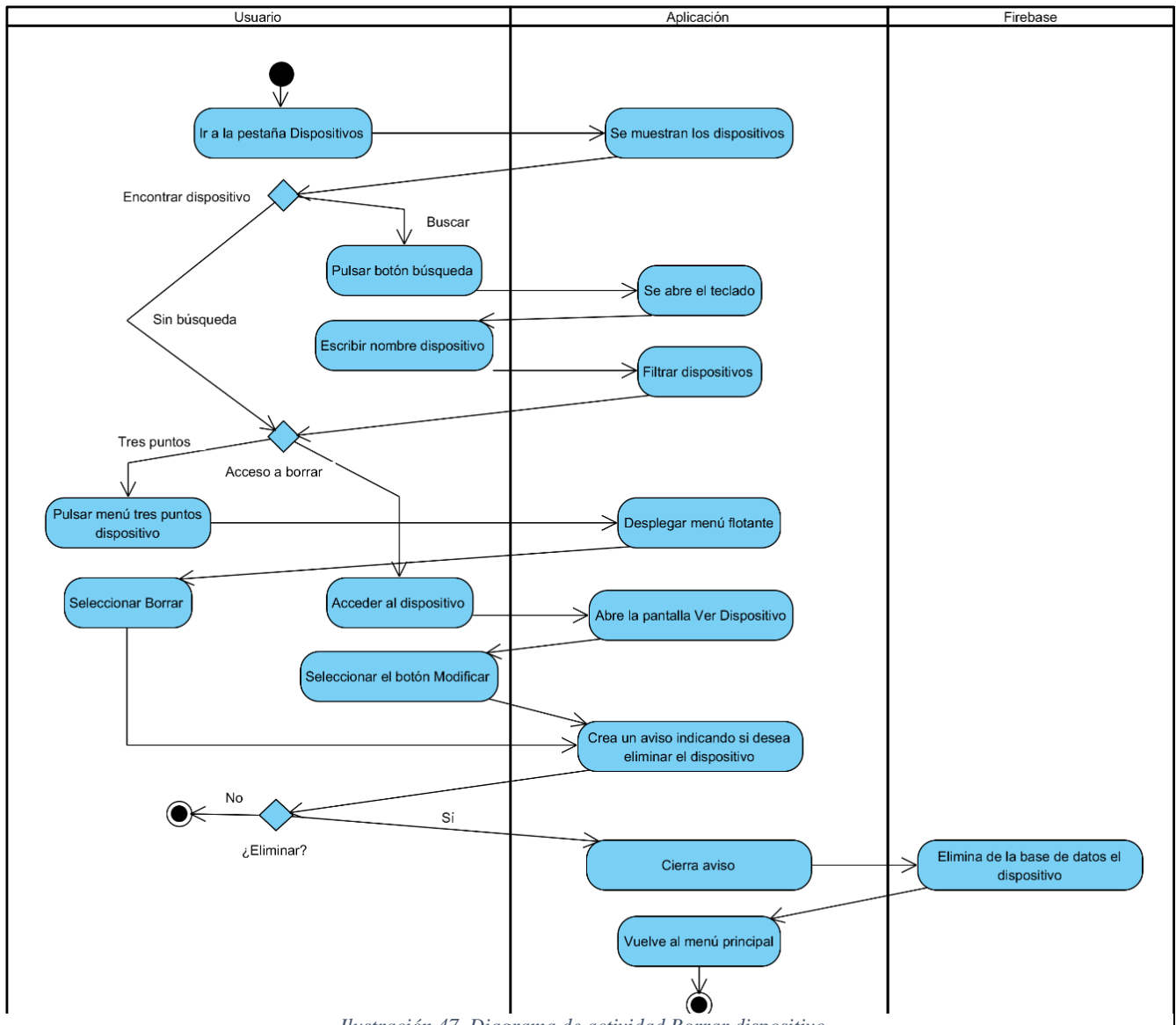


Ilustración 47. Diagrama de actividad Borrar dispositivo

5.2.4.3. Ver valores sensores

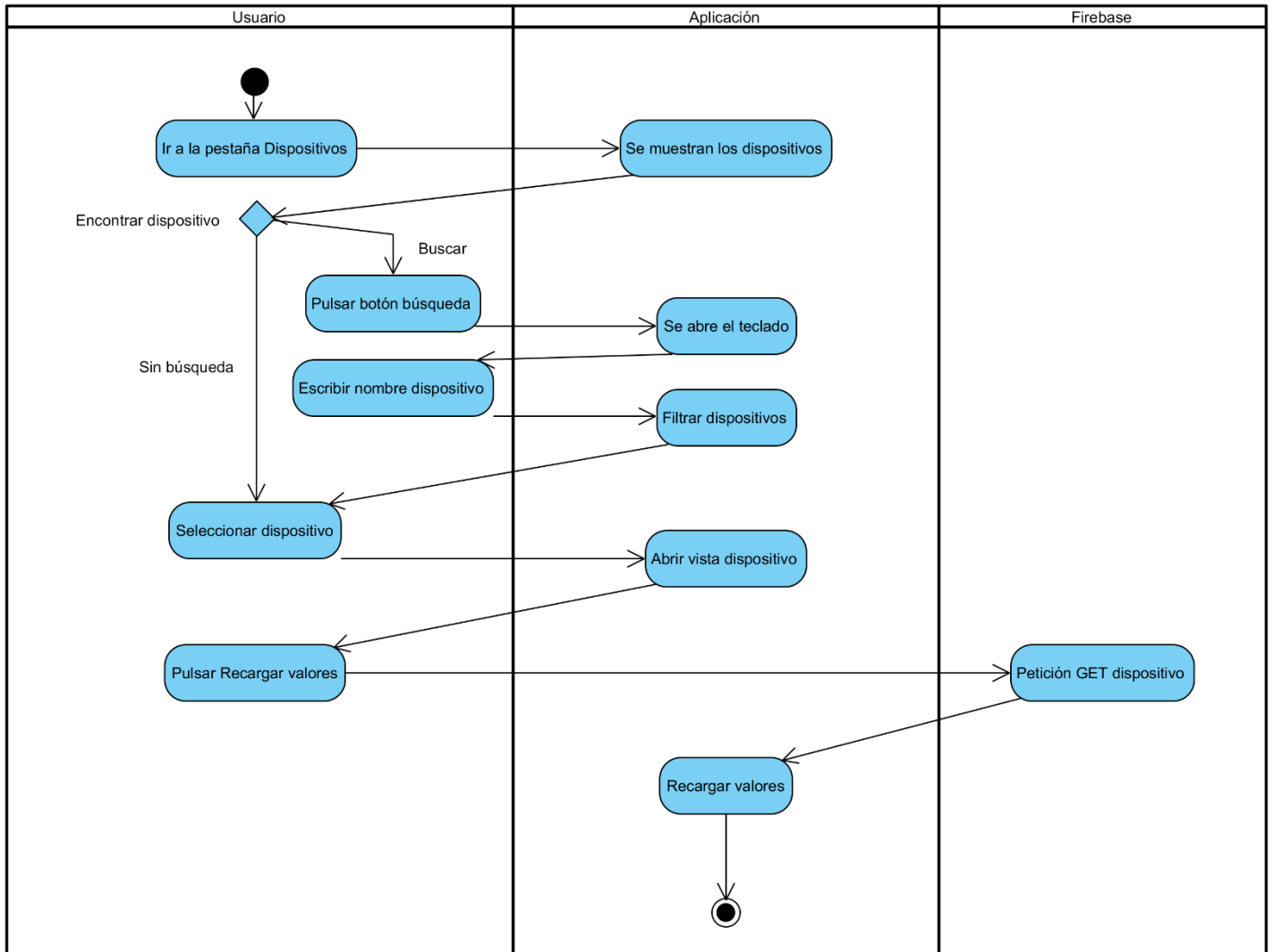


Ilustración 48. Diagrama de actividad Ver valores sensores

5.2.4.4. Modificar valor acción

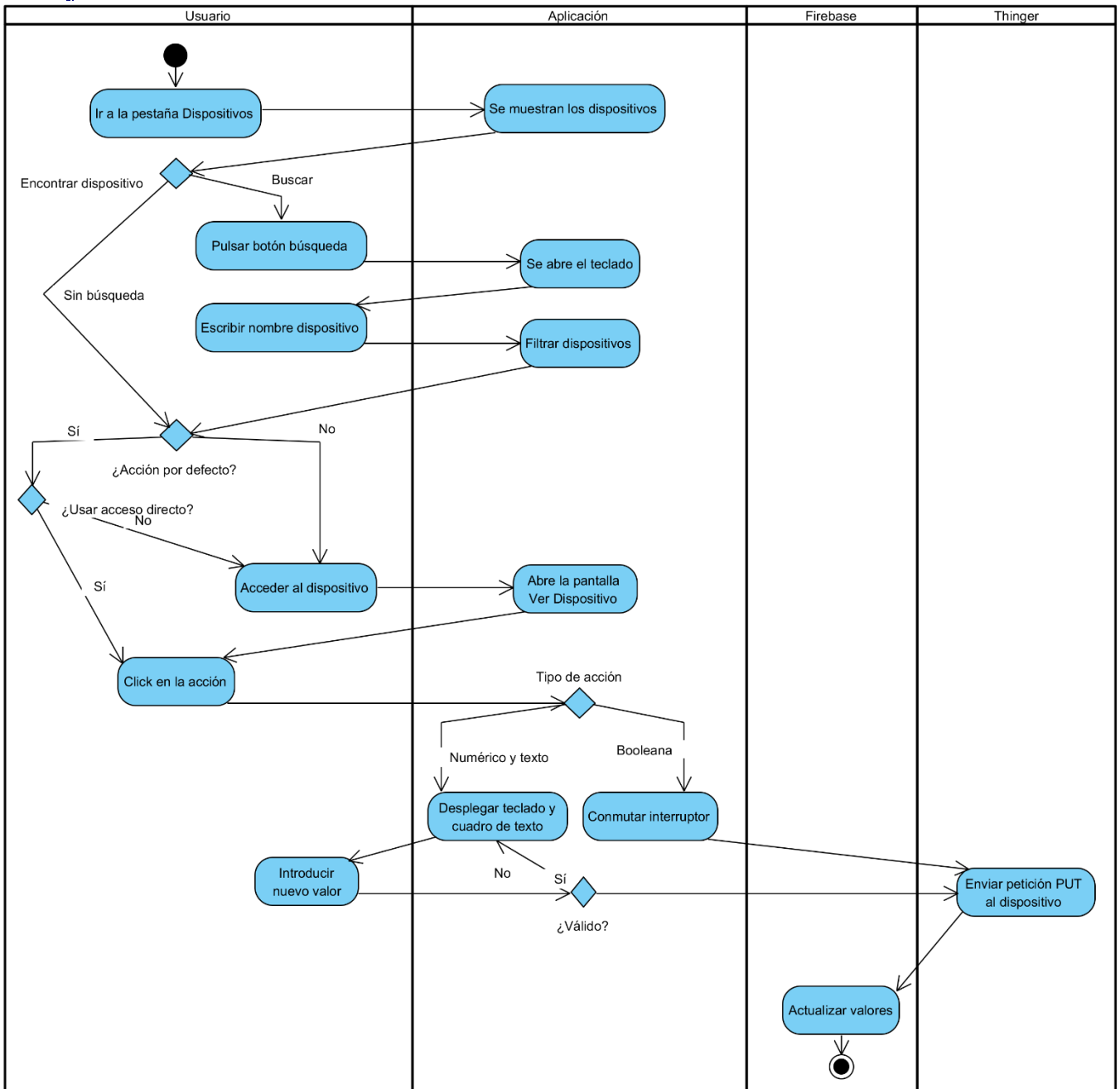


Ilustración 49. Diagrama de actividad Modificar valor acción

5.2.5. Gestionar escena

5.2.5.1. Modificar escena

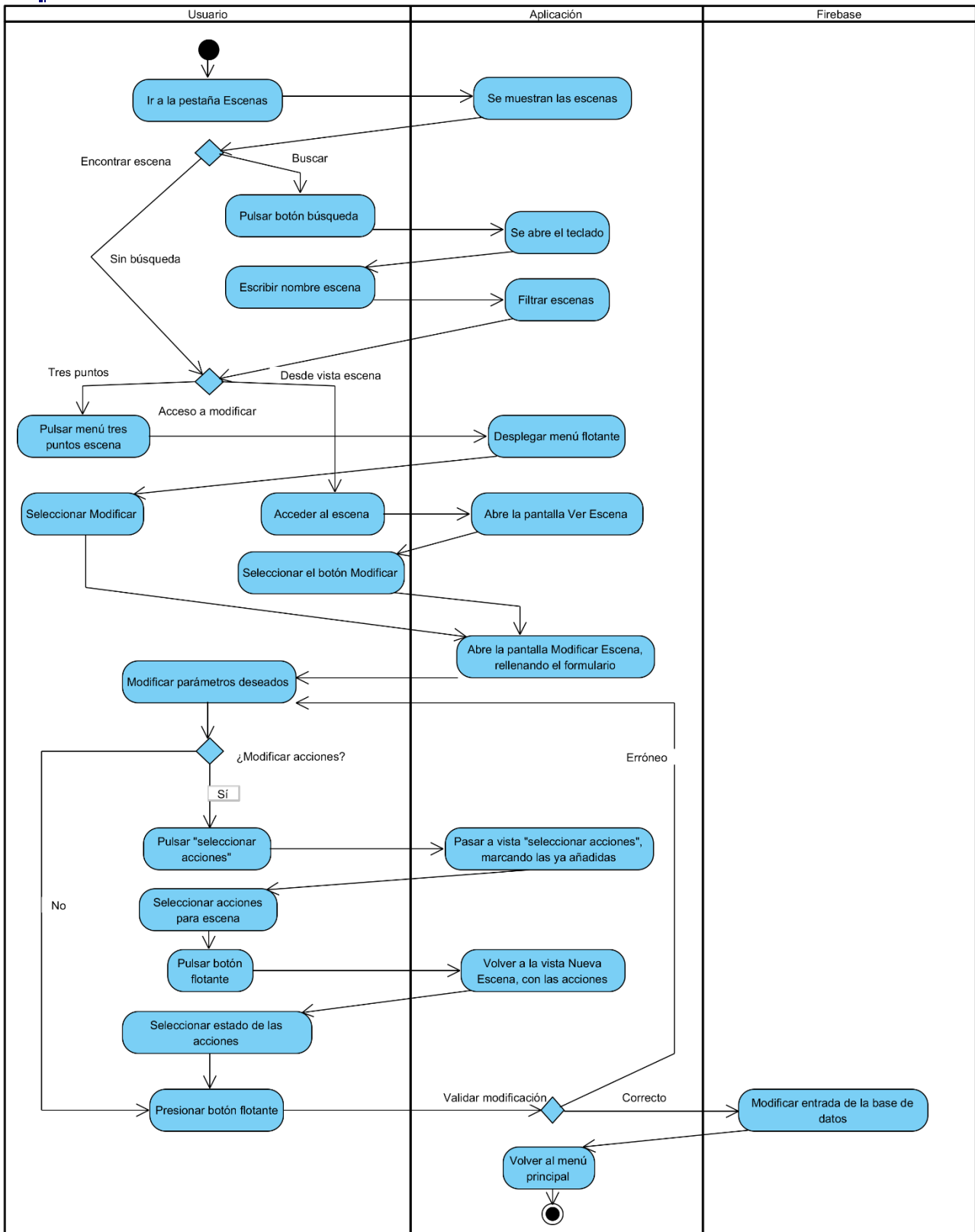


Ilustración 50. Diagrama de actividad Modificar escena

5.2.5.2. Borrar escena

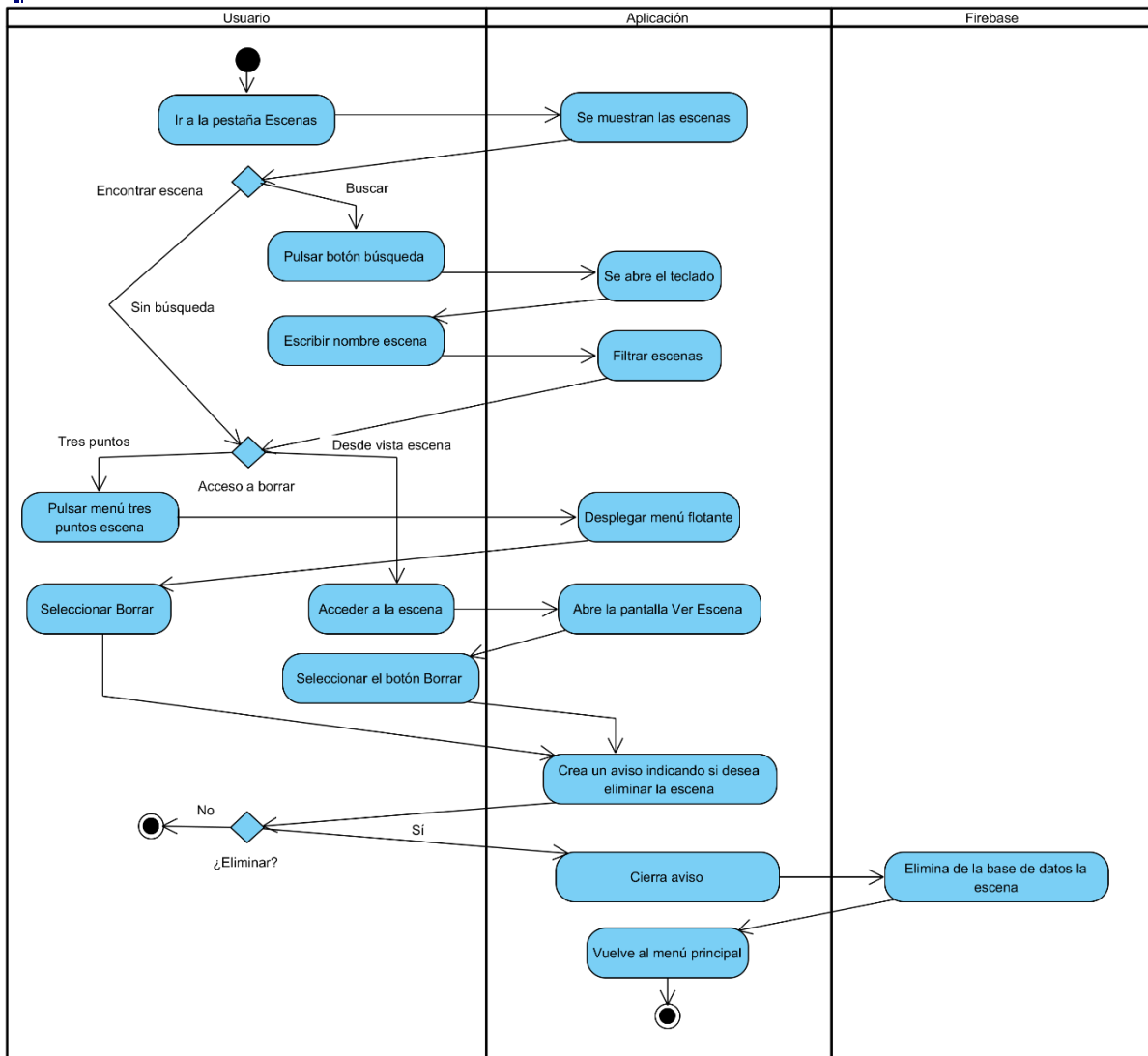


Ilustración 51. Diagrama de actividad Borrar escena

5.2.5.3. Activar / desactivar escena

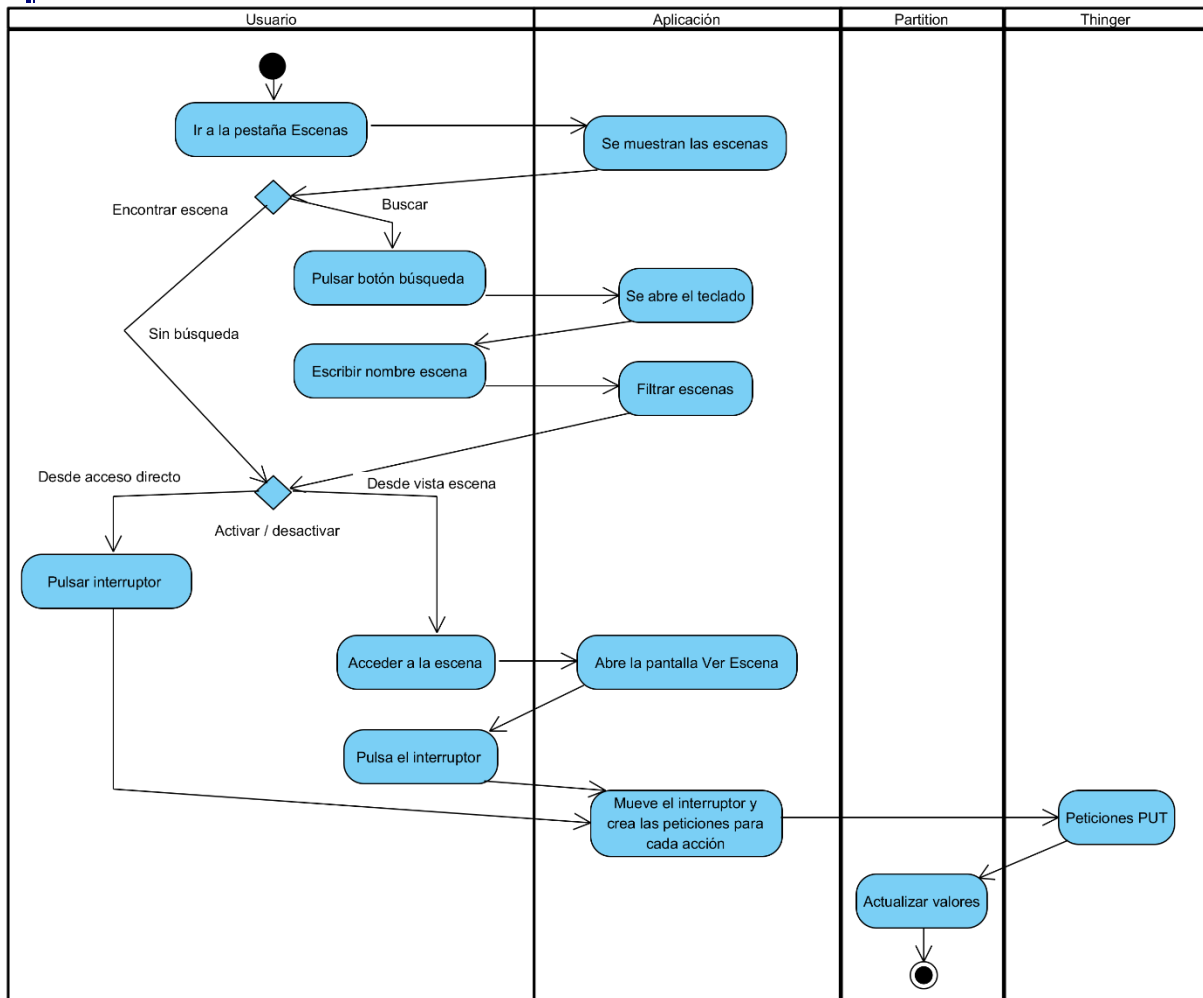


Ilustración 52. Diagrama de actividad Activar / desactivar escena

5.2.6. Gestionar localización

5.2.6.1. Modificar localización

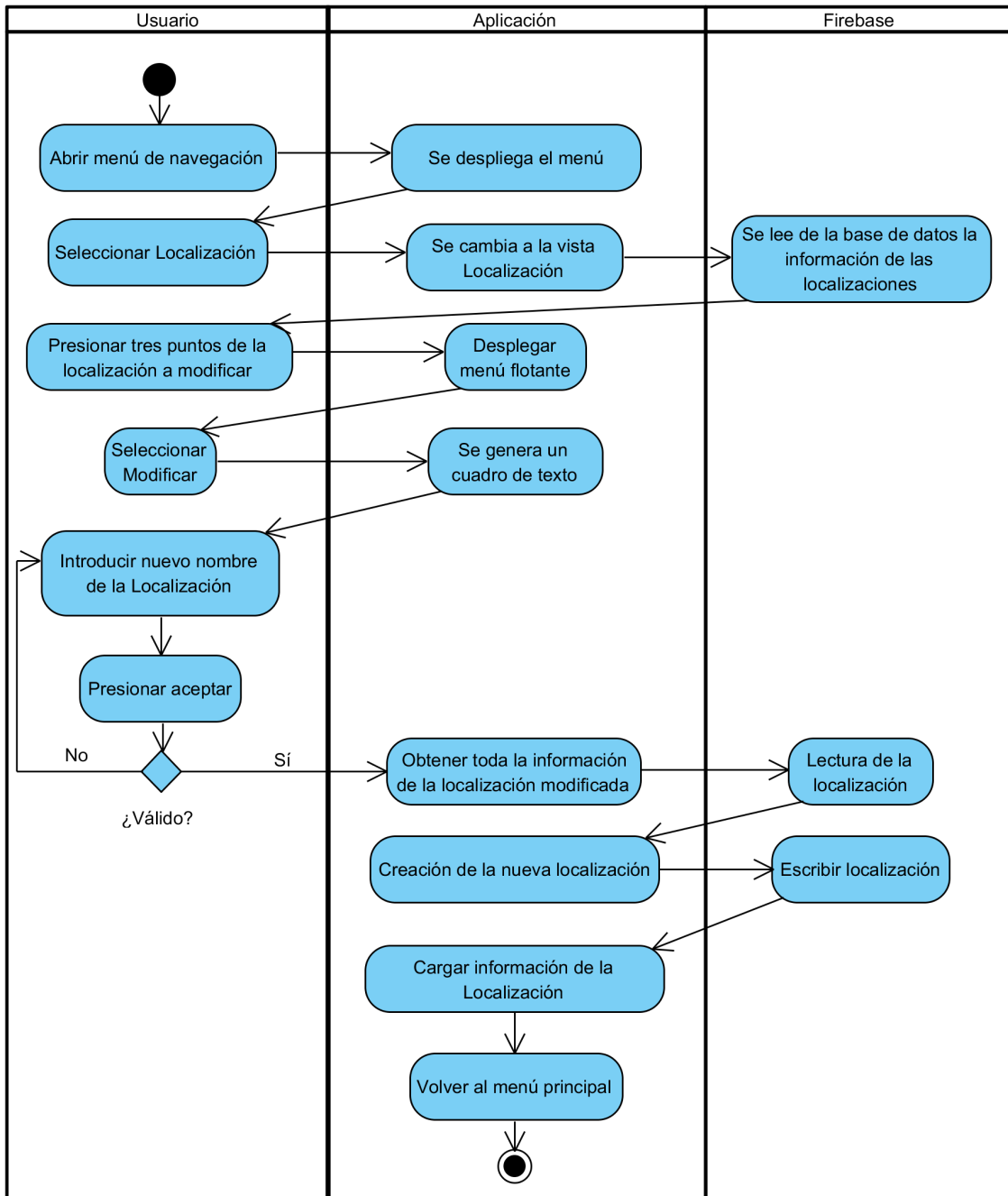


Ilustración 53. Diagrama de actividad Modificar localización

5.2.6.2. Borrar localización

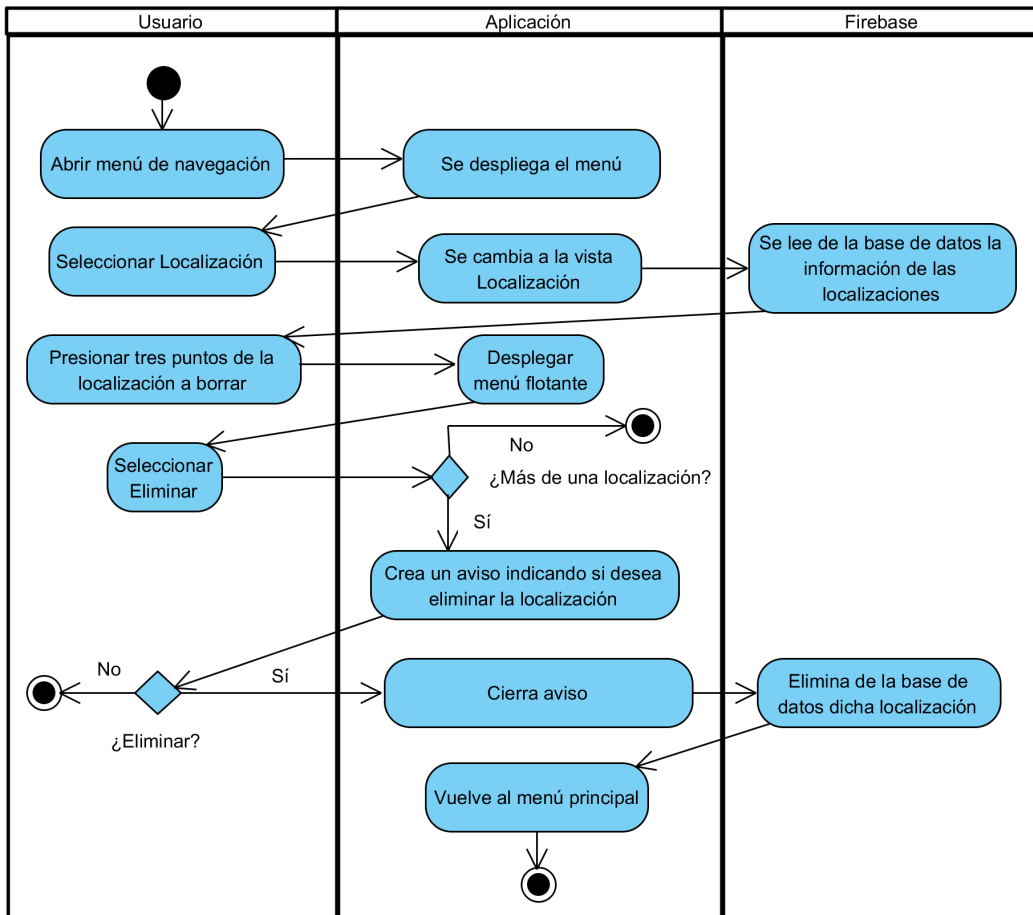


Ilustración 54. Diagrama de actividad Borrar localización

5.2.6.3. *Cambiar localización*

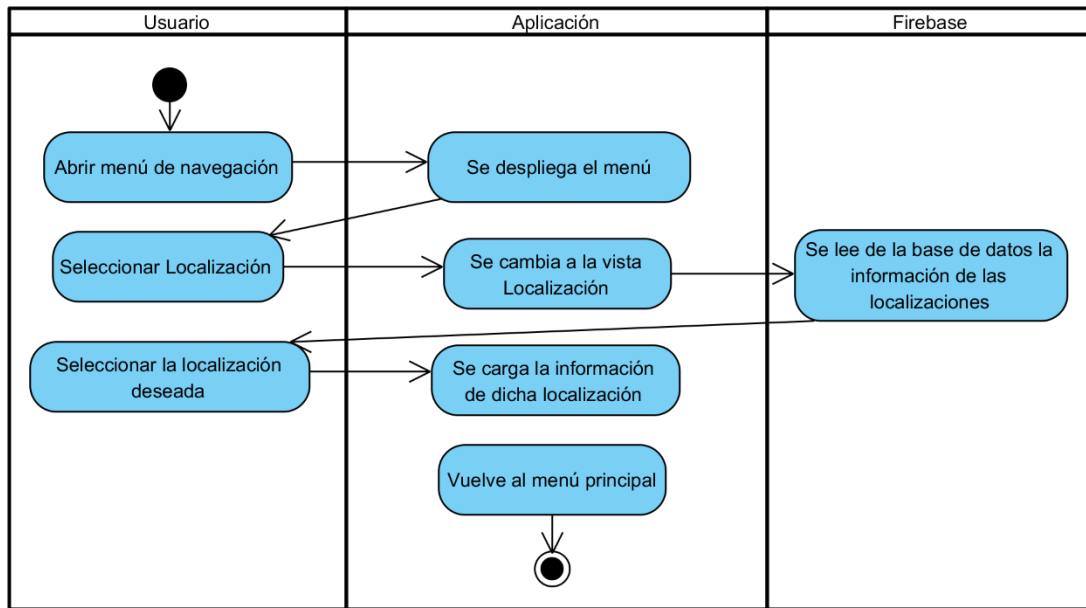


Ilustración 55. Diagrama de actividad Cambiar localización

5.2.6.4. Nueva localización

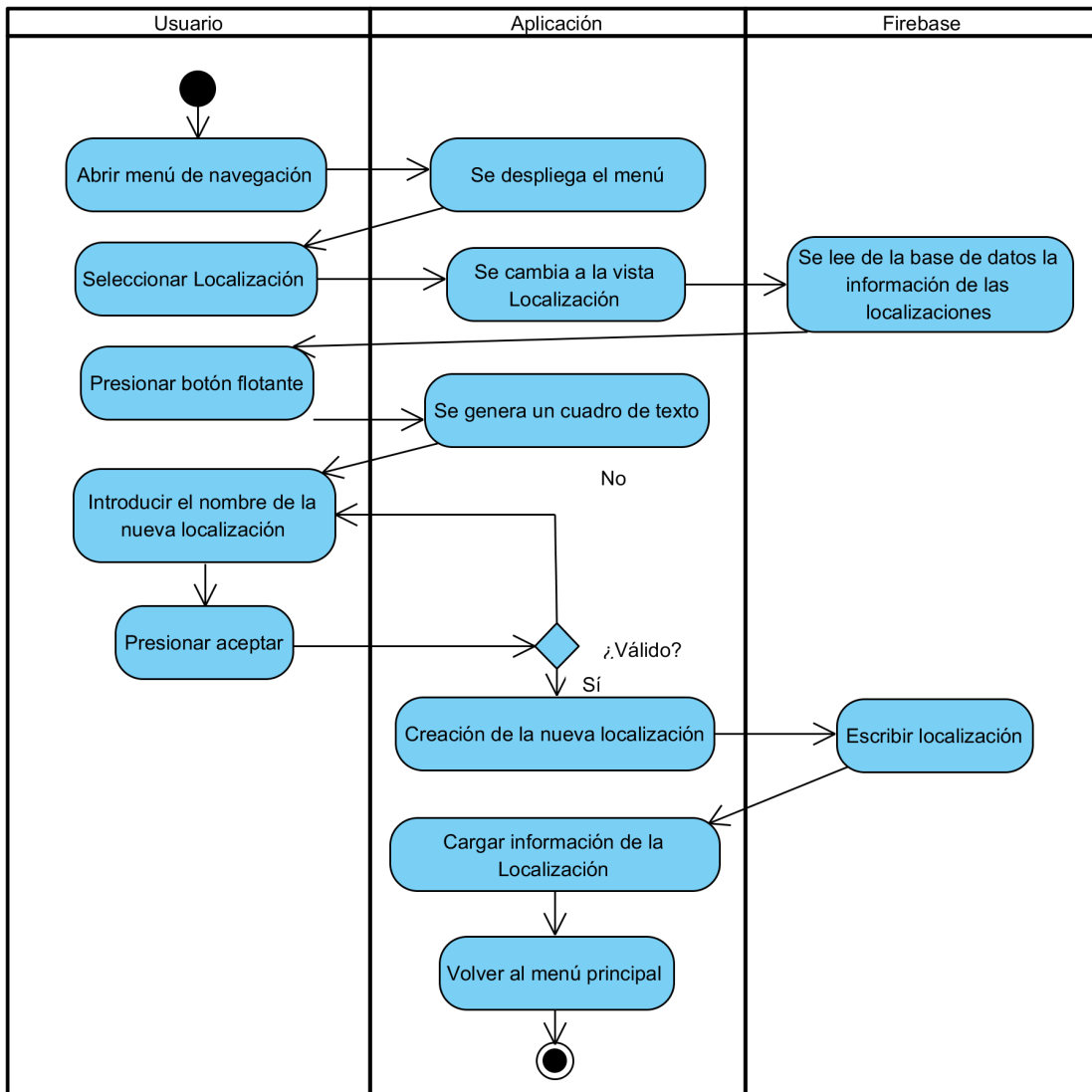


Ilustración 56. Diagrama de actividad Nueva localización

5.2.7. Gestionar perfil

5.2.7.1. Cerrar sesión

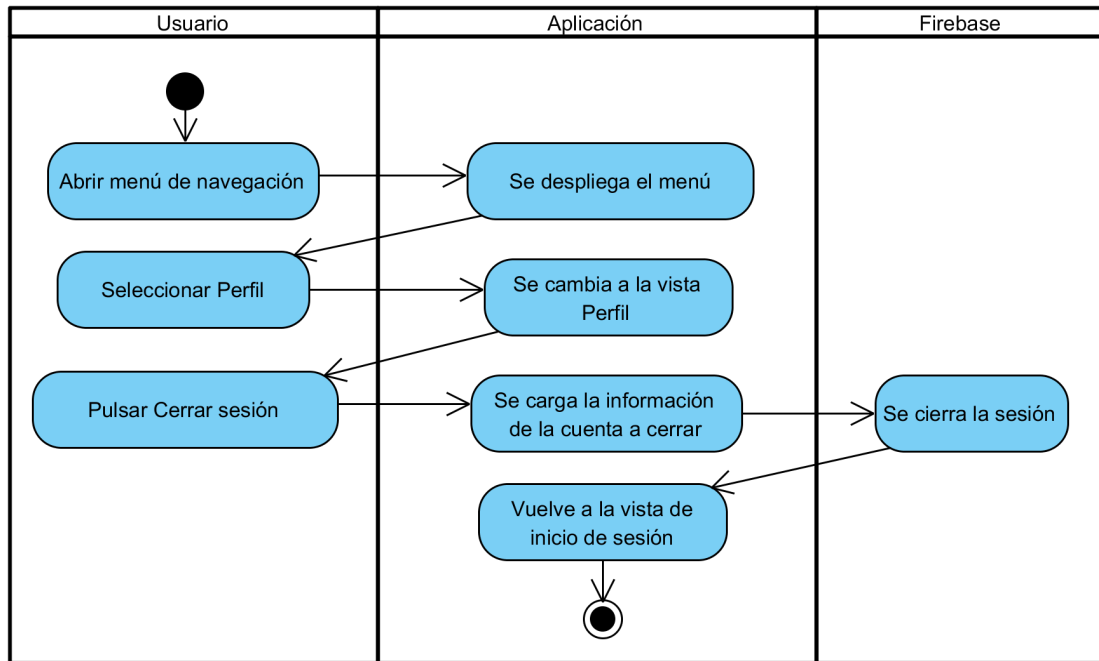


Ilustración 57. Diagrama de actividad Cerrar sesión

5.2.7.2. Revocar acceso

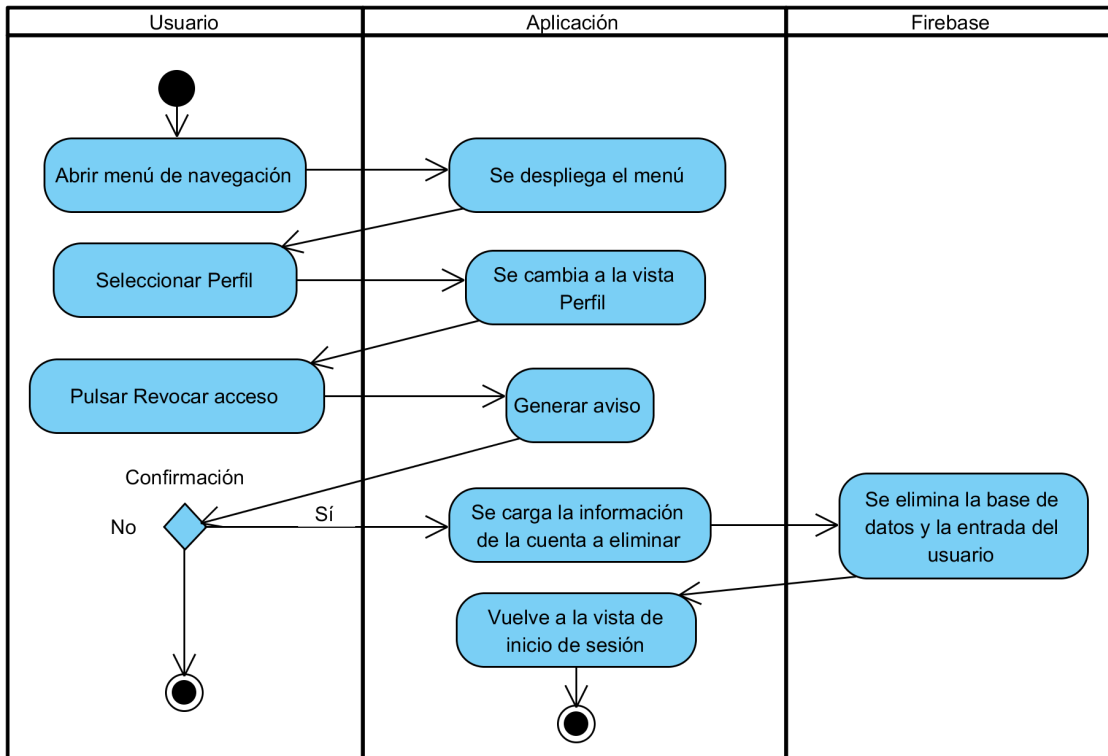


Ilustración 58. Diagrama de actividad Revocar acceso

5.2.8. Acerca de la aplicación

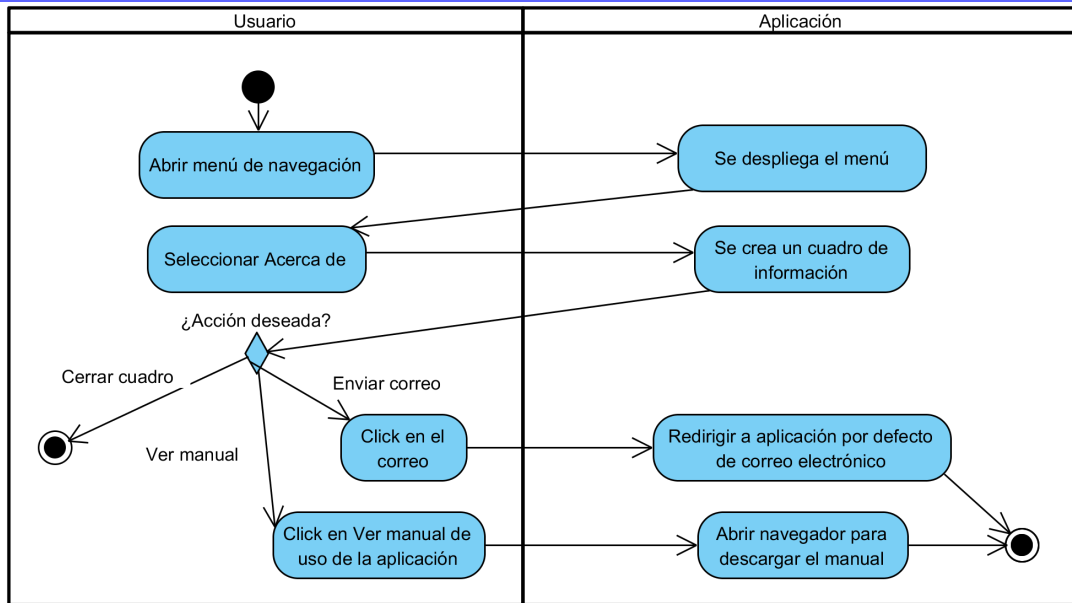


Ilustración 59. Diagrama de actividad Acerca de

5.3. Diseño de la Base de Datos

En este apartado vamos a especificar la forma en la que se almacena la información de cada usuario en la base de datos utilizada.

Esta base de datos NoSQL se estructura mediante un objeto árbol JSON, generando distintos nodos (ramas) dentro de la estructura del árbol.

Al ser una base de datos online, el paradigma utilizado para su diseño ha buscado primar el menor consumo de Internet. Para ello, se ha decidido crear una estructura de datos anidados para cada usuario.

De esta manera contamos con dos claras ventajas:

1. Controlamos a qué datos puede acceder cada usuario, mediante reglas que permiten el acceso a cada usuario únicamente a su entrada en la base de datos.
2. Facilitar la obtención de datos en la programación de la aplicación, puesto que las peticiones son asíncronas, y a más peticiones, mayor complejidad.

Destacar también que la base de datos proporcionada, permite una profundidad de 23 niveles, utilizando la aplicación únicamente nueve de los mismos, por lo que es una práctica más que válida.

A continuación, mostramos un diagrama explicativo de la estructura de la base de datos implementada:

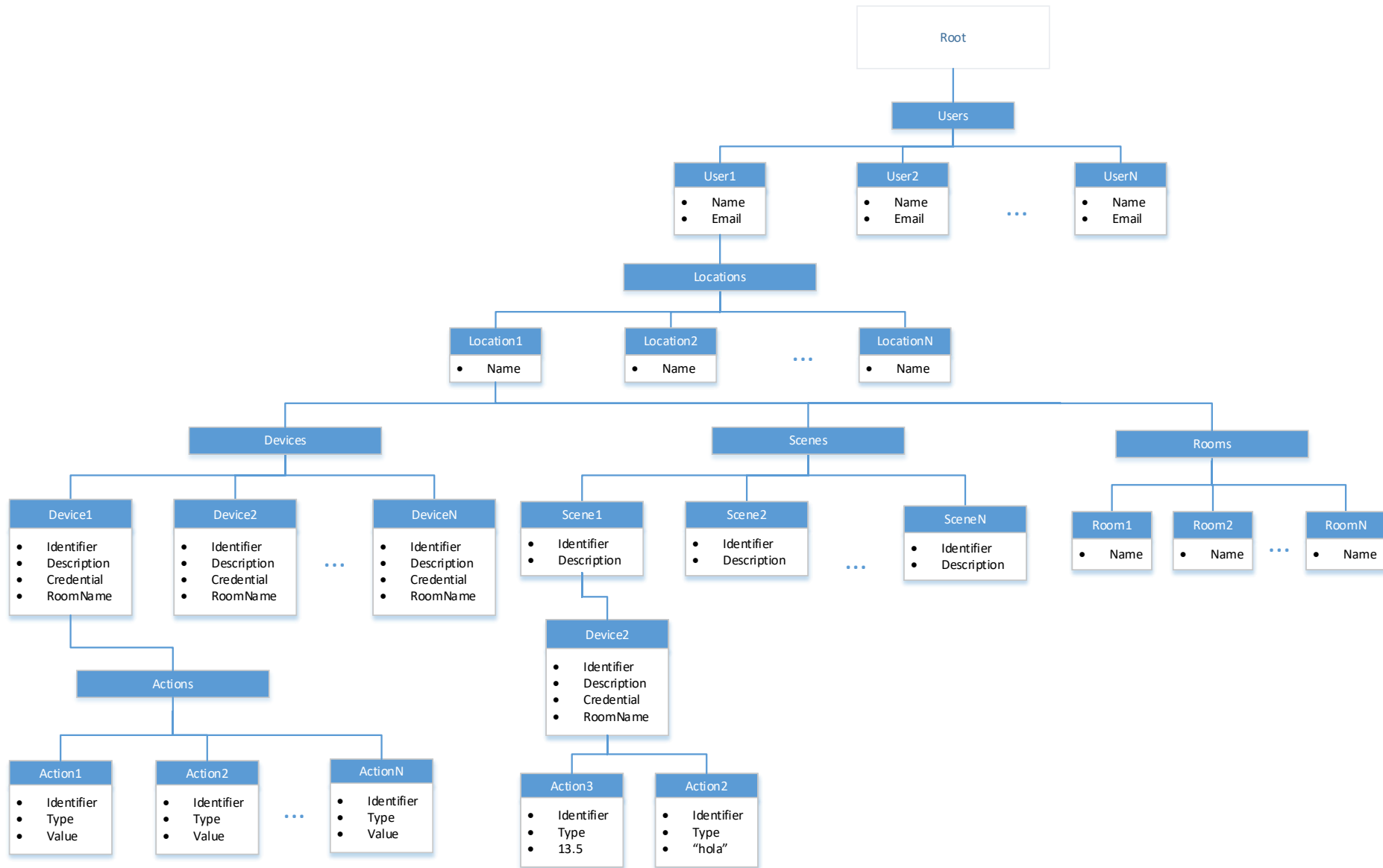


Ilustración 60. Diagrama Base de datos

6. Pruebas

En este apartado se van a detallar las pruebas a realizar por cada caso de uso. Estas pruebas, definidas por el analista, y llevadas a cabo por el programador, desvelarán la calidad del producto realizado.

Se van a realizar pruebas de varios tipos:

- De hardware: comprueba las correctas capacidades del terminal de prueba, desde una vista de usuario, sin entrar en tecnicismos del terminal.
- De flujo: comprueba el correcto funcionamiento de los diagramas de actividad realizados. Estas pruebas se realizarán siguiendo los casos de uso de la aplicación. Indicar que, cuando se produzcan decisiones por parte del usuario, se realizarán varias pruebas para contemplar el curso de cada caso.

Antes de comenzar con las pruebas, se explica la estructura de la documentación de las pruebas:

