

Trabajo de Fin de Grado

ChronoStreetTurist

Memoria



VNiVERSIDAD
DE SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Alumno

Guillermo Calvo Suarez

Tutores

Luís Martín Liras

Fernando Rodríguez

Sara Rodríguez González

Certificado de los tutores

D. Luís Martín Liras, D. Fernando Rodríguez, representantes de la empresa HP, y D^a. Sara Rodríguez González, profesora del Departamento de Informática de la Universidad de Salamanca.

HACEN CONSTAR:

Que el trabajo titulado "ChronoStreetTurist" ha sido realizado por D. Guillermo Calvo Suárez, con el número de documento 71037052B y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado de la Titulación Grado de Ingeniería Informática de la universidad de Salamanca.

Y para que así conste a todos los efectos oportunos.

RODRIGUEZ
GONZALEZ SARA -
70864126E

Firmado digitalmente por
RODRIGUEZ GONZALEZ SARA -
70864126E
Fecha: 2022.07.05 22:33:42
+02'00'

Fernando
Rodríguez
Vicente

Digitally signed
by Fernando
Rodríguez Vicente
Date: 2022.07.06
09:37:10 +02'00'

Luis
Francisco
Martin Liras

Digitally signed by Luis
Francisco Martin Liras
DN: cn=Luis Francisco
Martin Liras, o, ou,
email=luis.martin.liras@h
p.com, c=ES
Date: 2022.07.06 09:30:13
+02'00'

En Salamanca, a 25 de junio de 2022.

Índice

0.	Resumen.....	11
0.1.	Palabras Clave	13
0.	Summary	14
1.	Introducción	16
2.	Objetivos del sistema	19
2.1.	Objetivos Funcionales	19
2.2.	Objetivos no funcionales.....	21
2.3.	Objetivos personales.....	22
3.	Trabajos relacionados	23
3.1.	Arqueoguide.....	23
3.2.	Realidad Aumentada Helsinki	24
3.3.	Aurasma	25
3.4.	ChronoTourist	25
4.	Conceptos teóricos.....	27
4.1.	Realidad Aumentada	27
4.1.1.	Historia de la Realidad Aumentada.....	27
4.1.2.	Niveles y Tipos de Realidad Aumentada	32
4.1.3.	Usos de Realidad Aumentada	34
4.2.	Tratamiento de la imagen	35
4.2.1.	Elementos de la imagen	35
4.2.2.	Espacios de color	36
4.3.	CMS	37
4.4.	APIS RESTful.....	38
4.5.	CLIENTE-SERVIDOR.....	39
4.6.	MVP.....	40
5.	Técnicas y Herramientas	41
5.1.	Herramientas case.....	41
5.1.1.	EzStimate.....	41
5.1.2.	Microsoft Office Project	41
5.1.3.	VisualParadigm.....	42
5.1.4.	Git	42
5.1.5.	GitLab	43
5.1.6.	REM	43
5.1.7.	Doxygen.....	43

5.1.8.	Procreate	44
5.1.9.	JSDoc	44
5.2.	Técnicas y herramientas desarrollo	45
5.2.1.	Vuforia	45
5.2.2.	Drive, API Google.....	47
5.2.3.	Firebase	47
5.2.4.	Visual Studio.....	49
5.2.5.	Unity	49
5.2.6.	UnitySimpleFileBrowser	50
5.2.7.	Node.JS.....	50
5.2.8.	NPM.....	50
5.2.9.	Ajax.....	51
5.2.10.	Xampp	51
5.2.11.	Apache Server	52
5.2.12.	Google Chrome y Microsoft Edge	53
5.2.13.	Bootstrap.....	53
5.2.13.1.	Jasny Bootstrap.....	53
5.2.13.1.	Bootstrap-material-design	54
5.2.14.	JQuery.....	54
5.2.15.	SweetAlert.....	55
5.2.16.	Perfect scrollbar jQuery	55
5.2.17.	DevTols	55
4.3	Lenguajes.....	56
4.3.1.	C#.....	56
4.3.2.	JavaScript.....	56
4.3.3.	Html	56
4.3.4.	PHP	57
4.3.5.	JSON	57
4.3.6.	CSS.....	57
6.	Aspectos relevantes del desarrollo del proyecto.....	58
6.1.	Metodología	58
6.2.1	SCRUM.....	59
6.2.	Estimación de esfuerzo	61
6.2.1.	Complejidad de los Casos de Uso.....	62
6.2.2.	Complejidad de Actores	63
6.2.3.	Factores.....	64

6.2.3.1. Factores de complejidad Técnica	64
6.2.3.2. Factores de complejidad del Entorno	65
6.3. Planificación temporal.....	67
6.3.1. Diagrama Gantt	70
6.4. Especificación de requisitos	72
6.4.1. Paquetes. RF.....	72
6.4.2. Actores. RF.....	73
6.4.3. Casos de uso. RF	74
6.4.4. Requisitos de Información	76
6.4.5. Requisitos No Funcionales	77
6.5. Análisis del sistema	78
6.5.1. Modelo de Dominio.....	78
6.5.2. Diagrama de paquetes	81
6.5.3. Diagramas de casos de uso	83
6.6. Diseño del sistema	84
6.7.1 Diseño arquitectónico	84
6.6.1.1. Subsistemas de diseño	86
6.6.1.2. Diagrama de despliegue	95
6.6.1.3. Diagramas de Secuencia.....	96
6.6.2. Diseño de los datos	97
6.6.2.1. Diagrama de base de datos.....	100
6.6.3. Diseño de la Interfaz.....	101
6.6.4. Diseño procedimental	103
6.7. Pruebas.....	106
6.7.1. Tipos Usuarios y Pruebas	106
6.7.1.1. Prueba registrar usuario nuevo e iniciar sesión.	107
6.7.1.2. Prueba recuperar contraseña.....	109
6.7.1.3. Prueba ver lista de Targets y búsqueda con filtro.....	111
6.7.1.4. Prueba subir nuevo Target.	112
6.7.1.5. Prueba ver Target y modificarlo.....	114
6.7.1.6. Prueba crear Usuario/Administrador.....	115
6.7.1.7. Prueba ver lista de usuarios e información de estos.	117
6.7.1.8. Prueba modificar Usuario.....	119
6.7.1.9. Prueba cambiar Rol.	121
6.7.1.10. Prueba borrar usuario.	122
6.7.1.11. Prueba Sacar Captura de Pantalla.	123