Identificador	PX.YY, siendo X el tipo de prueba. <ul style="list-style-type: none"> • H – Hardware • F – Flujo
Descripción	Se explica brevemente el funcionamiento de la prueba.
Estado inicial	Se especifica el estado inicial de las cuestiones relativas a la prueba a realizar.
Estado final	Se especifica el estado final de las cuestiones relativas a la prueba realizada.
Procedimiento de prueba	Se detallan los pasos a seguir para llevar a cabo la prueba.
Resultado de la prueba	Se indica si la prueba fue exitosa o de lo contrario, se especifica la diferencia con el resultado esperado.

Tabla 112. Tabla estructura Prueba

6.1. Hardware

Identificador	PH.01
Descripción	Prueba del funcionamiento y condiciones del sistema operativo
Estado inicial	Dispositivo móvil apagado
Estado final	Dispositivo móvil encendido
Procedimiento de prueba	<ul style="list-style-type: none"> • Presionar el botón de encendido del teléfono hasta su inicio. • Ingresar en los ajustes del sistema • Ingresar en “información del teléfono” • Observar si en la entrada “Versión de Android” cuenta con la versión 4.4 o superior.
Resultado de la prueba	Éxito

Tabla 113. Prueba PH.01

Identificador	PH.02
Descripción	Prueba de conexión a Internet
Estado inicial	Dispositivo móvil encendido con tarjeta SIM
Estado final	Dispositivo conectado a Internet
Procedimiento de prueba	<p>Vía WiFi:</p> <ul style="list-style-type: none"> • Ingresar en los ajustes del sistema • Ingresar en WiFi • Acceder a una Red Wifi <p>Vía datos móviles:</p> <ul style="list-style-type: none"> • Ingresar en los ajustes del sistema • Ingresar en Uso de datos • Activar datos
Resultado de la prueba	Éxito

Tabla 114. Prueba PH.02

6.2. Flujo

6.2.1. Inicio de sesión con cuenta de Google

Identificador	PF.01
Descripción	Prueba del flujo de inicio de sesión
Estado inicial	Dispositivo móvil en la aplicación
Estado final	Acceso al menú principal de la aplicación
Procedimiento de prueba	<ul style="list-style-type: none"> • Pulsar sobre el botón “iniciar sesión con Google” • Escoger la cuenta deseada.
Resultado de la prueba	Éxito

Tabla 115. Prueba PF.01

6.2.2. Nuevo dispositivo

6.2.2.1. Habitación ya creada, más de una acción

Identificador	PF.02
Descripción	Prueba del flujo de nuevo dispositivo
Estado inicial	Dispositivo móvil en el menú principal
Estado final	Dispositivo en el menú principal, con el dispositivo añadido
Procedimiento de prueba	<ul style="list-style-type: none"> • Pulsar sobre el botón “Nuevo dispositivo” • Rellenar el identificador, descripción y credenciales. • Seleccionar habitación • Presionar el botón flotante para confirmar • Seleccionar acción por defecto
Resultado de la prueba	Éxito

Tabla 116. Prueba PF.02

6.2.2.2. Crear habitación, una sola acción

Identificador	PF.03
Descripción	Prueba del flujo de nuevo dispositivo
Estado inicial	Dispositivo móvil en el menú principal
Estado final	Dispositivo en el menú principal, con el dispositivo añadido
Procedimiento de prueba	<ul style="list-style-type: none"> • Pulsar sobre el botón “Nuevo dispositivo” • Rellenar el identificador, descripción y credenciales. • Seleccionar “Nueva habitación” • Insertar nombre habitación y aceptar • Presionar el botón flotante para confirmar
Resultado de la prueba	Éxito

Tabla 117. Prueba PF.03

6.2.3. Nueva escena

Identificador	PF.04
Descripción	Prueba del flujo de nueva escena
Estado inicial	Dispositivo móvil en el menú principal
Estado final	Dispositivo en el menú principal, con una escena añadida
Procedimiento de prueba	<ul style="list-style-type: none"> • Pulsar sobre el botón “Nueva escena” • Rellenar el identificador, descripción. • Pulsar el botón “Seleccionar acciones” • Presionar las acciones. • Presionar el botón flotante para confirmar acciones. • Establecer estado de cada acción. • Presionar el botón flotante para confirmar escena.
Resultado de la prueba	Éxito

Tabla 118. Prueba PF.04

6.2.4. Gestionar dispositivo

6.2.4.1. Modificar dispositivo

6.2.4.1.1. Búsqueda, vía tres puntos

Identificador	PF.05
Descripción	Prueba del flujo de nueva escena
Estado inicial	En el menú principal, con dispositivos añadidos
Estado final	En el menú principal, con el dispositivo modificado
Procedimiento de prueba	<ul style="list-style-type: none"> • Ir a la pestaña Dispositivos • Pulsar el botón de búsqueda • Insertar el nombre del dispositivo • Seleccionar el menú de tres puntos • Seleccionar Modificar • Modificar los parámetros deseados • Presionar el botón flotante
Resultado de la prueba	Éxito

Tabla 119. Prueba PF.05

6.2.4.1.2. Sin Búsqueda, vía vista dispositivo

Identificador	PF.06
Descripción	Prueba del flujo de nueva escena
Estado inicial	En el menú principal, con dispositivos añadidos
Estado final	En el menú principal, con el dispositivo modificado
Procedimiento de prueba	<ul style="list-style-type: none"> • Ir a la pestaña Dispositivos • Acceder al dispositivo • Seleccionar el botón Modificar • Modificar los parámetros deseados • Presionar el botón flotante
Resultado de la prueba	Éxito

Tabla 120. Prueba PF.06

6.2.4.2. Borrar dispositivo

6.2.4.2.1. Con búsqueda, vía tres puntos

Identificador	PF.07
Descripción	Prueba del flujo de nueva escena
Estado inicial	En el menú principal, con dispositivos añadidos
Estado final	En el menú principal, sin el dispositivo eliminado
Procedimiento de prueba	<ul style="list-style-type: none"> • Ir a la pestaña Dispositivos • Pulsar el botón de búsqueda • Insertar el nombre del dispositivo • Seleccionar el menú de tres puntos • Seleccionar Borrar • Pulsar Eliminar
Resultado de la prueba	Éxito

Tabla 121. Prueba PF.07

6.2.4.2.2. Sin búsqueda, vía vista dispositivo

Identificador	PF.08
Descripción	Prueba del flujo de nueva escena
Estado inicial	En el menú principal, con dispositivos añadidos
Estado final	En el menú principal, sin el dispositivo eliminado
Procedimiento de prueba	<ul style="list-style-type: none"> • Ir a la pestaña Dispositivos • Acceder al dispositivo • Seleccionar el botón Borrar • Pulsar Eliminar
Resultado de la prueba	Éxito

Tabla 122. Prueba PF.08

6.2.4.3. Ver valores sensores

Identificador	PF.09
Descripción	Prueba del flujo de gestionar dispositivo. Ver valores sensores
Estado inicial	En el menú principal, con dispositivos añadidos
Estado final	En la vista de dispositivos, con valores de sensores
Procedimiento de prueba	<ul style="list-style-type: none"> • Ir a la pestaña Dispositivos • Acceder al dispositivo • Pulsar Recargar Valores
Resultado de la prueba	Éxito

Tabla 123. Prueba PF.09

6.2.4.4. Modificar valor acción

6.2.4.4.1. Valor booleano, desde el acceso directo, con búsqueda

Identificador	PF.10
Descripción	Prueba del flujo de gestionar dispositivo. Modificar valor
Estado inicial	En el menú principal, con un dispositivo añadido con acción booleana por defecto
Estado final	En la vista de dispositivos, con la acción en otro estado
Procedimiento de prueba	<ul style="list-style-type: none"> • Ir a la pestaña Dispositivos • Pulsar el botón de búsqueda • Insertar el nombre del dispositivo • Pulsar el interruptor
Resultado de la prueba	Éxito

Tabla 124. Prueba PF.10

6.2.4.4.2. Valor texto, desde el dispositivo, sin búsqueda

Identificador	PF.11
Descripción	Prueba del flujo de gestionar dispositivo. Modificar valor
Estado inicial	En el menú principal, con un dispositivo añadido con acción texto
Estado final	En la vista de dispositivos, con la acción en otro estado
Procedimiento de prueba	<ul style="list-style-type: none"> • Ir a la pestaña Dispositivos • Acceder al dispositivo • Seleccionar acción • Introducir valor
Resultado de la prueba	Éxito