6.7.1.12.	Prueba completa subir Target y visualizarlo en la APP.	124
6.7.1.13.	Pruebas de seguridad.	127
7.	Sistema desarrollado.	128
7.1.	Aplicación Android.	128
1.1.	Escena inicio de sesión / registro.	128
1.2.	Escena Realidad Aumentada.	129
8.	Aplicación Web.	133
8.1.	Inicio de sesión/Registro.	133
3.2.	Home.	134
3.3.	Manejo de Targets.	135
1.4.	Usuarios. (Administrador)	136
1.5.	Perfil del Usuario.	137
1.6.	AboutUs.	138
9.	Conclusiones.	139
10.	Líneas de trabajo futuras.	141
9.	Bibliografía.	142

LISTA DE IMÁGENES

Ilustración 1. Archeoguide off [15].	23
Ilustración 2. Archeoguide on [15].	24
Ilustración 3. Aurasma [18].	25
Ilustración 4. ChronoTourist [9].	26
Ilustración 5. Ejemplo Realidad Aumentada [3].	27
Ilustración 6. Sensorama (patente) [9].	28
Ilustración 7. Boceto de la espada de Damocles [9].	29
Ilustración 8. Videoplace, [9].	29
Ilustración 9. Virtual Fixtures [9].	30
Ilustración 10. KARMA [9].	30
Ilustración 11. ARQuake, [9].	31
Ilustración 12. PokemonGO [9].	32
Ilustración 13. RA basada en marcadores.	32
Ilustración 14. RA sin marcadores.	33
Ilustración 15. RA a través de objetos.	33
Ilustración 16. RA basada en proyección [9].	33
Ilustración 17. Smart Terrain.	33
Ilustración 18. imagen con diferente número de Bits [9].	35
Ilustración 19. Modelo RGB.	36
Ilustración 20. Modelo HSV.	36
Ilustración 21. Modelo YcbCr.	36
Ilustración 22. API RestFul [14].	38
Ilustración 23. Cliente-Servidor.	39
Ilustración 24. MVP.	40
Ilustración 25. EzEstimate.	41
Ilustración 26. M. Office Project.	41
Ilustración 27. VisualParadigm.	42
Ilustración 28. Git.	42
Ilustración 29. GitLab.	43
Ilustración 30. Doxygen.	43
Ilustración 31. Procreate.	44
Ilustración 32. JSDoc.	44
Ilustración 33. Vuuforia.	45
Ilustración 34. Add Target.	46
Ilustración 35. Comparacion RA.	47
Ilustración 36. Drive.	47
Ilustración 37. Google Cloud Platform.	47
Ilustración 38. Firebase.	48
Ilustración 39. Tabla usuarios Firebase.	48
Ilustración 40. VisualStudio.	49
Ilustración 41. Unity.	49
Ilustración 42. nodejs.	50
Ilustración 43. NPM.	50
Ilustración 44. Ajax.	51
Ilustración 45. xampp.	51
Ilustración 46. interfaz xampp.	52

Ilustración 47. Apache.....	52
Ilustración 48. Navegadores.....	53
Ilustración 49. Bootstrap.....	53
Ilustración 50 menu navbar.....	53
Ilustración 51. navbar.....	53
Ilustración 52. jQuery.....	54
Ilustración 53. Ejemplo Ajax jQuery.....	54
Ilustración 54. sweetalert.....	55
Ilustración 55. alerta sweetalert.....	55
Ilustración 56. DevTools.....	55
Ilustración 57. C#.....	56
Ilustración 58. JavaScript.....	56
Ilustración 59. html.....	56
Ilustración 60.php.....	57
Ilustración 61. Json.....	57
Ilustración 62. css.....	57
Ilustración 63. Metodología agile.....	58
Ilustración 64. SCRUM.....	60
Ilustración 65. EzEstimate.....	66
Ilustración 66. planificación temporal Resumida.....	67
Ilustración 67. planificación temporal.....	68
Ilustración 68. Diagrama Gantt 1.....	70
Ilustración 69. Diagrama Gantt 2.....	71
Ilustración 70. Paquetes.....	72
Ilustración 71. Actores.....	73
Ilustración 72. Ejemplo tabla actores.....	73
Ilustración 73. Gestión de usuarios.....	74
Ilustración 74. Gestión de Targets.....	74
Ilustración 75. Gestión de Realidad Aumentada.....	75
Ilustración 76. Gestión de captura de pantalla.....	75
Ilustración 77. Ejemplo tabla casos de uso.....	75
Ilustración 78. Ejemplo tabla Requisito Información.....	76
Ilustración 79. Ejemplo tabla Requisito no Funcional.....	77
Ilustración 80. Modelo de Dominio.....	78
Ilustración 81. diagrama de paquetes reducido.....	81
Ilustración 82. Diagrama de paquetes extenso.....	82
Ilustración 83. Ejemplo caso de uso análisis.....	83
Ilustración 84. arquitectura unity.....	84
Ilustración 85. Arquitectura del sistema.....	85
Ilustración 86. Diagrama de paquetes.....	86
Ilustración 87. Paquete Views.....	87
Ilustración 88. Paquete GoogleClouds.....	87
Ilustración 89. Paquete Apis_VWS.....	88
Ilustración 90. Paquete componentes.....	89
Ilustración 91. Paquete Gestión y control de Targets.....	90
Ilustración 92. Paquete gestión y control de usuarios.....	91
Ilustración 93. Gestión Realidad Aumentada.....	92
Ilustración 94. Gestión de captura de pantalla.....	93

Ilustración 95.diagrama de paquetes completo	94
Ilustración 96. Diagrama de Despliegue.....	95
Ilustración 97. Ejemplo caso de uso diseño.	96
Ilustración 98. Esquema Bases de Datos.....	99
Ilustración 99. Diagrama de bases de datos.	100
Ilustración 100. mockup I.....	101
Ilustración 101. APIS.....	103
Ilustración 102. APIS.....	103
Ilustración 103. json_encode.	103
Ilustración 104. Emulador CloudFunctions.	105
Ilustración 105. Registro.....	107
Ilustración 106. Email confirmación correo	107
Ilustración 107.Inicio de sesión	108
Ilustración 108. Home usuario	108
Ilustración 109. Home administrador.	108
Ilustración 110. Recuperación de contraseña.....	109
Ilustración 111. Correo restablecimiento de contraseña.....	109
Ilustración 112. vista cambio de contraseña.....	110
Ilustración 113. Confirmación restablecimiento de contraseña.	110
Ilustración 114. Target Manager User.....	111
Ilustración 115. Target Manager Admin.	111
Ilustración 116. Ejemplo filtrado de Targets.	112
Ilustración 118. Formulario New Target.	113
Ilustración 117. Opciones avanzadas.	113
Ilustración 119. Target subido correctamente.....	113
Ilustración 120. Comprobación Target subido.....	113
Ilustración 121. Editar Target.	114
Ilustración 122. Ver Target.....	114
Ilustración 123. Modificar Target.	114
Ilustración 124. Resultado modificación Target.....	115
Ilustración 125. Target modificado.	115
Ilustración 126. Vista Users.	115
Ilustración 127. Create user	116
Ilustración 128. Respuesta creación usuario.....	116
Ilustración 129. lista de usuarios con el nuevo usuario.	116
Ilustración 130. Lista usuarios.....	117
Ilustración 131. información de otro usuario (admin).	117
Ilustración 132. User profile.....	120
Ilustración 133. Usuario modificado.	120
Ilustración 134. Userupdated.....	120
Ilustración 135. Cambio Rol	121
Ilustración 136. Rol cambiado.....	121
Ilustración 137. Info usuario a borrar	122
Ilustración 138. Confirmación borrar usuario	122
Ilustración 139. Respuesta borrar usuario	122
Ilustración 140. Lista sin el usuario borrado.	123
Ilustración 141.almacenamiento del móvil.....	123
Ilustración 142. Prueba visualizar monumento nuevo 1.	124

Ilustración 143 Target Pendiente	124
Ilustración 144. Activar Target.	124
Ilustración 145. Target Activo	124
Ilustración 146.inicio sesión unity	125
Ilustración 147. Vista opciones unity	125
Ilustración 148. Realidad aumentada off	125
Ilustración 149. Realidad Aumentada on.....	126
Ilustración 151. información monumento	126
Ilustración 150. Wiki del monumento.....	126
Ilustración 152. seguridad sesión activa	127
Ilustración 153. Aviso actividad ilícita	127
Ilustración 154. Vista Registro.....	129
Ilustración 155. Vista inicio Sesión.....	129
Ilustración 156. Vista Panel Principal	130
Ilustración 157. Vista Inf sobre la app	131
Ilustración 158. Vista Realidad Aumentada Desactivada.....	131
Ilustración 159. Vista Realidad Aumentada Activada	132
Ilustración 160. Vista información del monumento	132
Ilustración 161. Inicio Sesión Web	133
Ilustración 162. Home Administrador	134
Ilustración 163. Home Usuario.....	134
Ilustración 164. Administrador Targets Administrador.	135
Ilustración 165. Administrador Targets Usuario	135
Ilustración 166. Users.....	136
Ilustración 167. UserProfile.....	137
Ilustración 168. AboutUs.....	138

Lista de tablas

Tabla 1. Gestión y control de usuarios.....	19
Tabla 2. Gestión y control de Targets.....	20
Tabla 3. Gestión de Realidad Aumentada.....	20
Tabla 4. Gestión de capturas de pantalla.....	20
Tabla 5. complejidad según las transacciones.....	62
Tabla 6. Pesos Casos de Uso.....	62
Tabla 7. Complejidad de Actores.....	63
Tabla 8. Pesos Actores.....	63
Tabla 9. Factores de complejidad Técnica.....	64
Tabla 10. Factores de complejidad del Entorno.....	65

0. Resumen

Poco a poco las tecnologías van avanzando hasta el punto de que en los objetivos de las grandes empresas se encuentra la creación de gafas o lentes inteligentes que nos hagan la vida mucho más fácil y que pasen a ser esenciales en nuestras vidas.

Estas lentes nos darán la capacidad de ver objetos o widgets virtuales sobre la propia realidad. A esto es a lo que llamamos Realidad Aumentada y cuenta con un amplio abanico de posibilidades de desarrollo, desde aspectos puramente matemáticos, como podrían ser cálculos de medidas en una construcción, hasta aspectos médicos, como podría ser la indicación de qué estructuras seccionar en una intervención quirúrgica.

La Realidad Aumentada es el conjunto de tecnologías que hacen posible la interacción entre los objetos del mundo físico y los objetos del mundo virtual a través de dispositivos tecnológicos como las cámaras de los smartphones, tablets, ordenadores, etc.