Tabla 125. Prueba PF.11

6.2.5. Gestionar escena

6.2.5.1. Modificar escena

6.2.5.1.1. Con búsqueda, vía tres puntos, sin modificar acciones

Identificador	PF.12
Descripción	Prueba del flujo de gestionar escena. Modificar escena
Estado inicial	En el menú principal, con una escena añadida.
Estado final	En el menú principal, con la escena modificada.
Procedimiento de prueba	<ul style="list-style-type: none"> • Ir a la pestaña Escenas • Pulsar botón búsqueda • Escribir nombre escena • Pulsar menú tres puntos de la escena • Seleccionar modificar • Modificar el identificador • Presionar el botón flotante
Resultado de la prueba	Éxito

Tabla 126. Prueba PF.12

6.2.5.1.2. Sin búsqueda, vía vista escena, modificando acciones

Identificador	PF.13
Descripción	Prueba del flujo de gestionar escena. Modificar escena
Estado inicial	En el menú principal, con una escena añadida.
Estado final	En el menú principal, con la escena modificada.
Procedimiento de prueba	<ul style="list-style-type: none"> • Ir a la pestaña Escenas • Acceder a la escena • Seleccionar botón Modificar • Modificar el identificador • Pulsar Seleccionar acciones • Seleccionar otra acción • Pulsar el botón flotante para volver • Ajustar el estado de la nueva acción de la escena • Presionar el botón flotante
Resultado de la prueba	Éxito

Tabla 127. Prueba PF.13

6.2.5.2. *Borrar escena*

6.2.5.2.1. *Con búsqueda, vía tres puntos*

Identificador	PF.14
Descripción	Prueba del flujo de gestionar escena. Borrar escena
Estado inicial	En el menú principal, con una escena añadida.
Estado final	En el menú principal, sin dicha escena.
Procedimiento de prueba	<ul style="list-style-type: none"> • Ir a la pestaña Escenas • Pulsar botón búsqueda • Escribir nombre escena • Pulsar menú tres puntos de la escena • Seleccionar borrar • Seleccionar Sí, eliminar
Resultado de la prueba	Éxito

Tabla 128. Prueba PF.14

6.2.5.2.2. *Sin búsqueda, vía vista escena*

Identificador	PF.15
Descripción	Prueba del flujo de gestionar escena. Modificar escena
Estado inicial	En el menú principal, con una escena añadida.
Estado final	En la vista de escenas, sin dicha escena.
Procedimiento de prueba	<ul style="list-style-type: none"> • Ir a la pestaña Escenas • Acceder a la escena • Seleccionar botón Borrar • Seleccionar Sí, eliminar
Resultado de la prueba	Éxito

Tabla 129. Prueba PF.15

6.2.5.3. *Activar / desactivar escena*

6.2.5.3.1. *Con búsqueda, desde acceso directo*

Identificador	PF.16
Descripción	Prueba del flujo de gestionar escena. Borrar escena
Estado inicial	En el menú principal, con una escena añadida.
Estado final	En el menú principal, con la escena activada.
Procedimiento de prueba	<ul style="list-style-type: none"> • Ir a la pestaña Escenas • Pulsar botón búsqueda • Escribir nombre escena • Pulsar el interruptor de la escena
Resultado de la prueba	Éxito

Tabla 130. Prueba PF.16

6.2.5.3.2. Sin búsqueda, vía vista escena

Identificador	PF.17
Descripción	Prueba del flujo de gestionar escena. Borrar escena
Estado inicial	En el menú principal, con una escena añadida.
Estado final	En la vista de la escena, con la misma activa.
Procedimiento de prueba	<ul style="list-style-type: none"> • Ir a la pestaña Escenas • Acceder a la escena • Pulsar el interruptor de la escena
Resultado de la prueba	Éxito

Tabla 131. Prueba PF.17

6.2.6. Gestionar localización

6.2.6.1. Modificar localización

Identificador	PF.18
Descripción	Prueba del flujo de gestionar localización. Modificar localización
Estado inicial	En el menú principal.
Estado final	En el menú principal, en la localización modificada.
Procedimiento de prueba	<ul style="list-style-type: none"> • Abrir el menú de navegación • Seleccionar Localización • Presionar menú tres puntos de la localización • Seleccionar Modificar • Introducir el nuevo nombre de la localización • Presionar Aceptar
Resultado de la prueba	Éxito

Tabla 132. Prueba PF.18

6.2.6.2. Borrar localización

Identificador	PF.19
Descripción	Prueba del flujo de gestionar localización. Borrar localización
Estado inicial	En el menú principal.
Estado final	En la vista de localizaciones, sin dicha localización.
Procedimiento de prueba	<ul style="list-style-type: none"> • Abrir el menú de navegación • Seleccionar Localización • Presionar menú tres puntos de la localización • Seleccionar Eliminar • Confirmar eliminar
Resultado de la prueba	Éxito

Tabla 133. Prueba PF.19

6.2.6.3. Cambiar localización

Identificador	PF.20
Descripción	Prueba del flujo de gestionar localización. Cambiar localización
Estado inicial	En el menú principal.
Estado final	En el menú principal, en la localización seleccionada.
Procedimiento de prueba	<ul style="list-style-type: none"> • Abrir el menú de navegación • Seleccionar Localización • Seleccionar la localización deseada
Resultado de la prueba	Éxito

Tabla 134. Prueba PF.20

6.2.6.4. Nueva localización

Identificador	PF.21
Descripción	Prueba del flujo de gestionar localización. Nueva localización
Estado inicial	En el menú principal.
Estado final	En el menú principal, en la nueva localización.
Procedimiento de prueba	<ul style="list-style-type: none"> • Abrir el menú de navegación • Seleccionar Localización • Presionar el botón flotante • Escribir el nombre de la localización • Presionar Aceptar
Resultado de la prueba	Éxito

Tabla 135. Prueba PF.21

6.2.7. Gestionar perfil

6.2.7.1. Cerrar sesión

Identificador	PF.22
Descripción	Prueba del flujo de gestionar perfil. Cerrar sesión
Estado inicial	En el menú principal.
Estado final	En la vista de inicio de sesión.
Procedimiento de prueba	<ul style="list-style-type: none"> • Abrir el menú de navegación • Seleccionar Perfil • Presionar el botón Cerrar sesión
Resultado de la prueba	Éxito

Tabla 136. Prueba PF.22

6.2.7.2. Revocar acceso

Identificador	PF.23
Descripción	Prueba del flujo de gestionar perfil. Revocar acceso
Estado inicial	En el menú principal.
Estado final	En la vista de inicio de sesión.
Procedimiento de prueba	<ul style="list-style-type: none"> • Abrir el menú de navegación • Seleccionar Perfil • Presionar el botón Revocar acceso • Confirmar
Resultado de la prueba	Éxito

Tabla 137. Prueba PF.23

6.2.8. Acerca de

Identificador	PF.24
Descripción	Prueba del flujo de Acerca de
Estado inicial	En el menú principal.
Estado final	En el menú principal con el cuadro de texto desplegado.
Procedimiento de prueba	<ul style="list-style-type: none"> • Abrir el menú de navegación • Seleccionar Acerca de
Resultado de la prueba	Éxito

Tabla 138. Prueba PF.24

Identificador	PF.25
Descripción	Prueba del flujo de Acerca de. Enviar correo
Estado inicial	En el menú principal.
Estado final	En la aplicación por defecto de correo electrónico
Procedimiento de prueba	<ul style="list-style-type: none"> • Abrir el menú de navegación • Seleccionar Acerca de • Click en el correo
Resultado de la prueba	Éxito

Tabla 139. Prueba PF.25

Identificador	PF.26
Descripción	Prueba del flujo de Acerca de. Ver manual
Estado inicial	En el menú principal.
Estado final	En la aplicación por defecto de navegador
Procedimiento de prueba	<ul style="list-style-type: none"> • Abrir el menú de navegación • Seleccionar Acerca de • Click en el Ver manual de uso de la aplicación
Resultado de la prueba	Éxito

Tabla 140. Prueba PF.26

7. Conclusiones y líneas futuras

7.1. Conclusiones

En este apartado se ahondará sobre las conclusiones a las que se han llegado tras la finalización del proyecto. Se comenzará con las conclusiones a nivel técnico sobre el proyecto, y posteriormente las conclusiones del único miembro de desarrollo.

7.1.1. Conclusiones a nivel de proyecto

A lo largo del desarrollo se han ido cubriendo los distintos objetivos fijados al comienzo del proyecto, los cuales repasamos para valorarlos finalmente:

- ✓ Se pretendía analizar los sistemas domóticos existentes en la actualidad. Este apartado queda más que cubierto con un análisis extenso, haciendo hincapié en las funcionalidades que ofrece cada uno de los sistemas actuales, para encontrar sus puntos débiles y carencias, para orientar el proyecto a cubrir dichas necesidades, como quedó claro en el estado del arte.
- ✓ Se pretendía analizar las necesidades que debe tener una herramienta de gestión domótica actual. Este apartado se solventó tras realizar los requisitos de usuario del apartado 4.1.2, donde el analista del proyecto definió las necesidades de la aplicación, validados por una selección de posibles usuarios mediante el wireframing realizado, y culminado con los posteriores casos de uso.
- ✓ Se pretendía diseñar la arquitectura del sistema a desarrollar. Este objetivo se solventó gracias al diseño de la aplicación realizado por el analista, en el que definió la arquitectura a implementar para el correcto funcionamiento de la aplicación. Además, gracias a los diagramas de actividad, se detalló de manera clara y sencilla el funcionamiento de cada una de las funcionalidades a realizar por el usuario, facilitando la posterior implementación.
- ✓ Y, por último, se pretendía desarrollar la aplicación. Esta ardua tarea la completó el programador únicamente, con un esfuerzo impagable y una calidad final más que digna, cumpliendo perfectamente todas las pruebas definidas por el analista.

7.1.2. Conclusiones a nivel personal

Personalmente, este proyecto me ha hecho valorar más la cantidad de aplicaciones que consumimos día a día, y que nos hacen la vida tan y tan fácil. Es una pena que apenas se valore el trabajo arduo que hay detrás de ellas.

En cuanto al proyecto en sí, ha sido satisfactorio ver cómo iba creciendo y creciendo una vez se comenzó con el desarrollo, ver como todo el esfuerzo de un mes, abstracto, del cual no tienes referencias ni garantías de que vaya funcionar, acabe funcionando y aligerando el trabajo que sin una buena planificación hubiera sido imposible.

Este ha sido mi primer proyecto Android serio, y ha sido un reto personal muy satisfactorio como he indicado antes. Cada vez que se consigue implementar una funcionalidad nueva, que después de días de trabajo te percatas de un error ínfimo que se cargaba toda la ejecución...

El saber que esta aplicación es útil, que alguien si tiene estas necesidades puede disponer de ella y utilizarla sin ningún tipo de problema ni modificación, es reconfortante.

Por último, quizá lo más importante, las capacidades de las que dispongo a día de hoy son infinitamente superiores a las del día 1 del proyecto. Desarrollo de una aplicación Android desde cero, lleva una planificación al día, generar una cantidad inmensa de diagramas y requisitos totalmente coherentes, y todo por mí mismo, sin ayuda y todo bien hecho, me parece que el trabajo duro ha dado sus frutos.

7.2. Líneas futuras

La aplicación desarrollada cumple con creces una función muy concreta, para la cual fue diseñada, pero siempre puede ser mejorada e implementar sobre ella mejoras que otorguen una mayor funcionalidad a la misma.

Con el desarrollo realizado, surgieron diversas ideas que no terminaron implementándose, ya sea por falta de tiempo o por simples incapacidades técnicas para llevarse a cabo. Algunas de ellas son:

- Programar acciones, o conjuntos de acciones a realizar en un momento concreto, ya sea bajo una necesidad temporal, como, por ejemplo, encender las luces a las 12 de la noche; o gracias a los parámetros de los dispositivos, como, por ejemplo, bajar las persianas si la temperatura sube de 30 grados.
- Permitir acceso a terceros controlar los dispositivos domóticos. Al fin y al cabo, es general vivir en compañía y tener que vincular los dispositivos cada persona con su usuario no es la mejor opción. Esta opción además podría extenderse

bastante, permitiendo variar el grado de uso de los dispositivos en ciertas condiciones, notificaciones del uso al propietario de la cuenta, contar con un historial para observar el uso, etc.

- Implementar una funcionalidad de ajustes, dando personalización a la aplicación. Modo nocturno, sistemas de accesibilidad, control sobre el tamaño de las fuentes y los colores...
- Desarrollar la aplicación en otros sistemas, por ejemplo, de acceso vía web.
- Permitir el acceso de usuario mediante sistemas de seguridad, como huella dactilar, escáner de iris, etc.

8. Anexo A. Manual de uso

En este primer anexo se va a explicar el funcionamiento de la aplicación, desde su instalación en el dispositivo Android, crear los dispositivos Arduino para utilizar en la aplicación, y por último cómo utilizar las funcionalidades de la aplicación.

8.1. Instalación de la aplicación

Para instalar la aplicación, simplemente hay que acceder a la tienda de aplicaciones de Google, la aplicación Google Play Store, y buscar “domótica Grupo Inteligencia Artificial Aplicada”, siendo la siguiente aplicación:

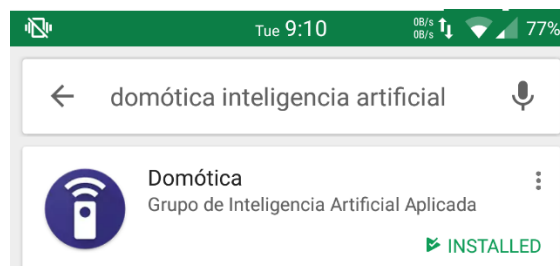


Ilustración 61. Instalación de la aplicación

También se puede acceder directamente por el siguiente enlace:

<https://play.google.com/store/apps/details?id=com.giaa.domotic&hl=es>

8.2. Creación de dispositivos Arduino

Para crear un dispositivo a utilizar en la aplicación, es necesario que conste de los ajustes indicados en la página del sistema Thingier. A continuación, explicamos paso a paso como dejar funcional un NodeMCU para su funcionamiento en la aplicación, utilizando el IDE Arduino.

8.2.1. Descargar librerías

Antes de comenzar con el código del dispositivo, es necesario dejar la librería de Thingier preparada. Para ello, es necesario ir al gestor de librerías de Arduino. Para ello, hay que seguir los siguientes pasos:

Programa → Incluir librería → Gestionar librerías → escribir Thingier.io → descargar la última versión (última utilizada en el proyecto: 2.7.1).

A continuación, hay que conectar por Micro-USB el dispositivo NodeMCU al ordenador.

Una vez conectada, ir a Herramientas → Gestor de tarjetas → escribir esp8266 e instalarlo. Esto descarga todas las librerías necesarias para utilizar el dispositivo.

8.2.2. Código Arduino

Una vez se tiene el entorno preparado, hay que introducir el código al Arduino. A continuación, se explica el código:

8.2.2.1. Includes y variables globales

Es necesario importar en el fichero las siguientes librerías:

```
#include <ESP8266WiFi.h>           // WiFi
#include <ThingyESP8266.h>         // Thingy
const char* ssid = "";           // Nombre red WiFi
const char* password = "";       // Contraseña red WiFi
#define USERNAME "gaaaUC3M"      // Imprescindible esta cuenta de Thingy
#define DEVICE_ID ""             // Nombre del dispositivo
#define DEVICE_CREDENTIAL ""     // Credenciales del dispositivo
ThingyESP8266 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
```

8.2.2.2. Setup

Una vez generado el objeto de Thingy, y la conexión WiFi, generamos el código inicial setup():

```
void setup() {
    thing.add_wifi(ssid, password);           // Inicializa el WiFi
    pinMode(BUILTIN_LED, OUTPUT);           // Inicializa el LED
    thing["led"] << digitalPin(BUILTIN_LED); // Introduce la primera acción
    // Introducir las acciones deseadas...
}
```

Se introduce la acción led por defecto, puesto que es la más sencilla para comprobar el funcionamiento de la aplicación.

8.2.2.3. Loop

Una vez inicializado el sistema, únicamente hay que introducir la siguiente línea:

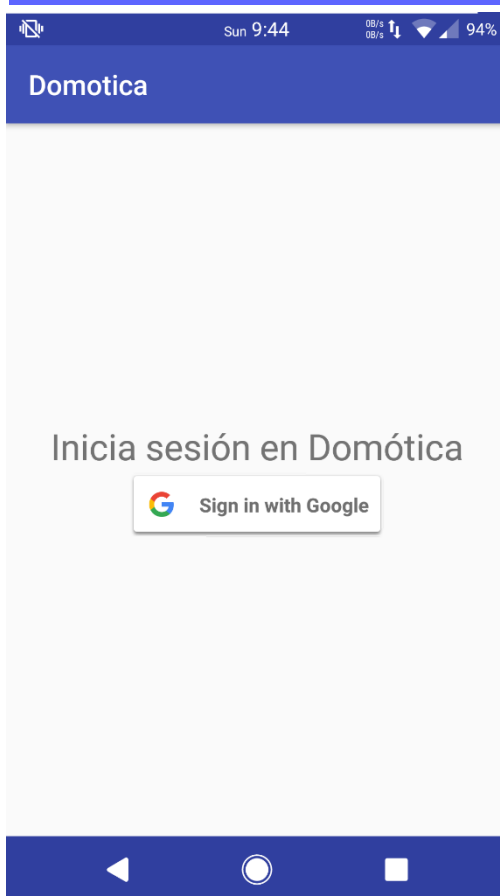
```
void loop() {  
    thing.handle();  
}
```

Indicar que en este método se puede añadir funcionalidad para actualizar cualquier variable, que esté relacionada con las acciones.

8.3. Manual de la aplicación

Por último, se va a explicar el funcionamiento de la aplicación, funcionalidad por funcionalidad, partiendo desde la vista de un nuevo usuario de la aplicación.

8.3.1. Iniciar sesión



Una vez abierta la aplicación, se muestra la siguiente pantalla. En ella apenas hay información, únicamente se pide iniciar sesión con la cuenta de Google.

En el caso de tener más de una cuenta de Google, se muestra cuál de ellas utilizar.

Indicar que, en futuros accesos, siempre y cuando no se haya cerrado sesión, esta vista no será necesaria, accediendo directamente al menú principal.