Como puede deducirse, no es lo mismo que la Realidad Virtual, ya que, a pesar de estar entrelazadas, plantean revoluciones diferentes. Mientras que la Realidad Virtual permite crear un mundo virtual desde cero, con todo lo que queramos, pero un mundo fantástico, la Realidad Aumentada agrega elementos virtuales (información adicional en forma de gráficos o imágenes) a nuestro entorno real.

Gracias al desarrollo de la tecnología (sobre todo móvil), la Realidad Aumentada ha evolucionado en todos los ámbitos, desde los videojuegos hasta el turismo, permitiéndonos así recorrer la ciudad y conocer la historia de la misma sin la necesidad de un guía.

ChronoTulist es una aplicación móvil de carácter turístico y educativo que nos permite visualizar la evolución de los núcleos urbanos y comprobar *in-situ* cómo les ha afectado el paso del tiempo. Para ello se realiza un proceso de selección de imágenes de los archivos históricos de los distintos monumentos de la ciudad y, mediante el uso de herramientas de Realidad Aumentada y bases de datos relacionales, estas imágenes se superpondrán a la real de la cámara cuando nos encontremos frente a dichos monumentos.

La aplicación también cuenta con una interfaz web, donde:

- Los usuarios tendrán la posibilidad de llevar un control de los Targets publicados, creación, modificación y eliminación de Targets, un control del perfil de usuario y algunas funcionalidades menos prioritarias.
- Los administradores tendrán la posibilidad de llevar el control de los Targets de todos usuarios, creación, modificación y eliminación, pudiéndolos filtrar, al igual que los usuarios, y activar en caso de que el Target publicado sea aprobado por el administrador. También llevarán el control de los usuarios, pudiendo hacer creación, eliminación o modificación de estos.
- Funcionalidades menos prioritarias, como, en los Administradores, las tareas que deben realizar, las fechas límite de dichas tareas, listado de los usuarios sospechosos... o en los usuarios, explicación de la Realidad Aumentada, realidad virtual, uso de la app...

Separar en aplicación móvil y web nos brinda simplicidad y minimalismo a la hora del manejo de los usuarios.

Cuando se habla de Target, se refiere a toda la información sobre el monumento y su configuración de Realidad Aumentada. Cada Target describe un monumento, es decir, contiene la imagen que comparará con la imagen que recibe de la cámara, la imagen a superponer sobre esta anterior, y parámetros de configuración como escala, posición, descripción....

0.1. Palabras Clave

Realidad Aumentada, Target, Base de datos relacional, Realidad Virtual, APP, turismo, guía turística.

0. Summary

Gradually, technology has been developed to the extent that big corporations goals are in the production of smart glasses that become essentials and make our lives easier.

These lenses will give us the capability to appreciate objects or virtual widgets about the reality itself. This is what can we call Augmented Reality and it is supported by a wide range of possibilities for development, from purely mathematical aspects, such as specific measures adapted to a construction, to medical issues, as it can be the indication of which structures can be selected in a surgery.

Augmented Reality could be described as the set of technologies that make possible the interaction between the physical and virtual world objects by technological devices such as smartphones cameras, digital pads, laptops, etc.

As it can be seen, it is not the same as Virtual Reality, in spite of being intertwined, resided different revolutions.

While Virtual Reality enables to create a virtual world from scratch, as much as desired, but a fantastic world, Augmented Reality adds virtual elements (additional information shaped like graphics and images) to our actual environment.

Thanks to the development of technologies (specially mobile), Augmented Reality has evolved in all areas, from video games to tourism, allowing us to travel through a city and discovering its history without the need of a tourist guide.

ChronoTulist is a mobile app with touristic and educational purposes that allows us to visualize the evolution of the urban centers and to check in situ how they have been affected by the passage of time. To do so, a process of selection of images from the historical archives of the different monuments of the city is carried out and, through the use of Augmented Reality tools and relational databases, these images will be superimposed on the real one of the camera when we are in front of these monuments.

The app provides an interface web, where:

- Users will have the possibility to keep track of published Targets, creation, modification and elimination of Targets, user profile control and some lower priority functionalities.
- Administrators will have the possibility to control the Targets of all users, creation, modification and deletion, being able to filter them, as well as users, and activate them in case the published Target is approved by the administrator. They will also have control of users, being able to create, delete or modify them.
- Lower priority functionalities, such as, for Administrators, the tasks to be performed, the deadlines for these tasks, list of suspicious users... or for users, explanation of augmented reality, virtual reality, use of the app...

Separating into mobile and web application gives us simplicity and minimalism when it comes to user management.

When we talk about Target, we refer to all the information about the monument and its Augmented Reality configuration. Each Target describes a monument, i.e. it contains the image that will be compared with the image received from the camera, the image to be superimposed on the previous one, and configuration parameters such as scale, position, description.....

1. Introducción

Este documento recoge la memoria del Trabajo De Fin de Grado de Ingeniería Informática realizado por el alumno Guillermo Calvo Suárez, en colaboración con la empresa HP, tras la superación de un proceso de selección. El proyecto tiene como título “ChronoTulist”, los tutores de este trabajo han sido Sara Rodríguez González, de la Universidad de Salamanca, Luis Martín Liras y Fernando Rodríguez, de la empresa HP.

Poco a poco las tecnologías van avanzando, hasta el punto de que entre los objetivos de las grandes empresas se encuentra la creación de gafas o lentes inteligentes que nos hagan la vida mucho más fácil, y que pasen a ser esenciales en nuestras vidas.

Estas lentes nos darán la capacidad de ver objetos o widgets virtuales sobre la propia realidad. A esto es a lo que llamamos Realidad Aumentada, y cuenta con un amplio abanico de posibilidades de desarrollo, desde aspectos puramente matemáticos, como podrían ser cálculos de medidas en una construcción, hasta aspectos médicos, como podría ser la indicación de qué estructuras seccionar en una intervención quirúrgica.

La Realidad Aumentada es el conjunto de tecnologías que hacen posible la interacción entre los objetos del mundo físico y los objetos del mundo virtual a través de dispositivos tecnológicos como las cámaras de los smartphones, tablets, ordenadores, etc.

En la actualidad, los guías turísticos han sufrido mucho por la evolución en el mundo de internet, y, para acabar con ellos, la pandemia. Las personas ya no necesitan un guía que les vaya informando cuando toda esa información, incluso más, la tienen en la palma de la mano.

Lo mismo ha ocurrido con los taxistas, la población ha evolucionado y se ha adaptado al uso de nuevas formas de conseguir u ofrecer el mismo producto (Úber).

El objetivo principal del proyecto es la creación de una aplicación que permita la visualización de la evolución de los núcleos urbanos y la comprobación *in-situ* de cómo les ha afectado el paso del tiempo, simulando un guía turístico, mediante el uso de herramientas de Realidad Aumentada y bases de datos relacionales. Para ello también se cuenta con una interfaz web, donde los usuarios van a poder contribuir en el desarrollo de la aplicación, creando, modificando y eliminando monumentos, y donde los administradores podrán llevar el control de los usuarios y de los monumentos que publiquen estos últimos.

Con la aplicación se pretende conseguir esa evolución de la población, de manera que el producto que se está vendiendo sea menos costoso en muchos aspectos, sobre todo monetaria y físicamente. La aplicación da también la posibilidad de una fácil adaptación, ya que no es necesaria una persona físicamente, es posible acceder a la información en cualquier parte del mundo y a cualquier hora, sin coste monetario y con la capacidad de visualizar de manera histórica, desde hace siglos, la realidad que vemos, todo ello gracias a internet y a nuevas tecnologías que van evolucionando día a día, como la Realidad Aumentada.

El presente documento pretende describir los aspectos más importantes del proyecto, este está compuesto por:

- **Introducción:** Breve presentación del tema que se aborda en el proyecto para poner al lector en situación.

- **Objetivos:** Explicación precisa y concisa de cuáles son los objetivos que se pretenden alcanzar con la realización del proyecto.
- **Conceptos Teóricos:** Conceptos que se han necesitado para la comprensión y desarrollo de una determinada materia necesaria para la realización del proyecto.
- **Técnicas y herramientas:** Técnicas y herramientas que han sido utilizadas para el desarrollo del proyecto.
- **Aspectos relevantes del desarrollo:** Aspectos más interesantes del desarrollo del proyecto.
- **Trabajos relacionados.**
- **Conclusiones y líneas de trabajo futuras:** Conclusiones obtenidas tras la realización del proyecto y posibles ampliaciones y desarrollos de este.

En la entrega del proyecto se adjunta este documento, el código de desarrollador y la documentación técnica. Esta última se compone de:

Anexo I. Plan de proyecto software

En este anexo se recoge toda la información relacionada con la estimación del esfuerzo y la planificación temporal del mismo, para así alcanzar los objetivos marcados. Esta planificación temporal se presenta de manera muy detallada, indicando las tareas y los tiempos de ejecución de estas en días.

Anexo II. Especificación de requisitos del software

Recoge todos los requisitos que debe cumplir el sistema, para conseguir los objetivos marcados. Dentro del mismo nos podremos encontrar:

- Objetivos del sistema.
- Requisitos Funcionales.
 - Diagramas de paquetes.
 - Definición de actores.
 - Casos de uso.
- Requisitos de Información.
- Requisitos no Funcionales.

Este anexo define las directrices iniciales que va a seguir el proyecto.

Anexo II-b. Análisis de Requisitos

Recoge la parte de análisis del sistema software, consiguiendo poco a poco una visión más clara del proyecto, facilitando así la comprensión de este.

Este compuesto por:

- Modelo de dominio.
- Paquetes de análisis.
- Diagramas de secuencia de análisis.

Anexo III. Especificación del diseño

En este anexo nos adentramos ya en el desarrollo del sistema software, es decir, le vamos dando la forma que queremos que tenga finalmente. Esta parte es muy importante, se van a tomar decisiones sobre la arquitectura, los datos, la interfaz y los detalles procedimentales del software.

Anexo IV. Documentación técnica de programación

Recoge la documentación técnica de los diferentes scripts que forman el programa, este hecho para que los programadores entiendan la estructura del proyecto.

Anexo V. Manual de usuario

Recoge todos los tutoriales, manuales y guías de usuario que sean necesarios para que un usuario realice un correcto manejo de la aplicación.

2. Objetivos del sistema

El objetivo principal del proyecto es la creación de una aplicación que permita la visualización de la evolución de los núcleos urbanos y la comprobación *in-situ* de cómo les ha afectado el paso del tiempo, simulando un guía turístico, mediante el uso de herramientas de Realidad Aumentada y bases de datos relacionales. Como objetivo también se encuentra que los usuarios puedan contribuir en el desarrollo del sistema, creando, modificando y eliminando monumentos y que los administradores puedan llevar el control de los usuarios y de los monumentos que publiquen estos últimos, todo esto gracias a una interfaz web.

2.1. Objetivos Funcionales

Respecto a los objetivos funcionales las siguientes tablas recogen los identificados, incluyendo la gestión y control de usuarios, de los Targets, de Realidad Aumentada y de Captura de pantalla.