Ilustración 62. Manual Iniciar sesión

8.3.2. Menú principal

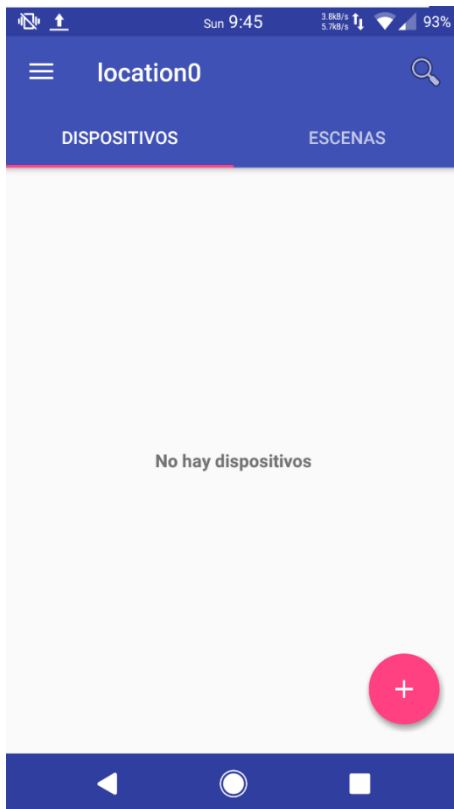


Ilustración 63. Manual Menú principal 1

Una vez accedemos al menú principal, observamos un sistema con dos pestañas, en las que se almacenarán los dispositivos y las escenas añadidas. Para este fin, se utiliza el botón flotante de la esquina inferior derecha.

Al presionar dicho botón, se desplegarán otros dos botones, en los que seleccionar qué se desea añadir, si una escena o un dispositivo.

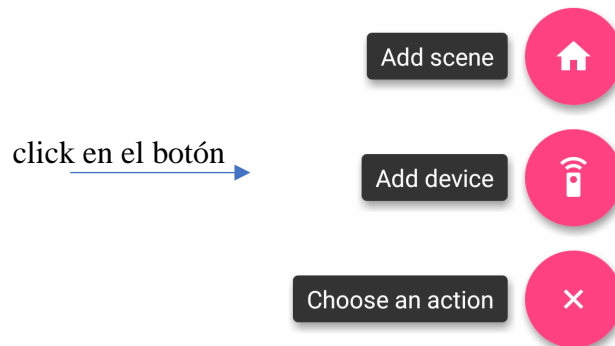


Ilustración 64. Manual Menú principal 2

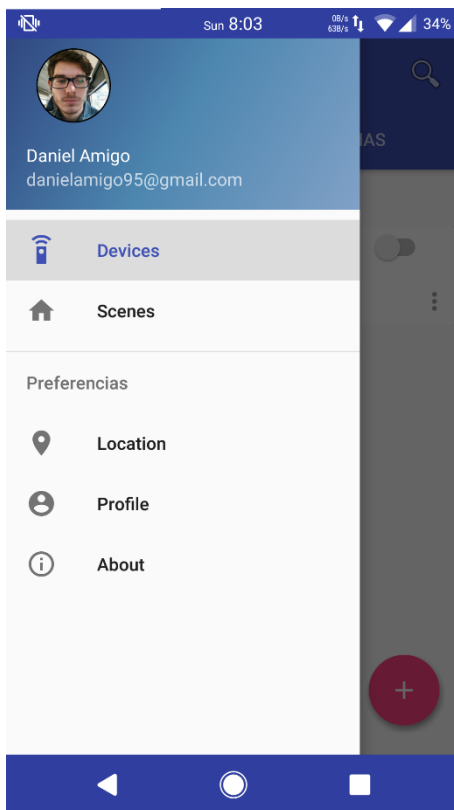


Ilustración 65. Manual Menú principal 3

Además, contamos con un menú desplegable en el lado izquierdo, el cual se muestra al presionar en el botón de la esquina superior izquierda, o deslizando desde el lateral izquierdo hacia el centro.

Este menú muestra la información de la cuenta de Google utilizada, y oculta el acceso a varias vistas de la aplicación:

- Localización: donde gestionar las distintas localizaciones que añade el usuario.
- Perfil: donde poder cerrar sesión y revocar acceso a la aplicación, eliminando la información.
- Acerca de: donde ver la información referente a la aplicación.

8.3.3. Dispositivos

8.3.3.1. Nuevo dispositivo

Si se presiona el botón de nuevo dispositivo, como se explicó anteriormente, se lleva a la siguiente vista.

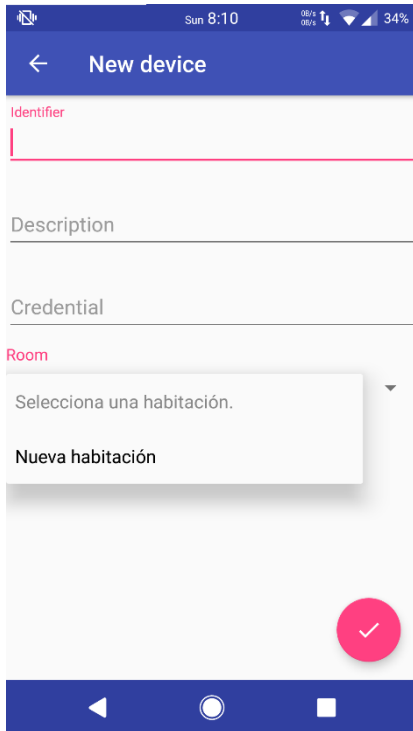


Ilustración 66. Manual Nuevo dispositivo 1

En esta vista se nos muestra un formulario, en el cual añadir los datos referentes al dispositivo a vincular, siendo el identificador y las credenciales los parámetros añadidos al dispositivo Arduino, y la descripción una breve explicación del usuario del dispositivo.

Posteriormente, se pide seleccionar la habitación en la que se encuentra el dispositivo. En caso de no encontrar la habitación deseada, se puede añadir una nueva presionando sobre la opción “Nueva habitación”, donde se genera un cuadro de texto para añadir una nueva.

Una vez completado el formulario, la aplicación conecta con el dispositivo, en un tiempo máximo de 7 segundos. Transcurridos esos segundos, se muestra al usuario la lista de acciones del dispositivo, para seleccionar la que desea como acción por defecto.

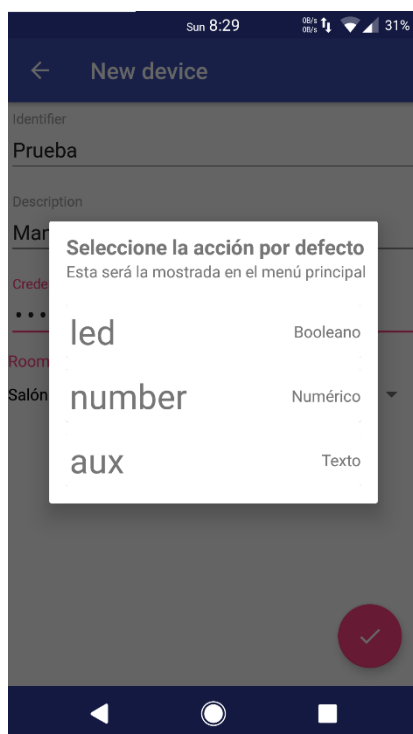


Ilustración 67. Manual Nuevo dispositivo 2

La selección de la acción por defecto cambia el estilo del dispositivo en el menú principal, permitiendo el ajuste rápido de la misma. Una vez seleccionada, se vuelve al menú principal.

8.3.3.2. Dispositivo menú principal

Una vez añadido el dispositivo, se queda en el menú principal de la siguiente manera:

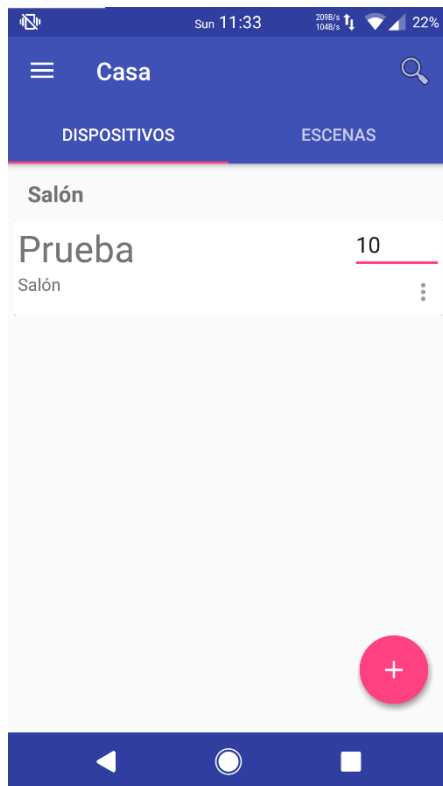


Ilustración 69. Manual Dispositivo menú principal 1

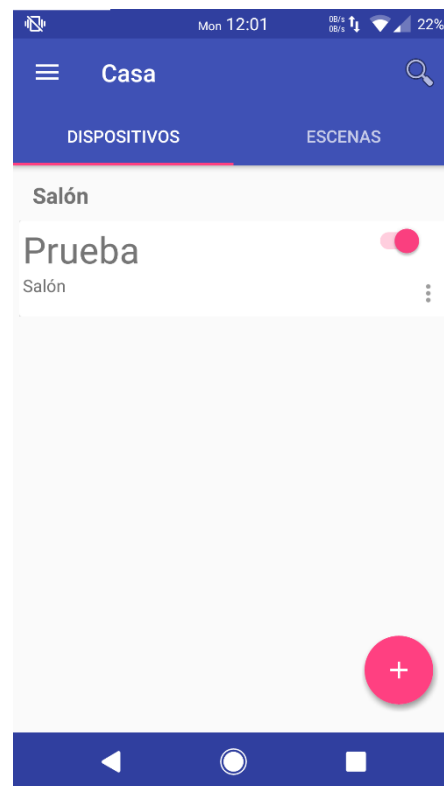


Ilustración 68. Manual Dispositivo menú principal 2

En función del tipo de acción por defecto seleccionada, se tendrá una de las dos vistas:

En la de la izquierda se seleccionó la acción booleana, y se puede pulsar el interruptor para conmutar el estado de la acción.

En la de la derecha se seleccionó cualquiera de las otras acciones, tipo numérica o tipo texto. Al pulsar sobre el cuadro de texto, se abre una ventana para introducir el nuevo valor.

Además, los dispositivos cuentan con un menú de tres puntos donde poder acceder a un botón para editar el dispositivo, y borrar el mismo, como se explicará posteriormente.

8.3.3.3. Vista dispositivo

Al realizar click en un dispositivo en el menú principal, se pasa a la vista del dispositivo.

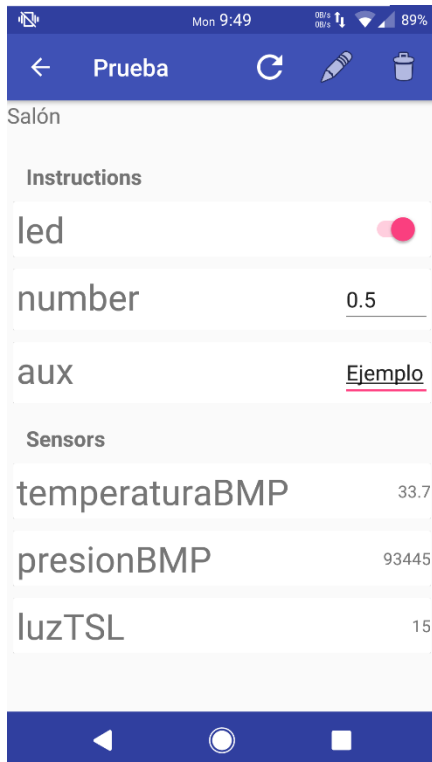


Ilustración 70. Manual Vista dispositivo

En esta, podemos observar varias funcionalidades:

En la barra superior, contamos con una flecha hacia atrás, para volver al menú principal, así como el nombre del dispositivo clickado. Después, tenemos tres botones: el primero de ellos sirve para recargar los valores de los sensores del dispositivo, y el segundo y el tercero sirven para editar el dispositivo o borrarlo.

Si vamos bajando, observamos la descripción del dispositivo, y después las instrucciones del dispositivo. Estas son modificables al realizar click sobre ellas, exactamente igual que como se explicó anteriormente en el menú principal.

Después contamos con los sensores, que son otras acciones que contempla el dispositivo, pero únicamente para mostrarnos información.

8.3.3.4. Editar dispositivo

Si pulsamos el botón editar en la vista anterior, o desde el menú de tres puntos de un dispositivo, accedemos a la vista de editar.

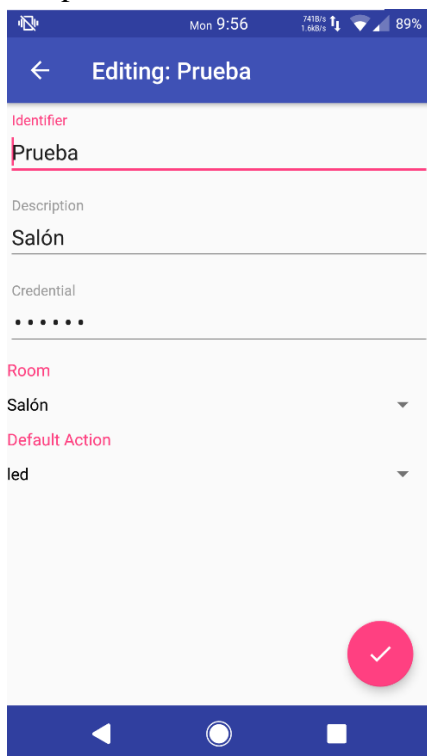


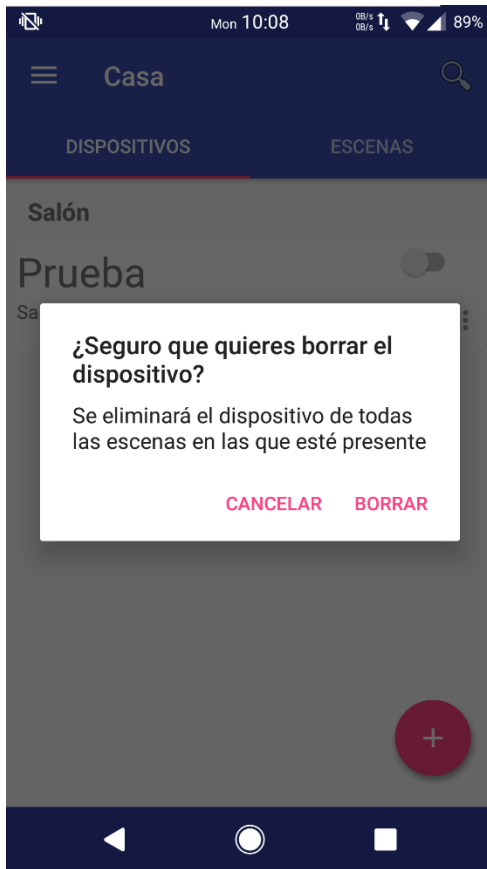
Ilustración 71. Manual Editar dispositivo

En esta contamos con un formulario similar al de la vista de nuevo dispositivo, pero ya rellenado con los valores actuales. Además, se cuenta con un parámetro más, la acción por defecto, al conocer ya las acciones del mismo.

Si se produce algún cambio en el dispositivo, se debe presionar el botón flotante de la esquina inferior derecha para salvar los cambios.

8.3.3.5. Borrar dispositivo

Para eliminar un dispositivo, como para editarlo, hace falta presionar el botón borrar en la vista anterior, o desde el menú de tres puntos de un dispositivo.



Al realizarlo se abre un aviso, en el que se indican las consecuencias de eliminar un dispositivo.

Al presionar Borrar, se elimina el dispositivo de la cuenta del usuario.

Ilustración 72. Manual Borrar dispositivo

8.3.4. Escenas

Una vez explicados los dispositivos, pasamos a las escenas. Una escena es una composición de acciones en un determinado estado.

8.3.4.1. Nueva escena

Al presionar el botón flotante del menú principal podemos acceder a la vista para crear una nueva escena.

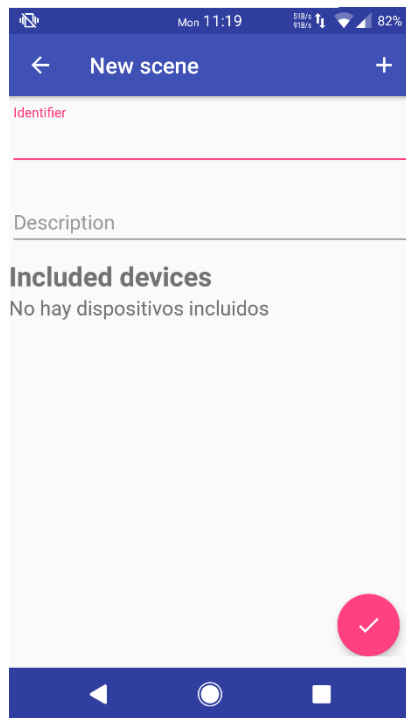


Ilustración 73. Manual Nueva escena 1

En esta vista nos aparecen todos los dispositivos, con sus correspondientes acciones, disponibles para ser seleccionadas. Una vez seleccionadas las acciones deseadas, con el botón flotante se añaden a la escena, y volvemos a la vista anterior.

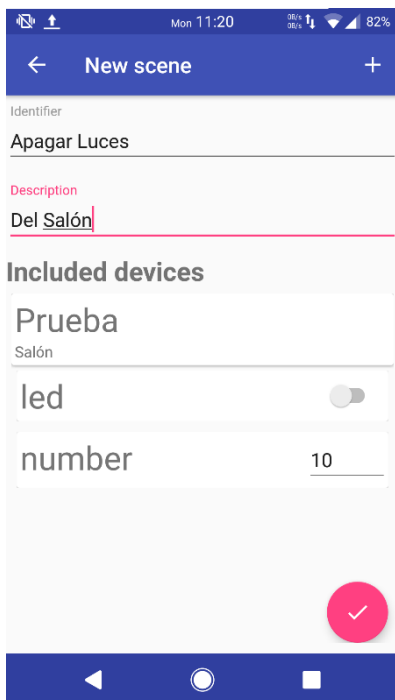


Ilustración 75. Manual Nueva escena 3

Al entrar en ella, observamos un simple formulario, en el que rellenar el identificador de la escena y una descripción.

Además, se nos indica que no tenemos acciones añadidas a la escena. Para este fin, tenemos el botón en la esquina superior derecha, en el que pasamos a una vista en la que seleccionar las acciones a incluir en la escena.

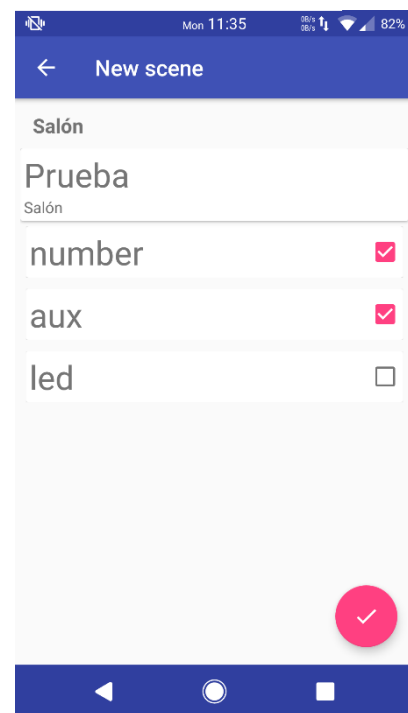


Ilustración 74. Manual Nueva escena 2

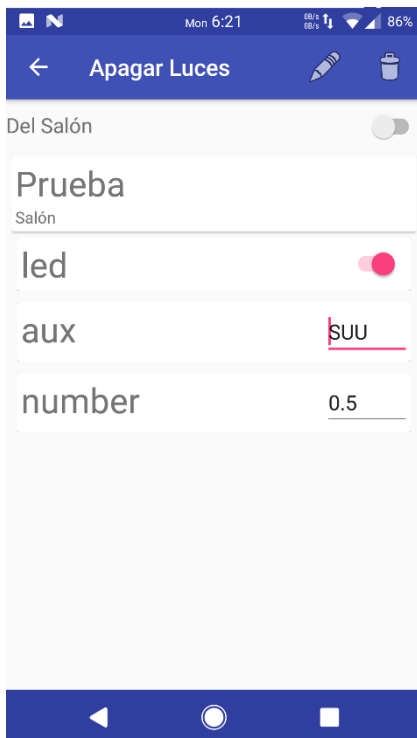
Al volver a la vista, aparecen las acciones seleccionadas, listas para especificar el estado al que se desea que se activen una vez se active la escena.