GESTIÓN Y CONTROL DE USUARIOS

OBJ-0001 Gestión y Control de Usuarios	
Autores	<ul style="list-style-type: none"> • <u>¡Error! No se encuentra el origen de la referencia.</u>
Fuentes	<ul style="list-style-type: none"> • <u>¡Error! No se encuentra el origen de la referencia..</u> • <u>¡Error! No se encuentra el origen de la referencia.</u> • <u>¡Error! No se encuentra el origen de la referencia.</u>
Descripción	<p>El sistema deberá ser capaz de manejar diferentes roles de usuarios:</p> <ul style="list-style-type: none"> - Usuario no registrado: Permite registrarse mediante un formulario de registro. - Usuario registrado: Permite visualizar y modificar la información generada por el sistema del propio usuario. - Administrador: Permite visualizar y modificar la información generada por el sistema del propio administrador y visualizar y modificar cierta información, aceptada por la ley de protección de datos de los usuarios. También pueden crear nuevos usuarios/administradores. <p>Para acceder al sistema es necesario autenticarse, así como verificar el correo a la hora del registro.</p> <p>Deberá ser capaz de cerrar sesión cualquiera de los usuarios, y en caso de que se pierdan las credenciales de la caché web cerrar sesión. Debe regular el acceso de un tipo de usuario a las interfaces de otro tipo diferente.</p>

Tabla 1. Gestión y control de usuarios.

GESTION Y CONTROL DE TARGETS

OBJ-0002 Gestión y Control de Targets	
Autores	<ul style="list-style-type: none"> • <u>¡Error! No se encuentra el origen de la referencia.</u>
Fuentes	<ul style="list-style-type: none"> • <u>¡Error! No se encuentra el origen de la referencia..</u>

	<ul style="list-style-type: none"> • <u>¡Error! No se encuentra el origen de la referencia.</u> • <u>¡Error! No se encuentra el origen de la referencia.</u>
Descripción	<p>El sistema deberá ser capaz de manejar los Targets subidos a la nube de Vuforia, tanto creación, como modificación, como eliminación. Y por tanto deberá ser capaz del correcto manejo de todos los archivos y datos asociados a estos y su distribución.</p> <p>El sistema deberá distinguir correctamente entre usuario y administrador y sus posibles acciones.</p> <p>El sistema también deberá ser capaz del correcto filtrado por diferentes atributos como Nombre, Ciudad, Correo, Activos...</p>

Tabla 2. Gestión y control de Targets.

GESTION DE REALIDAD AUMENTADA

OBJ-0003 Gestión de Realidad Aumentada	
Autores	<ul style="list-style-type: none"> • <u>¡Error! No se encuentra el origen de la referencia.</u>
Fuentes	<ul style="list-style-type: none"> • <u>¡Error! No se encuentra el origen de la referencia..</u> • <u>¡Error! No se encuentra el origen de la referencia.</u> • <u>¡Error! No se encuentra el origen de la referencia.</u>
Descripción	<p>El sistema deberá ser capaz de reconocer monumentos mediante la cámara del dispositivo y superponer imágenes correctamente sobre ellas, incluyendo también el correcto manejo de la información asociada al Target reconocido.</p>
Subobjetivos	<ul style="list-style-type: none"> • [OBJ-0004] Reconocimiento de monumentos: El sistema deberá ser capaz de comparar la realidad con los Targets almacenados en la nube encontrando similitudes mediante IA y así superponer las imágenes. • [OBJ-0005] superposición de monumentos: Una vez reconocido el monumento, el sistema deberá ser capaz de recoger la imagen a superponer de drive y la información asociada del servidor y colocarla con su correcta escala y posición.

Tabla 3. Gestión de Realidad Aumentada.

GESTION DE CAPTURAS DE PANTALLA

OBJ-0006 Gestión de Capturas de pantalla.	
Autores	Guillermo Calvo
Fuentes	<ul style="list-style-type: none"> • <u>¡Error! No se encuentra el origen de la referencia..</u> • <u>¡Error! No se encuentra el origen de la referencia.</u> • <u>¡Error! No se encuentra el origen de la referencia.</u>
Descripción	<p>El sistema deberá ser capaz de capturar la pantalla y guardarla en el almacenamiento del móvil.</p>

Tabla 4. Gestión de capturas de pantalla.

2.2. Objetivos no funcionales

En esta sección se detallarán los requisitos no funcionales, los cuales se encargan de buscar la calidad del sistema software:

- **Usabilidad:** el sistema deberá *proporcionar simplicidad y fácil manejo a todos los usuarios, para ellos facilitará una interfaz sencilla e intuitiva sin sobrecarga de información.*
- **Seguridad:** el sistema deberá *tener unos criterios de seguridad mínimos para no comprometer la información de los usuarios ni de los Targets.*
- **Portabilidad:** el sistema *deberá poder desplegarse en cualquier sistema operativo*
- **Compatibilidad:** el sistema *deberá poder correr en cualquier navegador.*
- **Rendimiento:** el sistema *deberá funcionar usando el menor consumo posible, es decir, el mínimo necesario.*
- **Almacenamiento:** el sistema *deberá almacenar la información del sistema de manera segura en bases de datos y accesible a tiempo real. Firestore, Drive y Servidor.*
- **Fiabilidad:** el sistema *deberá levantarse inmediatamente tras una caída para no perder el funcionamiento básico de la aplicación y la consistencia de los datos.*
- **Escalabilidad:** el sistema *deberá ser escalable para poder almacenar toda la cantidad de datos del sistema que pueda llegar a tener la aplicación, sabiendo que puede ser una aplicación a nivel internacional, y por tanto mucha cantidad de datos.*

2.3. Objetivos personales

Como objetivos personales teóricos en la creación de este proyecto, está la adquisición de conocimiento en bases de datos no SQL, aprendizaje de nuevas tecnologías puntas en el mercado actualmente, como nodeJS, Realidad Aumentada, Ajax, CMS..., aprendizaje de lenguajes de programación. Así como el perfeccionamiento de los conocimientos adquiridos tanto en la universidad de Salamanca, como en la universidad Complutense de Madrid.

Otro tipo de objetivo personal marcado es el aprendizaje de gestión de tiempo, es decir, como organizar un proyecto a gran escala y como organizar un equipo de trabajo, para realizar las entregas a tiempo.

Adicionalmente, como objetivo más personal que los demás, es la adquisición de la capacidad para desarrollar proyectos de grandes empresas como pueden ser HP, Google, IBM o Microsoft.

3. Trabajos relacionados

En esta sección de la memoria se muestran trabajos que en aspectos teóricos y técnicos son parecidos al proyecto desarrollado.

3.1. Archeoguide

Archeoguide [15] es un proyecto que consigue que el usuario que use la aplicación logre visualizar, mediante la fusión del mundo real con el mundo virtual, reconstrucciones 3D de lugares en ruinas. Para ello, la técnica usada por la aplicación se sustenta en el seguimiento de la posición basado en técnicas de imágenes como por ejemplo el registro de éstas, que consiste en una serie de imágenes de los lugares guardados en una base de datos que sirven como «áreas de referencia». A continuación, se utiliza un método que compara las similitudes entre las imágenes que recoge la cámara y las que se encuentran en la base de datos cuyas coordenadas asociadas están cerca de las proporcionadas por el GPS del dispositivo móvil, si las dos imágenes coinciden en un 30-40 % se procederá al proceso de superposición de la imagen en la base de datos sobre la que proporciona el dispositivo. Las técnicas que utiliza el proceso de correlación entre las dos imágenes son: translación, rotación de las imágenes para que tengan la misma perspectiva y escalado de la imagen de la base de datos.

Video de demostración: [Projeto Archeoguide \(2000-2002\) - Vencedor de Premios LAVAL Virtual 2002 - YouTube](#)



Ilustración 1. Archeoguide off [15].

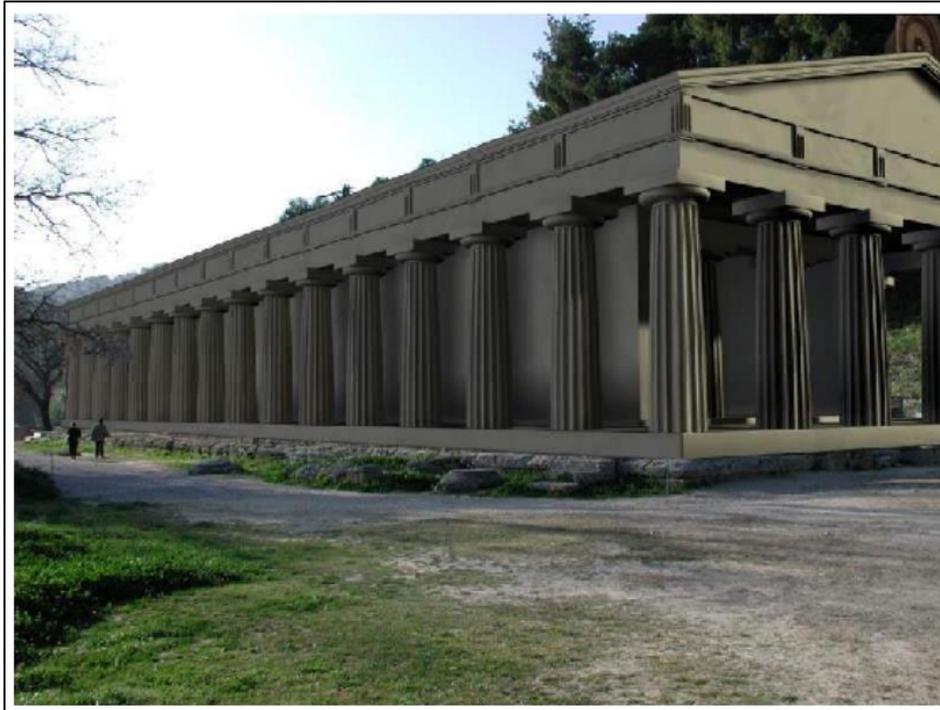


Ilustración 2. Arqueoguide on [15].

3.2. Realidad Aumentada Helsinki

Esta aplicación hace uso de la Realidad Aumentada sin marcadores para mostrar imágenes históricas de monumentos de la ciudad de Helsinki [16]. El funcionamiento de la aplicación se basa en dos técnicas de reconocimiento visual:

- **Nube de puntos:** esta técnica consiste en realizar una reconstrucción 3D del monumento mediante la realización de un gran número de fotos de este en distintas condiciones meteorológicas o con distinta iluminación y así es posible colocar manualmente las imágenes en una posición y perspectiva determinada. Con esta técnica es posible estimar de una manera automática las ubicaciones de la cámara y las posiciones 3D de las características naturales para hacer la reconstrucción 3D dispersa del entorno.
- **Imágenes en 2D como marcadores:** al igual que en ChronoStreetTulist, esta aplicación también hace uso de imágenes de los monumentos para que, al enfocar a ese monumento, la imagen histórica se superponga. Además, mediante el uso de los sensores del GPS del propio teléfono móvil, esta aplicación proporciona un seguimiento 3D más rudimentario en el caso de que exista un fallo en los métodos de seguimiento visual.