Si no se han seleccionado correctamente las acciones, se puede volver a la ventana de selección, manteniendo el estado y las acciones seleccionadas.

Una vez se pulse el botón flotante, se validará que los parámetros introducidos son correctos, y se almacenará la escena, volviendo al menú principal.

8.3.4.2. Vista escena

Al realizar click en una vista en el menú principal, se pasa a la vista de la escena.

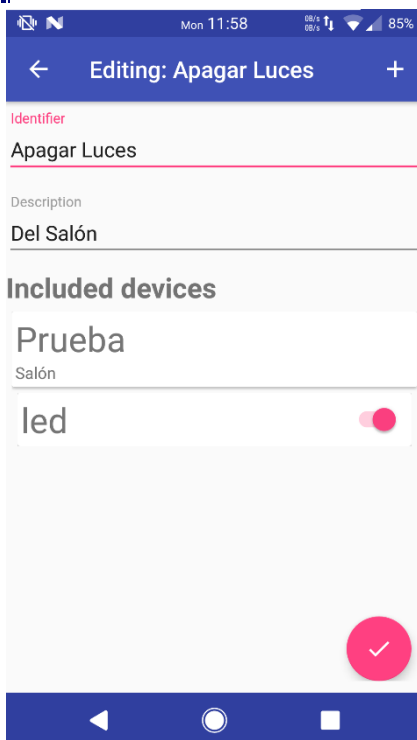


En esta contamos con los botones de editar la escena y para eliminarla. Además, mostramos la descripción de la escena y a su lado un interruptor para activar la escena.

Después, se muestran las acciones incluidas en la escena, pudiendo variar su valor desde esta vista.

Ilustración 76. Manual Vista escena

8.3.4.3. Editar escena



Para editar una escena, hay que acceder desde el menú principal, presionando el menú de tres puntos sobre la escena a editar.

Una vez en la vista de editar escena, contamos con un formulario idéntico al de la vista de nueva escena, pero rellenado con los valores de la escena.

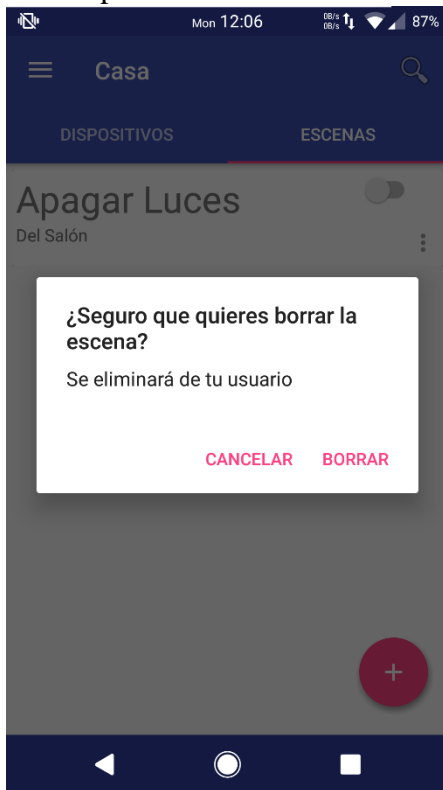
En este se puede acceder a la vista de añadir acciones, exactamente igual que en la vista anterior.

Una vez terminada la edición de la escena, se presiona el botón flotante para validar los cambios, y de ser correctos, almacenarlos.

Ilustración 77. Manual Editar escena

8.3.4.4. Borrar escena

Para eliminar una escena, hay que acceder desde el menú principal, presionando el menú de tres puntos sobre la escena a eliminar.



Al realizarlo se abre un aviso, en el que se indican las consecuencias de eliminar una escena.

Al presionar Borrar, se elimina la escena de la cuenta del usuario.

Ilustración 78. Manual Borrar escena

8.3.5. Localización

El usuario puede almacenar infinitas localizaciones ficticias en las que registrar sus dispositivos. Por defecto, una vez registrada la cuenta, se genera una localización inicial, denominada Casa.

8.3.5.1. Gestión localizaciones

Para acceder a la gestión de localizaciones, es necesario abrir, desde el menú principal, el menú desplegable, seleccionando la opción Localización.

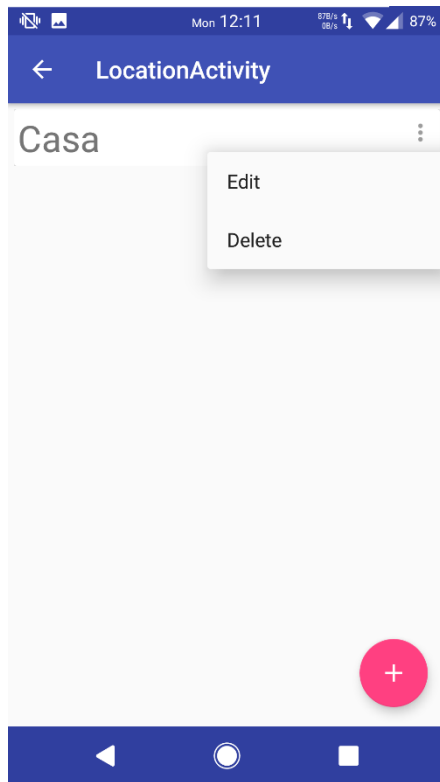


Ilustración 79. Manual Gestión localizaciones 1

Al presionar el botón flotante, se genera un cuadro de texto en el que introducir el nombre de la nueva localización. Este nombre no debe existir en la lista de localizaciones del usuario.

Por último, indicar que, al pulsar sobre una localización, se carga toda la información de dicha localización en el menú principal.

Una vez en la vista se muestran todas las localizaciones almacenadas por el usuario.

En esta vista se pueden eliminar y borrar las localizaciones seleccionadas, mediante el menú de tres puntos.

Al presionar sobre editar se genera un cuadro de texto para cambiar el nombre de la localización.

Al presionar borrar se crea un aviso indicando que al borrar la localización se pierden los dispositivos y escenas añadidos en dicha localización. Esta opción solo está disponible cuando haya más de una localización.

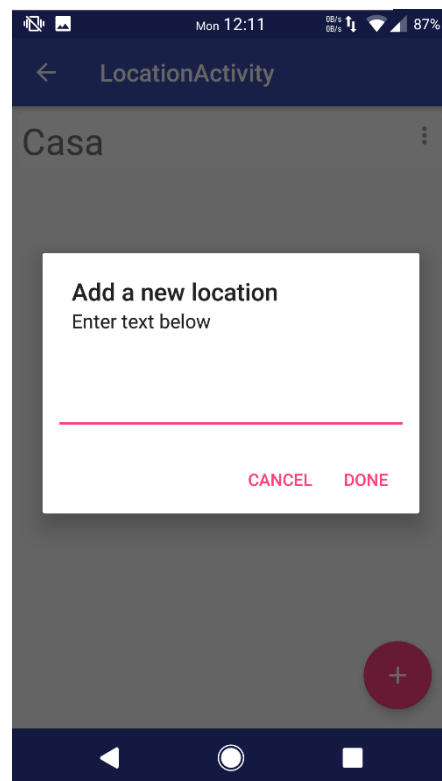
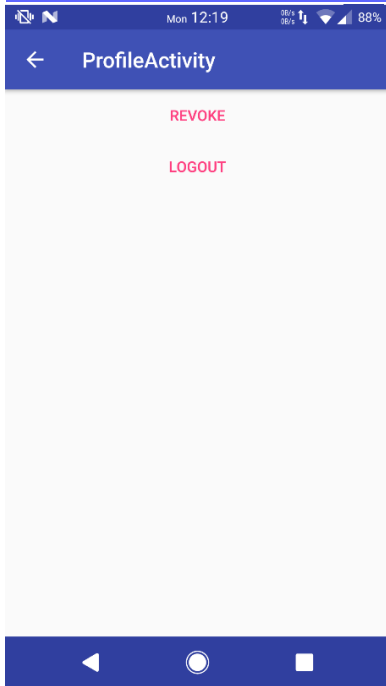


Ilustración 80. Manual Gestión localizaciones 2

8.3.6. Gestión del perfil



Para acceder a la gestión del perfil, es necesario abrir, desde el menú principal, el menú desplegable, seleccionando la opción Perfil.

En esta podremos cerrar sesión de la cuenta, volviendo a la vista de inicio de sesión.

La otra opción disponible es revocar el acceso a la aplicación. Con esta función, se elimina todos los datos almacenados en la aplicación de dicho usuario, incluidas todas las localizaciones, dispositivos y escenas.

Ilustración 81. Manual Gestión del perfil

8.3.7. Acerca de

Para acceder a la información relativa a la aplicación, es necesario abrir, desde el menú principal, el menú desplegable, seleccionando la opción Acerca de.

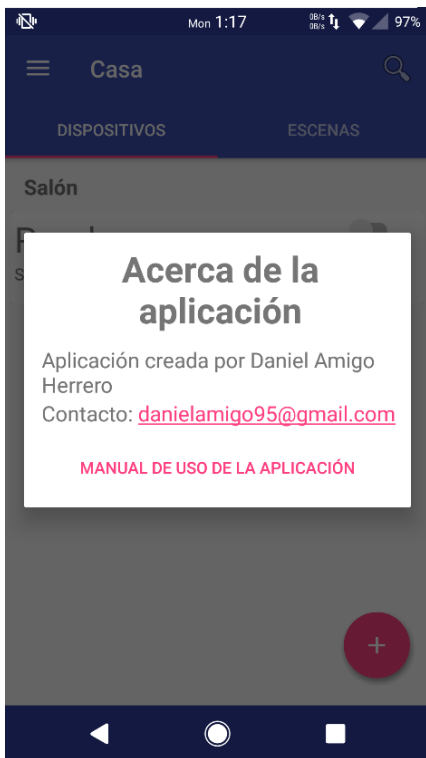


Ilustración 83. Manual Acerca de 2

Aquí se genera un cuadro de texto donde se muestra la información del desarrollador, su contacto, y un enlace para descargar el presente manual de usuario.



Ilustración 82. Manual Acerca de 1

9. Anexo B. Acrónimos, definiciones y referencias

Este anexo contiene todos los acrónimos y definiciones necesarias para comprender perfectamente los términos más técnicos utilizados en el documento. Además, se incluyen las referencias utilizadas a lo largo del documento, con enlaces disponibles en la medida de lo posible.

9.1. Acrónimos y definiciones

Android: sistema operativo que se emplea en dispositivos móviles, por lo general con pantalla táctil. De este modo, es posible encontrar tabletas (tablets), teléfonos móviles (celulares) y relojes equipados con Android, aunque el software también se usa en automóviles, televisores y otras máquinas.

API: Application Programming Interface. conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas, sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software.

Arduino: plataforma de hardware y software de código abierto, basada en una sencilla placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación Processing.

Bit: unidad mínima de información digital que puede ser tratada por un ordenador. Proviene de la contracción de la expresión binary digit (dígito binario).

Bluetooth: sistema de comunicación inalámbrica que permite la interconexión de diferentes dispositivos electrónicos (PCs, teléfonos fijos o móviles, agendas electrónicas, auriculares, etc.); es un estándar creado por importantes empresas del sector de la informática y de las telecomunicaciones. Bluetooth, que en inglés significa literalmente "diente azul", era el apodo de un jefe vikingo del siglo IX D.C.

C: lenguaje de programación creado en 1972 por Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B, a su vez basado en BCPL.

Caso de uso: descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso.

Dashboards: representación gráfica de las principales métricas

Diagrama de Actividad: también denominados diagramas de flujo, es utilizado en conjunción de un diagrama de casos de uso para ayudar a los miembros del equipo de desarrollo a entender como es utilizado el sistema y cómo reacciona en determinados eventos.

Diagrama Gantt: herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado

Ethernet: Sistema de red de área local de alta velocidad. Se ha convertido en un estándar de red corporativa.

Feedback: capacidad de un emisor para recoger reacciones de los receptores y modificar su mensaje, de acuerdo con lo recogido.

GB: Gigabyte. Unidad de medida de la capacidad de memoria y de dispositivos de almacenamiento informático (disquete, disco duro, CD-ROM, DVD, etc.). Un GB corresponde a 1.024 Megabytes.

Gradle: herramienta Java de automatización de builds.

Hardware: Componentes físicos de un ordenador o de una red, en contraposición con los programas o elementos lógicos que los hacen funcionar.

HomeKit: accesorios para la aplicación Apple Home.

IDE: Integrated Development Environment. Aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

IEEE: Institute of Electrical and Electronics Engineers

IES: Instituto de Educación Secundaria

Internet: Red de telecomunicaciones nacida en 1969 en los EE.UU. a la cual están conectadas centenares de millones de personas, organismos y empresas en todo el mundo, mayoritariamente en los países más desarrollados, y cuyo rápido desarrollo está teniendo importantes efectos sociales, económicos y culturales, convirtiéndose de esta manera en uno de los medios más influyentes de la llamada Sociedad de la Información y en la Autopista de la Información por excelencia. Fue conocida como ARPANET hasta 1974.

iOS: sistema operativo móvil de la multinacional Apple Inc, funcional en dispositivos iPhone, iPad e iPod Touch.

IoT: Internet of Things. Concepto que relaciona la interconexión entre un objeto físico cualquiera e Internet. Es decir, cualquier objeto tiene conectividad mediante una dirección IP unívoca y una serie de sensores que le proporcionan utilidades.

IP: Internet Protocol. Conjunto de reglas que regulan la transmisión de paquetes de datos a través de Internet. La versión actual es IPv4 mientras que en el proyecto Internet2 se intenta implementar la versión 6 (IPv6), que permitiría mejores prestaciones dentro del concepto QoS (Quality of Service).

Java: Lenguaje de programación desarrollado por la empresa Sun para la elaboración de pequeñas aplicaciones exportables a la red (applets) y capaces de operar sobre cualquier plataforma a través, normalmente, de navegadores WWW. Permite dar dinamismo a las páginas web.

JSON: JavaScript Object Notation. Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo

JUnit: conjunto de clases (framework) que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera.

KB: Kilobyte. Unidad de medida de la capacidad de memoria y de dispositivos de almacenamiento informático (disquete, disco duro, CD-ROM, DVD, etc.). Un KB corresponde a 1.024 bytes.

Layout: croquis de donde debe de ir cada elemento informativo de una página web.

LED: Light-Emitting Diode. tipo de diodo empleado en computadoras, paneles numéricos, etc.

Linux: Versión de libre distribución del sistema operativo UNIX; fue desarrollada por Linus Torvald.

MB: Megabyte. Unidad de medida de la capacidad de memoria y de dispositivos de almacenamiento informático (disquete, disco duro, CD-ROM, DVD, etc.). Un MB corresponde a 1.024 Kilobytes.

Microprocesadores: procesador de muy pequeñas dimensiones en el que todos los elementos están agrupados en un solo circuito integrado.

Micro-USB: Micro Universal Serial Bus. Tecnología de conectividad USB (Universal Serial Bus) con un tamaño mucho menor y una mayor durabilidad que el propio USB.

MHz: Megahercio. Unidad de medida de la frecuencia.

MP3: MPEG Audio Layer 3. Método de grabación y reproducción digital de audio que permite una buena calidad de sonido, teniendo los ficheros generados un tamaño razonable. Su rápida popularización a través de Internet es causa de grave preocupación para las empresas del ramo y a los organismos encargados de la protección de los derechos de autor.

MVC: Modelo Vista Controlador. Arquitectura de diseño que separa los distintos aspectos de la aplicación en tres capas.

MySQL: sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL).

NoSQL: enfoque hacia la gestión de datos y el diseño de base de datos que es útil para grandes conjuntos de datos distribuidos.

Open-Source: Código Abierto. software distribuido y desarrollado libremente.

Paradigma: provee (y determina) la visión y métodos de un programador en la construcción de un programa o subprograma.

PCB: Printed Circuit Board (circuito impreso en español). Conjunto de conductores y otros elementos eléctricos interconectados por los que pasa la corriente eléctrica.

PF: Prueba Flujo.

PH: Prueba Hardware.

PDA: Personal Digital Assistant. Ordenador de pequeño tamaño cuya principal función era en principio la de mantener una agenda electrónica, aunque cada vez más se va confundiendo con los ordenadores de mano y de palma, y se está integrando con otros dispositivos como los teléfonos móviles. Se le denomina también ordenador de bolsillo.

PDF: Portable Document Format. Formato de almacenamiento para documentos digitales independiente de plataformas de software o hardware.

Procesador: programa informático que procesa o somete a una serie de operaciones la información introducida en la computadora

Procesador gráfico: GPU. coprocesador dedicado al procesamiento de gráficos u operaciones de coma flotante, para aligerar la carga de trabajo del procesador central en aplicaciones como los videojuegos o aplicaciones 3D interactivas.

RAM: Random Access Memory. Memoria principal de la computadora, donde residen programas y datos, sobre la que se pueden efectuar operaciones de lectura y escritura.

Requisito de Software: necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio.

RSF: Requisito de Software Funcional

RSNF: Requisito de Software No Funcional

RUC: Requisito de Usuario de Capacidad

RUP: Rational Unified Process. Proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM.

RUR: Requisito de Usuario de Restricción

Script: Conjunto de caracteres formado por mandatos y secuencias de tecleo, que se utiliza muy a menudo en Internet para automatizar tareas muy habituales como, por ejemplo, la conexión a la red (login).

Servidor: Sistema que proporciona recursos (por ejemplo, servidores de ficheros, servidores de nombres). En Internet este término se utiliza muy a menudo para designar a aquellos sistemas que proporcionan información a los usuarios de la Red.

SDK: Software Development Kit. Conjunto de herramientas y programas de desarrollo que permite al programador crear aplicaciones para un determinado paquete de software, estructura de software, plataforma de hardware, sistema de computadora, consulta de videojuego, sistema operativo o similar.

SI: Sistema Internacional. Estándar científico de pesas y medidas.

Siri: sistema de reconocimiento de voz de Apple.

SO: Sistema Operativo (OS en inglés). Un sistema operativo es un programa especial que se carga en un ordenador tras ser encendido y cuya función es gestionar los demás programas, o aplicaciones, que se ejecutarán en dicho ordenador, como, por ejemplo, un procesador de texto o una hoja de cálculo, o la impresión de un texto en una impresora o una conexión a Internet.

Smart City: aquella ciudad que aplica las tecnologías de la información y de la comunicación (TIC) con el objetivo de proveerla de una infraestructura que garantice: Un desarrollo sostenible.

Smartphone: teléfono celular con pantalla táctil, que permite al usuario conectarse a internet, gestionar cuentas de correo electrónico e instalar otras aplicaciones y recursos a modo de pequeño computador.

SMS: Short Message Service. Procedimiento de envío y recepción de mensajes escritos de pequeño tamaño a través del teclado y la pantalla de los teléfonos móviles. Para agilizar su escritura y disminuir el coste del envío, está dando lugar a una forma específica y abreviada de escritura.

Software: programas o elementos lógicos que hacen funcionar un ordenador o una red, o que se ejecutan en ellos, en contraposición con los componentes físicos del ordenador o la red.

TFG: Trabajo de Fin de Grado.

Transistor: dispositivo electrónico semiconductor utilizado para entregar una señal de salida en respuesta a una señal de entrada.

UC: Caso de Uso. Descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso.

UI: Interfaz de Usuario. Medio con que el usuario puede comunicarse con una máquina, equipo, computadora o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo.

UML: Unified Modeling Language. Estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software (programas informáticos).

UNE-EN: Una Norma Española-European Norm. Conjunto de normas, normas experimentales e informes (estándares) creados en los Comités Técnicos de Normalización (CTN) de la Asociación Española de Normalización y Certificación (AENOR).

Unstructured Review: técnica de elicitación que tiene como objetivo de obtener mayor información para la posterior elicitación de requisitos.

USB: Universal Serial Bus. Estándar utilizado en los PCs con el fin de reconocer los dispositivos hardware (una impresora, un teclado, ...) y ponerlos en funcionamiento de forma rápida y sencilla. Elimina la necesidad de instalar adaptadores en el PC.

WiFi: Wireless Fidelity. Tecnología de comunicación inalámbrica que permite conectar a internet equipos electrónicos, como computadoras, tablets, smartphones o celulares, etc., mediante el uso de radiofrecuencias o infrarrojos para la transmisión de la información.

Wireframe: boceto donde se representa visualmente, de una forma muy sencilla y esquemática la estructura de una página web. El objetivo de estos es definir el contenido y la posición de los diversos bloques de tu web o aplicación.

XML: eXtensible Markup Language. Lenguaje desarrollado por el W3 Consortium para permitir la descripción de información contenida en el WWW a través de estándares y formatos comunes, de manera que tanto los usuarios de Internet como programas específicos (agentes) puedan buscar, comparar y compartir información en la red. El formato de XML es muy parecido al del HTML aunque no es una extensión ni un componente de éste.

9.2. Referencias

- [1] RTVE, 01 10 2015. [En línea]. Available: <http://www.rtve.es/noticias/20151001/79-hogares-espanoles-tiene-acceso-internet-mayoria-banda-ancha/1230482.shtml>. [Último acceso: 08 06 2017].
- [2] BOE, «Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal,» 14 de diciembre de 1999, pp. 43088-43099.
- [3] nramos, «Norberto Ramos,» 29 10 2014. [En línea]. Available: <http://www.personales.ulpgc.es/nramos.dit/?q=node/96>. [Último acceso: 08 06 2017].
- [4] IECOR. [En línea]. Available: <http://www.iecor.com/definicion-de-domotica/>. [Último acceso: 08 06 2017].
- [5] RAE. [En línea]. Available: <http://dle.rae.es/srv/fetch?id=E7W0v9b>. [Último acceso: 08 06 2017].
- [6] G. Arjona, 15 12 2014. [En línea]. Available: <https://www.observatoriodelaaccessibilidad.es/espacio-divulgativo/articulos/domotica-accessibilidad-hogar-digital.html>. [Último acceso: 08 06 2017].
- [7] «gfsapivitoria,» [En línea]. Available: <http://gfsapivitoria.blogspot.com.es/p/servicios-y-funciones-gestion-de.html>. [Último acceso: 08 06 2017].
- [8] J. M. F. Luna, 09 2006. [En línea]. Available: http://leo.ugr.es/J2ME/INTRO/intro_4.htm. [Último acceso: 08 06 2017].
- [9] S. F. Aroche, «Maestros del Web,» 06 06 2006. [En línea]. Available: <http://www.maestrosdelweb.com/compuhis/>. [Último acceso: 08 06 2017].
- [10] R. G. Latinoamérica. [En línea]. Available: <http://redgrafica.com/Breve-historia-de-los-celulares-y->. [Último acceso: 08 06 2017].
- [11] «El País,» 30 09 2010. [En línea]. Available: http://elpais.com/diario/2010/09/30/ciberpais/1285813479_850215.html. [Último acceso: 08 06 2017].

- [12] M. Canal, 31 01 2014. [En línea]. Available: <http://www.muycanal.com/2014/01/31/futuro-del-telefono-movil>. [Último acceso: 08 06 2017].
- [13] «Applesfera,» 13 10 2016. [En línea]. Available: <https://www.applesfera.com/ios/la-evolucion-de-ios-desde-sus-origenes-una-carrera-para-ser-el-mejor-sistema-operativo-movil-de-la-historia>. [Último acceso: 08 06 2017].
- [14] H. Hernández, «Malavida,» 17 11 2016. [En línea]. Available: <http://www.malavida.com/es/analisis/la-historia-de-android>. [Último acceso: 08 06 2017].
- [15] «Gartner,» 02 2017. [En línea]. Available: <http://www.gartner.com/newsroom/id/3609817>. [Último acceso: 08 06 2017].
- [16] M. Rouse, «TechTarget,» [En línea]. Available: <http://searchdatacenter.techtarget.com/es/definicion/Internet-de-las-cosas-IoT>. [Último acceso: 08 06 2017].
- [17] N. Rivera, «Hipertextual,» 20 06 2015. [En línea]. Available: <https://hipertextual.com/2015/06/internet-of-things>. [Último acceso: 08 06 2017].
- [18] E. M. Rodríguez, «El Mundo,» 07 07 2015. [En línea]. Available: <http://www.elmundo.es/tecnologia/2015/07/07/559add51e2704e892c8b459b.html>. [Último acceso: 08 06 2017].
- [19] D. Evans, «Cisco,» 04 2011. [En línea]. Available: http://www.cisco.com/c/dam/global/es_es/assets/executives/pdf/Internet_of_Things_IoT_IBSG_0411FINAL.pdf. [Último acceso: 08 06 2017].
- [20] «Educación intef,» [En línea]. Available: <http://platea.pntic.mec.es/~lmarti2/arduino/introardu/>. [Último acceso: 08 06 2017].
- [21] «Aprendiendo Arduino,» [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2016/12/11/hardware-arduino/>. [Último acceso: 08 06 2017].
- [22] «Fayerwayer,» 14 05 2014. [En línea]. Available: <https://www.fayerwayer.com/2015/05/nodemcu-el-sucesor-de-arduino-integra-wi-fi-y-cuesta-menos-de-ocho-dolares/>. [Último acceso: 08 06 2017].
- [23] Apple, «Home,» [En línea]. Available: <https://www.apple.com/es/ios/home/>. [Último acceso: 08 06 2017].
- [24] Wink. [En línea]. Available: <https://www.wink.com/>. [Último acceso: 08 06 2017].

- [25] S. Direct, «My Verisure,» [En línea]. Available: <https://www.securitasdirect.es/es/servicios/myverisure>.
- [26] Blynk. [En línea]. Available: <http://www.blynk.cc/>. [Último acceso: 08 06 2017].
- [27] Thinger. [En línea]. Available: <https://thinger.io/>. [Último acceso: 08 06 2017].
- [28] Google, «Firebase,» [En línea]. Available: <https://firebase.google.com/>. [Último acceso: 06 08 2017].
- [29] Amazon. [En línea]. Available: <https://aws.amazon.com/es/>. [Último acceso: 08 06 2017].
- [30] C. Larman, UML y patrones : introducción al análisis y diseño orientado a objetos y al proceso unificado, Prentice-Hall, 2002.
- [31] BOE. [En línea]. Available: https://www.boe.es/diario_boe/txt.php?id=BOE-A-2009-5688. [Último acceso: 08 06 2017].
- [32] S. Social. [En línea]. Available: http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm. [Último acceso: 08 06 2017].
- [33] «Días laborables,» [En línea]. Available: http://www.dias-laborables.es/dias_laborables_feriados_2017.htm#aochd. [Último acceso: 08 06 2017].
- [34] «citapreviainem,» [En línea]. Available: <http://www.citapreviainem.es/el-derecho-a-vacaciones/>. [Último acceso: 08 06 2017].
- [35] J. Horton, de *Android Programming for Beginners*, Packt Publishing, 2015, p. 400.

10. Anexo C. Inglés

Como se requiere en el plan de estudios de 2011, a continuación, se realiza un amplio resumen del trabajo en inglés. Para este fin se va a seguir la estructura del documento, explicando brevemente los distintos apartados, y en los que cuenten con tablas e ilustraciones, describiendo brevemente su cometido en vez de duplicar las mismas aquí.

Since it is needed in the study plan of 2011, a wide summary of the work in English is realized below. For this end, one is going to follow the structure of the document, explaining brief the different paragraphs, and in those who should possess tables and illustrations, describing brief his assignment instead of duplicating the same ones here.

Abstract

The sector of the home automation has grown very much in the last years. It's a very big area where exists much for exploring now that we have a technology more than enough for it. Besides the physical advantages that the home automation carries, as the security and the energy saving that it offers are more than demonstrated.

On the other hand, we have the mobile devices or smartphones, that ten years ago did not exist and they have taken root to the society in a way never seen before. Our actual society is not conceived without them, we all have at least one intelligent device.

If we join both concepts, we have an application for mobile devices (Android in our case), capable of controlling the home automated devices.

But there is a problem, this concept already exists, controlling devices of all kinds and brands. But, as will be described along the document, these solutions rely on very expensive devices for his functionality.

In this project, we will focus on an area more generic than fixed devices, working with the technology Arduino, giving support to all kinds of devices, on having been able to be programmed to realize any action.

In addition, this solution implies another benefit more, since the devices Arduino interaction with the application connected by WiFi, the project includes IoT's concept and gives a practical solution and of quality to the persons who look for a home automated wireless system of low budget, something that doesn't exist in the world.

Keywords: Android, Arduino, domotic, home automated system, IoT, application, Java, Thinger, Firebase.

10.1. Introduction

10.1.1. Motivation

10.1.1.1. Personal

This paragraph will try to explain to the readers of the document the reasons why this project created so much interest among the developers.

I want to start by emphasizing the support and sincerity that my tutor offered me, Miguel Ángel Patricio, from the beginning. Without him this project would never have existed. I had an idea of an application to do my Bachelor's thesis, and when a lot of teachers told me that it was a good idea, only Miguel Ángel thought about the idea and he explained me that it wasn't a good idea, because there are some similar applications that do exactly what I wanted to do.

Besides giving me his sincerer opinion about my idea, Miguel Ángel proposed me one idea that had called me the attention, to develop an application to give support to a home automated system, but without switchboard, based on intelligent devices connected to a server by WiFi.

After several days getting informed about the idea, I liked the idea very much, it was a very interesting one. There wasn't any similar application in the mobile market, there is no domotic system based on IoT.

In addition, it was providing to myself both priorities that I had marked for this Bachelor's thesis: to develop an application for devices Android, and something that is useful.

10.1.1.2. Technological

The humans always have been searching help to make their actions easier. From the prehistory, we always have had the curiosity and the vision to create a thing that could be useful. The technology always has existed, and has followed an exponential curve in our existence until now, where our life is ruled by the technological changes that take place.

The steam engine, the internal combustion engines, the electronic devices ... This way we have come to a society where always talks about the technological revolution, of how the Internet of the things and the social networks they are the nearby future, but we see at the time it happens and we can't see final products that use that technology for real needs.

With this technological motivation arises the idea of this application. In a booming area, as the home automation, highly wished and practically inaccessible for the middle class

because of the high costs involved for use, there doesn't exist any solution that exploits a concept so interesting as the IoT. Therefore, there exists a niche of unexplored market for the developers of mobile applications, which will include the application Android of this project.

Imagine having a 10 € home automated, programmable device, for the use that you want, easily to use, on having needed neither any cable nor an installation in the house and with a quality application to control it. Isn't it incredible?

10.1.1.3. Socio-economic environment

Today, it's a reality, you can see new construction housings with home automated systems factory installed, designed entirely for the needs of each family. But apart, there are a part of the users who demand domotic devices that satisfy his needs with home automated that they could adapt to his current housings. This comes strongly influenced by the evolution of the technology, especially of the Smartphones, which facilitate our life to the point that, we want this kind of help also in our housings.

In the economic aspect, this project is more than viable for any citizen. The technology has advanced so much, that anyone can obtain Arduino devices for less than five Euros. With one of them, and Internet in the housing, which is more that implanted in the society, having more than 75 % of the Spanish population I access to it, stays more that the economic aspect isn't an impediment for the citizens. Furthermore, a well programmed home automated system can save for the user. For example, maximizing the hours of light, or anticipating human mistakes, leave the lights on or others unforeseen.

In the social aspect, it is different. The Spanish population is separated by the technological changes. Younger and middle-aged population is adapted to the technological development. They have grown of the hand of social networks, infinity of different devices, and consume of Internet daily... they would see with good eyes the implantation of home automated systems in his housings. They are not afraid, in general, to the machines, see them as a help. On the other hand, the majority of the third age population, might show a major rejection to which a machine manages the door of his house, for example. They are firm in his habits, logically. It's something strange for them, they haven't grown up with this technology, and it might be an impediment for the project. It's not necessary to forget that this group of the population is the one that has a major possibility of being a buying potential of home automated systems.

Despite it, due to the low investment that it is necessary to realize to make it to work, it's possible to wait for a good reception of the application.

Finally, the project would not have scarcely environmental impact. It needs only electronic components, which have an extensive enough chain of recycling in the society, allowing re-use practically all the materials necessary for his construction.

10.1.2. Objectives

In this paragraph are explained the conclusions extracted from the meeting with the client and tutor, Miguel Ángel Patricio. At this meeting, the project manager performed an Unstructured Review to obtain the major quantity of possible information to transmit it to the development team.

Of this meeting, in short, the following conclusions were extracted about the application:

- Is necessary a database that stores the configuration for every user and his devices.
- The users must have access to different locations, to differentiate his devices if they possess several locations.
- The client wants a functionality to realize several actions in an automatic way, at the same time. For example, he commented, “to raise the blinds and to turn off the lights”.
- In addition, the client finishes emphasizing in that the project must be delivered in the second period of defense of the final degree of the UC3M in 2017.

With this knowledge, the aims marked in the project will be:

1. To analyze in depth the home automated systems existing at present, observing his functionalities and his limitations.
2. To analyze the needs that must have a home automated system management tool.
3. To design the architecture of the system to develop.
4. And, finally, to proceed to the development of the application.

10.1.3. Regulatory environment

In this paragraph, a study has been realized in the current regulatory framework in Spain, to develop the application Android without breaking any law that might compromise neither the same application nor to the developers.

The present project complies with all the in-force regulations, since it doesn't compile the information of the user any more that for the proper use of the application, and the application allows the user to eliminate all information about him on our system when he wishes it.

10.2. State of the art

In this section, we are going to realize a brief introduction to the subjects to be dealt in the present document, guaranteeing to any reader of the perfect comprehension of the following sections.

This section divides in three paragraphs, depending on the type of aspect to treat:

- Technological: it is deepened in the technological aspects to treat.
- Functional: there is realized a study of the market of home automated applications, observing the advantages and disadvantages of each one.
- Estudio de alternativas de solución: finalmente, se realiza un estudio sobre los aspectos tecnológicos requeridos para el desarrollo de la aplicación, comparando las alternativas existentes en el mercado.
- Study of alternatives of solution: finally, a study is realized on the technological aspects needed for the development of the application, comparing the existing alternatives on the market.

10.2.1. Technological

In this subpart, we explain the following topics: Home automation, mobile devices (Android), Internet of Things, Arduino.

10.2.2. Functional

In this subpart, we analyze the next mobile applications: Apple Home, Wink, Securitas Direct and Blynk. After seeing his advantages and disadvantages, we realized a table where we compare our project to the other applications:

<i>Advantages</i>	<i>Application</i>	Home	Wink	Securitas Direct	Blynk	Project
<i>Management of devices</i>		✓	✓		✓	✓
<i>Low cost of devices</i>					✓	✓
<i>Minimalist interface</i>		✓	✓	✓		✓
<i>Configuration of devices</i>		✓	✓			✓
<i>Multiple control of devices</i>		✓				✓
<i>Management of locations</i>		✓		✓		✓
<i>Control from out of house</i>		*		✓	✓	✓

Tabla 141. Comparison to other application

*: need a device inside the house

10.2.3. Study of alternatives of solution

Before carrying out the development of the application, it's necessary to observe what structure to mount to support the app, with the necessary conditions for his correct functioning.

In this paragraph, we will deepen into the existing alternatives into each of the needs that we require:

10.2.3.1. System of management and control of devices

The application needs a system to communicate with the devices Arduino, and to allow to implement his control from the application. For this assignment, due to the requirements of the client, we decided to use the application Thingier, a system of management of intelligent devices, which fits perfectly with our aim.

10.2.3.2. Database and server

The application needs a 24 hours of the day available database, in which to store all the information of the users, and for it we need a server that allows the service the database.

To settle this need, we have thought about the possibility of creating a database MySQL together with an own server, but it was a very expensive and complex solution, on what we decide for Google Firebase uses, a service online free, that amply meets our requirements.

10.3. Management and organization of the Project

This paragraph is going to explain the functioning of the team of development of this project. To do this, there are going to be detailed the used methodology and the roles occupied by members of the team.

10.3.1. Methodology of software development used

A good methodology is key to accomplish with the quality and the time demanded by a client. In this case, the group has decided to bet for Craig-Larman's method, due to his knowledge acquired during the degree. Should be noticed that this methodology will be adapted to the needs of this project, simplifying some of his tasks.

Craig-Larman is a method based on a small version of the Rational Unified Process (RUP), explaining held up a suitable way to develop software. Larman's method is based on one life cycle, which characteristics are:

- Is iterative, dividing the project in N chunks that are approached separately. Therefore, we realize analysis, coding and testing separately. This separation is realized according to cases of use.
- Is incremental: each of the chunks of the project is realized on the previous ones, obtaining every time a more functional product. Thanks to it, we can show functionality to the client quickly, making the client perceive the advance of the development.
- Is directed by the cases of use. We work to dividing the cases of use, which are transforming to functionality in the code.

Graphically, Craig-Larman can be summarized in:

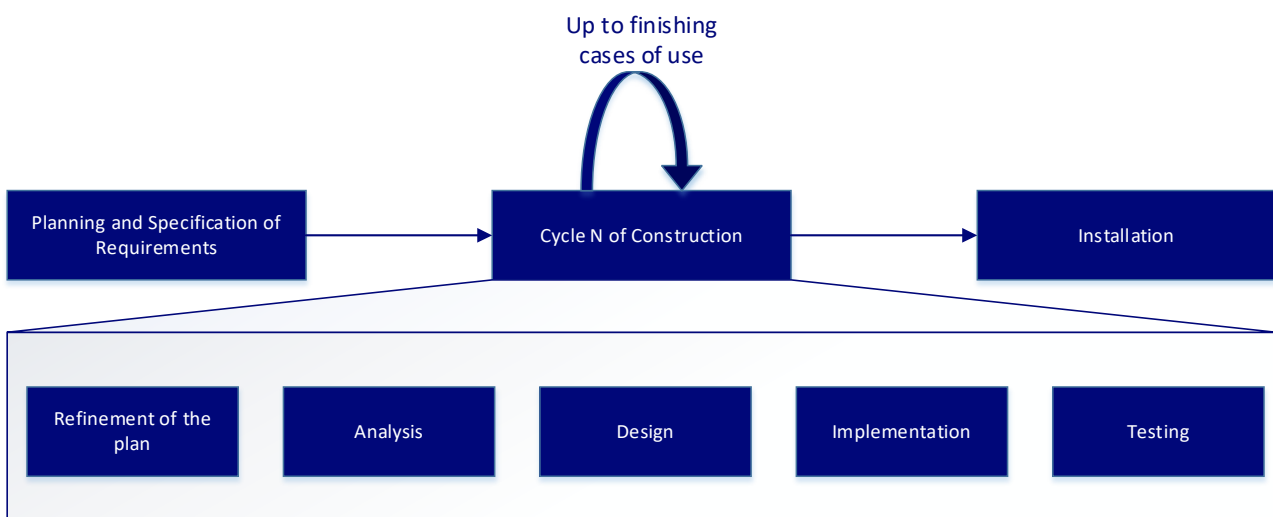


Ilustración 84. Craig Larman summarized

10.3.2. Organization of the project

In this paragraph, we are going to define the resources (human, hardware and software) necessary for the development of the project, as well as the roles generated inside the team. Due to being an individual project, a member will be able to perform more than one role.

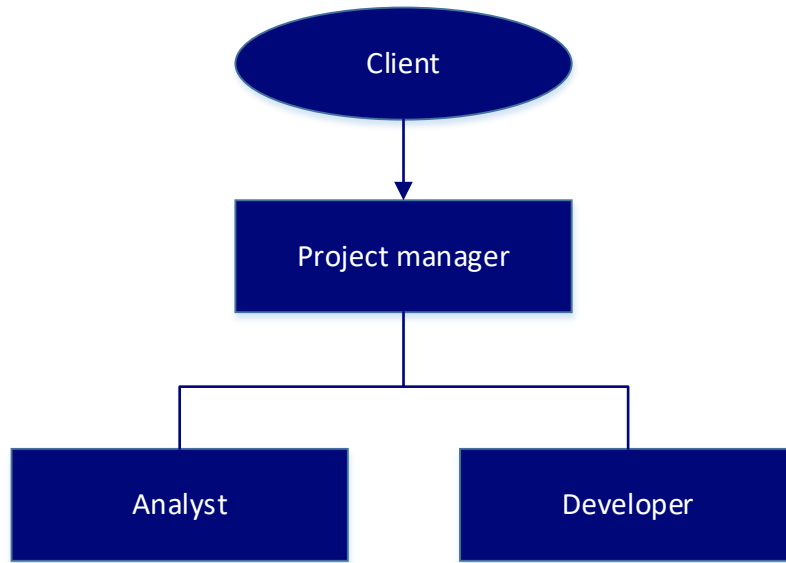


Ilustración 85. Roles

Client: represented by the tutor Miguel Ángel Patricio. He will take charge of transmitting his original idea to the project manager, and will verify the evolution of the project in each of the meetings after finishing each iteration.

Project manager: represented by Daniel Amigo. He will meet the client and will obtain the information about the request of the software. In addition, he is the person in charge of whom the period of the project is fulfilled in time, directing to the group of all time.

Analyst: represented by Daniel Amigo. He uses the information of the project manager to realize the specification of requirements, using different skills for it. Later, they realize all the necessary graphs, as well as the planning of the project.

Developer: represented by Daniel Amigo. He is the manager of developing the application. For it, he will use the resources explained later, and finally, will realize the tests of every set of cases of use.

10.3.3. Hardware resources

The development of this project has been realized entirely in the following computer:

Name	Lenovo Y50-70
Processor	Intel Core I7-4710HQ 2.5GHz. 4 cores, 8 threads
RAM Memory	16 GB
Graphics Processor	Nvidia 860M
Operative System	Windows 10 64 bits
Screen	15.6'' 1920 · 1080 pixels

Tabla 142. Computer resources

On the question of the tests will take place on the following smartphone:

Name	OnePlus One
Processor	Snapdragon 801
RAM Memory	3 GB
Graphics Processor	Adreno 330
Operative System	Android 6.1 Marshmallow
Screen	5.5'' 1920 · 1080 pixels

Tabla 143. Android resources

Finally, the device Arduino to use in the tests:

Name	NodeMCU
Processor	32 bits, 80MHz
RAM Memory	35K
Flash Memory	1Mb
Conectivity	WiFi, Micro-USB

Tabla 144. Arduino resources

10.3.4. Software resources

To execute this project, we will use the next software programs:

- Microsoft Word: To realize the present memory.
- Microsoft PowerPoint 2016: for the development of the presentation of the defense of the project.
- Microsoft Visio 2016: for the development of illustrations and graphs in the memory and in the presentation.
- Microsoft Project 2016: for the accomplishment of the graph GANTT.
- Android Studio: for the full development of the application.
- Arduino: to form the devices Arduino for the tests.
- Google Drive: to possess a backup of guarantees.
- TimeCamp: to measure the time of work of a precise way.
- Axure: to generate a prototype of the application.
- Adobe Acrobat DC Pro: For the final layout of the memory.
- Visual Paradigm: for the development of the graphs of activity related to the functionality of the application.

10.4. Planning and specification of requirements

At this point the requirements of the system are going to be specified. In this summary is going to be explained the procedure followed for the specification, as well as the translation of the table of sample.

The user's requirements start by being defined, to describe what actions can realize the user in the system. There are two types:

- Capacity requirements (RUC): they define the capacity of the application, always from the perspective of the functionality that the user can do.
- Restriction requirements (RUR): this type of requirements specifies the limitations of the system to realize the functions. They use to answer to the question: how does it do it?

Once with the list of user's requirements finished and validated, we will define the list of requirements software. This one is a refined version of the user's requirements, realized with language technical and orientated from the point of view of the system, so that the programmers understand and form perfectly the aim of the requirement in the system. There are two types too:

- Functionals (RSF): they define the behavior of the software to realizing and his functionalities.
- Non-functionals (RSNF): they focus on the design or the implementation of the system.

All the requirements follow this structure, of the IEEE 830:

Requirement identifier	It identifies in a univocal way to every requirement, by means of an acronym and a number X. The acronyms are: <ul style="list-style-type: none"> • RUC.X: User capacity requeriment. • RUR.X: User restriction requeriment. • RF.X: functional requirement. • RNF.X: non-functional requirement.
Requirement name	It indicates brief the content of the requirement.
Type	It indicates if it is a requirement or a restriction.
Source of the requirement	Indicates the one who has arranged the above-mentioned functionality. Client / Analyst
Priority of the requirement	It establishes the priority of carrying out the above-mentioned requirement inside the project. It takes the following values: Essential / Wished / Optional
Description of the requirement	It contains a detailed description on the requirement

Tabla 145. Example requirement

10.5. Prototyping

Once the analyst had the information obtained during the first meeting by the project manager with the client, he concluded realizing another skill of elicitation, realizing some prototypes.

These prototypes have been created to submit diverse users to different meetings of observation in a controlled environment, where the users interact with the prototypes. Thanks to these meetings, the analysts can catch the nature of the tasks, and fits the list of user's requirements.

10.6. Use cases

In this paragraph, there are exposed the use cases realized during the project. These (in the extended version) are exposed in expanded format, indicating this way the typical course of events and the possible alternative courses.

The use cases in expanded format will continue this structure:

Identifier	UC_XX, been XX the number of the use case.	
Name	Name of the use case.	
Actors	The actors involved in this use case.	
Priority	The priority of this use case.	
Source	Indicates the one who has arranged the above-mentioned functionality. Client / Analyst	
Pre-conditions	Pre-conditions required by the system.	
Post-conditions	Post-conditions required by the system.	
Description	The explanation of this use case.	
typical course of events		
	Actor	Sistema
	Action realized by the user	Action realized by the system
	They are numbered following the course of events	
Alternative course of events		
	Every alternative action is specified.	

Tabla 146. Use cases requirements

After all the use cases, a Traceability matrix links the use cases with the functional requirements.

10.7. Prioritization

In this paragraph, we define which cases of use realize in each cycle of development of the project. These will decide by means of a weighting of significant characteristics of the project. These characteristics are:

Characteristic	Weighting
On having realized the case of use, I will have a better comprehension of the design	0.2
Use of not known technology	0.15
Difficulty of development	0.15
Importance in the project	0.5

Tabla 147. Prioritization characteristics

Once we concrete the characteristics to valuing, and his weighting, we need to value the influence of each case of use with each of the characteristics, and obtain the sum of the total.

And with these results, we need to try to realize cycles of development in which each one guarantees on his end a minimal viable product that we can show to the client.

Finally, after all the calculations, we decided to realize three cycles of development.

10.8. Planning and budget

In this paragraph, there is attached the planning and the final budget of the project, to detail the resources used as exactly as is possible, both economics and temporary.

10.8.1. Planning

The used planning is ruled by the prioritization of cases of use realized, as well as the used methodology. I also want to remind that the times showed are as precise as possible because it has been parametrized by the application TimeCamp.

This project began on January 23 and ended on June 21 of the same year. They have worked a total of 260 hours on the project, distributed in 108 days of work, of which 80 of them were overcoming three hours dedicated to the project. The average of hours per day went of approximately three hours and a half to the day.

The graph Gantt, where each of the tasks can observe the real time that they have needed, is in the original version.

10.8.2. Budget

Later, with the resources used (indicated in the paragraph Organization of the project), and the planning, we show the calculation of the final budget of the project.

In this paragraph, there is calculated the final budget of the development of the application. For it, the resources observe hardware and software used (detailed previously), and then prices are established.

As for the human resources, on having known the number of hours dedicated to every role, a calculation of the minimum wage is realized of each one of the roles according to the Spanish laws, but to the being this slightly ancient law, the wages increases lightly by the client.

In addition, we calculate the indirect costs of light, water, transport and Internet to carry out the project.

Once calculated the costs of every type of resource, we show a summary with the final costs of the project.

Concept	Cost (euros)
Human resources	9616,95
Hardware resources	106,25
Software resources	599,65
Indirect costs	575
TOTAL WITHOUT VAT	10897,85
TOTAL WITH VAT	13186,40

Tabla 148. Budget summary

The final cost of the project is **Thirteen thousand, one hundred and eighty-six euros with forty centimes.**

10.9. Application design

In this paragraph, the design of the application will be described in a detailed way. In addition, there will be defined in a simple way the design and functioning of the used database.

10.9.1. Architecture design

The application follows an architecture of design based on the Model-View-Controller. This one separates the different aspects of the application in three layers, separating the information of the user of the interface, and the interface of the logic of the application.

In our application, the model will interact of the following way:

- Model: all the information defined by the system Firebase.
- View: it consists of all the files XML that define the style of the application.
- Controller: this layer consists of all the Java classes realized during the project. It's relevant to indicate that the sub-system Thinger belongs to this layer, since it is triggered by requests that are realized from the classes JAVA.

Graphically, this architecture can be summarized in:

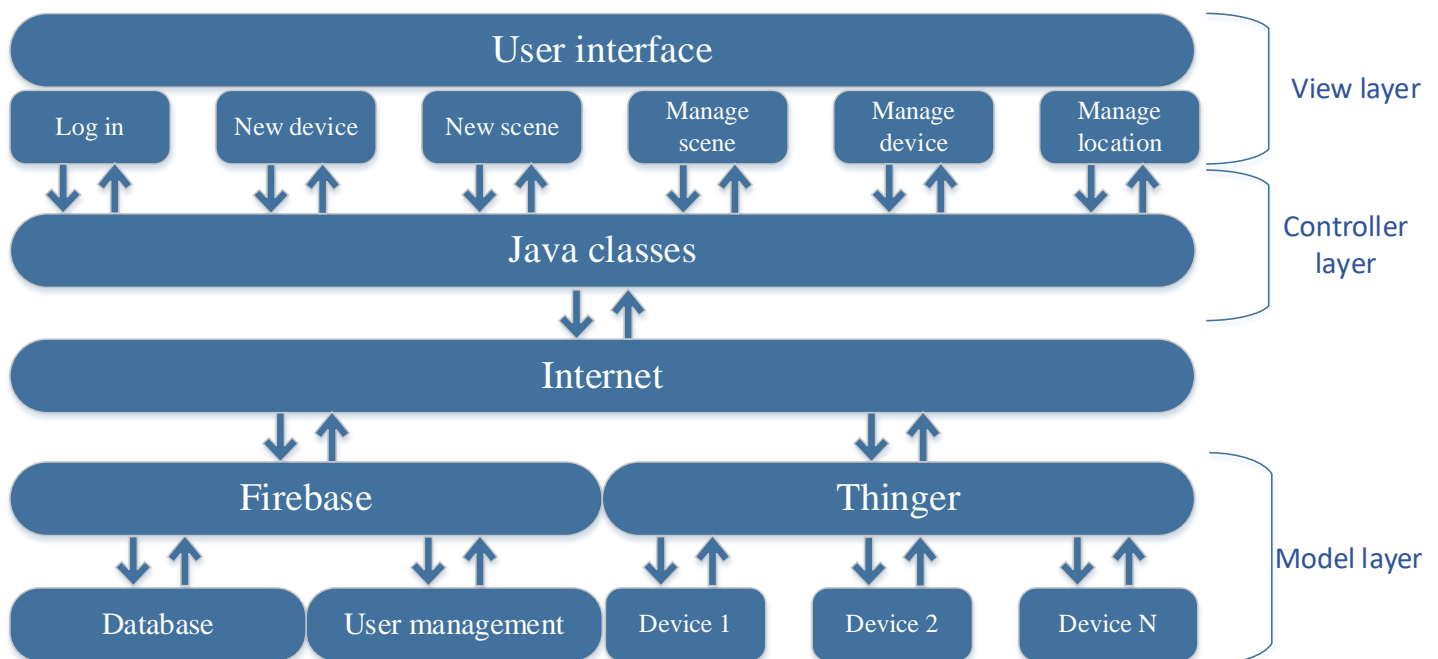


Ilustración 86. Architecture design

10.9.2. Activity diagrams

In this paragraph, we are going to present the graphs of activity relative to the flows of every action to be carried out by the user.

In this way, the flow realized by the application is understood perfectly, in a graphical way. These graphs are based on the use cases realized previously, realizing only those who possess functionality on the part by the user.

These diagrams have been made with Visual Paradigm, using the components of the standard UML.

10.9.3. Design of the database

In this paragraph, we are going to specify the form the information is stored in the database of Firebase.

This NoSQL database structures like an JSON tree, generating different nodes (branches) inside the structure of the tree. Being an online database, the paradigm used for his design has sought to give priority to the minor consumption of Internet. For it, it has been decided to create a structure of information sheltered for every user. With this paradigm, we have two clear advantages:

1. We control what information every user can get, with some rules that allow the access to every user only at his entry in the database.
2. To facilitate the obtaining of information in the programming of the application, since the requests are asynchronous, and to more requests, major complexity.

In the original version, we have a diagram of the structure of the database.

10.10. Tests

In this paragraph, are going to be detailed the tests to realize for each use case. These tests defined by the analyst, and carried out by the programmer, will reveal the quality of the realized product. There are going to be realized tests of several types:

- Hardware: verifies the correct capacities of the testing device, from a user's sight.
- Flow: verifies the correct functioning of the graphs of activity realized. These tests will be realized following the use cases of the application. To indicate that, when decisions should take place on the part of the user, several tests will be realized to contemplate the course of every use case.

The structure of the documentation of the tests is the following one:

Identifier	PX.YY, been X the type of the test, and YY the number. <ul style="list-style-type: none"> • H – Hardware • F – Flow
Description	It contains a detailed description of the test
Initial state	There is specified the initial state of the things to test.
Final state	There is specified the final state of the things to test.
Test procedure	The steps are detailed to continuing to carry out the test.
Test results	is indicated if the test was successful or otherwise, the difference is specified by the awaited result.

Tabla 149. Test example

10.11. Conclusions

In this paragraph, there will be deepened on the conclusions those who have come near after the ending of the project. It will be begun by the conclusions to technical level on the project, and later the conclusions of the only member of development.

10.11.1. Conclusions to the project level

Along the development there have been covered the different aims fixed for the beginning of the project, which we revise to value them finally:

- ✓ It was intended to analyze the needs that there must have a home automated current management tool. This paragraph was settled after realizing the user's requirements, where the analyst defined the needs of the application, validated by a selection of possible users trying a prototype, and reached with the later use cases.
- ✓ It was intended to design the system architecture to develop. This aim is settled thanks to the application design realized by the analyst, in which he defined the architecture to helping for the correct functioning of the application. In addition, thanks to the activity graphs, the functioning was detailed in a clear and simple way of each one of the functionalities to realizing for the user, facilitating the later implementation.
- ✓ Finally, it was intended to develop the application. The programmer completed this arduous task alone, with an invaluable effort and a final quality more that worthy, fulfilling perfectly all the tests defined by the analyst.

10.11.2. Conclusions to the personal level

Personally, this project has made me value more the quantity of applications that we consume day after day, and that make our life so easily. It is a shame that scarcely values the arduous work that exists behind them.

As for the project, it has been satisfactory to see how it was growing and growing once it was begun with the development, to see as all the effort of one month, abstract, of which you have references nor guarantees of which it goes work, it ends up by working and lightening the work that without a good planning had been impossible.

This one has been my first serious Android project, and has been a personal very satisfying challenge like I have indicated before. Whenever I manage to implement a new functionality, that after days of work you notice a negligible mistake that was loading the whole execution...

To know that this application is useful, that someone if he has these needs can have it and to use it without any type of problem or modification, it's comforting.

Finally, probably the most important thing, the capacities which I have a today are infinitely superior to those of the 1st of the project. Development of an Android application from zero, takes a planning a day, to generate an immense quantity of graphs and coherent requirements, and everything for me itself, without help and all made good, it seems to me that I work hard has borne fruit.

10.12. Future lines

The developed application fulfills fully a very concrete function, for whom was designed, but always it can be improved and implement on it improvements that grant a major functionality.

With this development, there arose diverse ideas that didn't end being implemented, due to a lack of time or for some simple technical disabilities to be carried out. Some of them are:

- To programme actions, or sets of actions to realize in a concrete moment, already be under a temporary need, like, for example, to turn on the lights at 12 p.m.; or thanks to the parameters of the devices, like, for example, the blinds go down if the temperature rises of 30 degrees.
- To allow access to thirds control the home automated devices. In the end, it's general to live in company and to have to link the devices every person with his user is not the best option. This option in addition might spread enough, allowing to change the degree of use of the devices in certain conditions, notifications of the use to the owner of the devices, to rely on a record to observe the use, etc.

- To implement a functionality of adjustments, giving customization of the application. Night mode, accessibility systems, control of the size of the texts and the colors ...
- Port the application to other systems, for example, to a web.
- To allow the users to protect his account with safety systems, as fingerprint, iris scanner, etc.