Video de demostración:

https://www.youtube.com/watch?v=5ZLNjnnXinA&ab_channel=VTTAugmentedReality

3.3. Aurasma

Esta aplicación de Android e IOS [17](retirada de la Google Play Store y IOS Store el 1 de julio de 2019) orientada para la educación que permite combinar objetos, imágenes y lugares del mundo real con elementos digitales (animaciones, vídeos u otras imágenes) para que, cuando el usuario enfoque el objeto del mundo real, el elemento digital salga, permitiendo así interactuar con el entorno [19]:

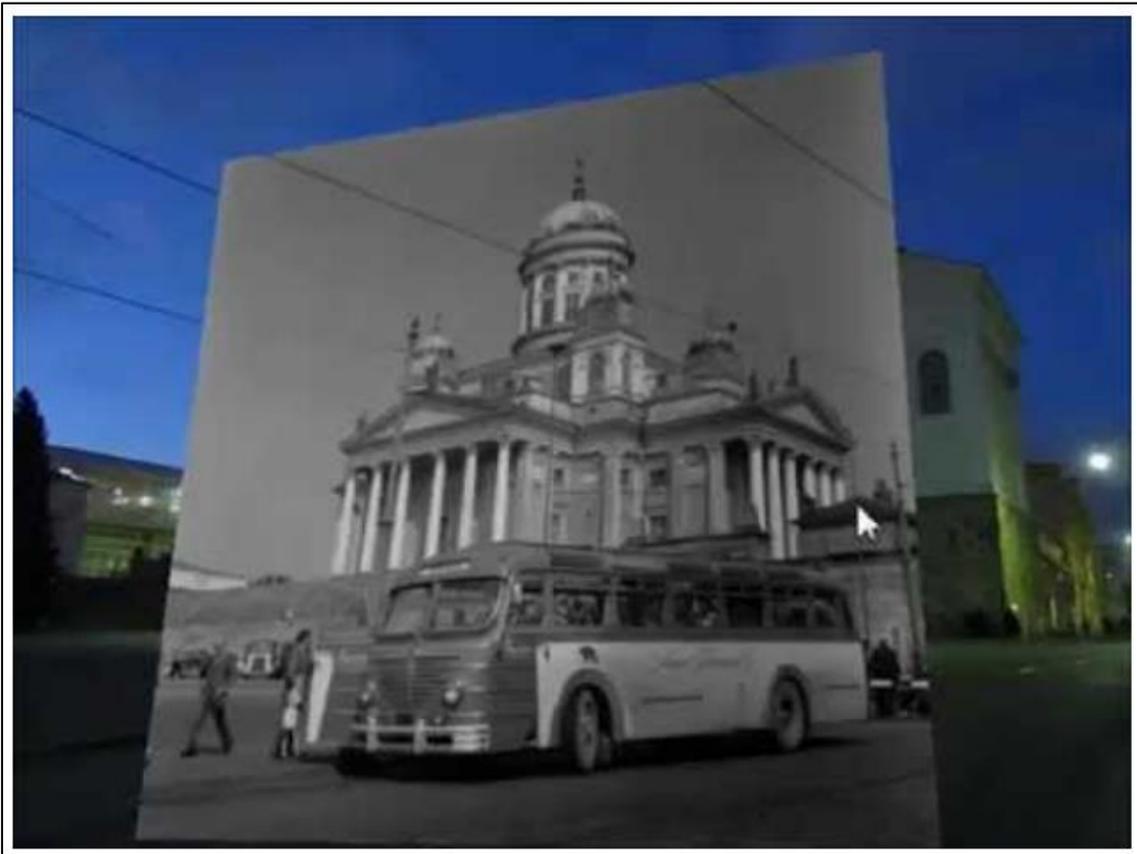


Ilustración 3. Aurasma [18].

Video de demostración: [Aurasma Demo - YouTube](#)

3.4. ChronoTourist

ChronoTourist [9] es una aplicación móvil que permite, mediante Realidad Aumentada, mostrar imágenes antiguas en la pantalla del móvil al apuntar con la cámara hacia un monumento del que se disponen imágenes antiguas. ChronoStreetTurst es la evolución de este proyecto, incluyendo las siguientes mejoras:

- Almacenamiento de las imágenes en la nube.
- Sistema de gestión de usuarios en la nube.
- Interfaz Web para el control de estas imágenes y usuarios.

- Mejoras en ciertos aspectos de la APP tanto visuales como funcionales.



Ilustración 4. ChronoTourist [9].

4. Conceptos teóricos

En esta sección se presentarán los conceptos teóricos necesarios para la correcta comprensión del proyecto.

4.1. Realidad Aumentada

La Realidad Aumentada es el conjunto de tecnologías que hacen posible la interacción entre los objetos del mundo físico y los objetos del mundo virtual a través de dispositivos tecnológicos como las cámaras de los smartphones, tablets, ordenadores, etc.

Se caracteriza por:

- Coordinar el mundo real y virtual.
- Permitir una interacción en tiempo real.
- Amoldarse al ámbito en el que se introduce.
- Relacionarse con todas las capacidades físicas del entorno.

Es importante no confundir la Realidad Aumentada con la Realidad Virtual porque, mientras que la Realidad Aumentada incluye componentes digitales en el mundo real, la Realidad virtual crea un entorno totalmente virtual [20].

4.1.1. Historia de la Realidad Aumentada

La primera referencia a la Realidad Aumentada aparece en 1901 en la novela «The Master Key» escrita por Lyman Frank Baum, en la que un demonio le regala a un niño unas gafas, llamadas «Character Maker», que mostraban una letra en la frente de las personas indicando su personalidad.



Ilustración 5. Ejemplo Realidad Aumentada [3].

En la década de los 50, Morton Heilig [1] creó la primera máquina virtual llamada Sensorama (Ilustración 6. Sensorama (patente)), que consistía en una cabina cerrada en la que se proyectaba la película, la persona se aislaba sensorialmente de la realidad mediante altavoces, sensación de viento y olores. La primera película que se proyectó en esta cabina consistía en un viaje en bicicleta por las calles de Manhattan.

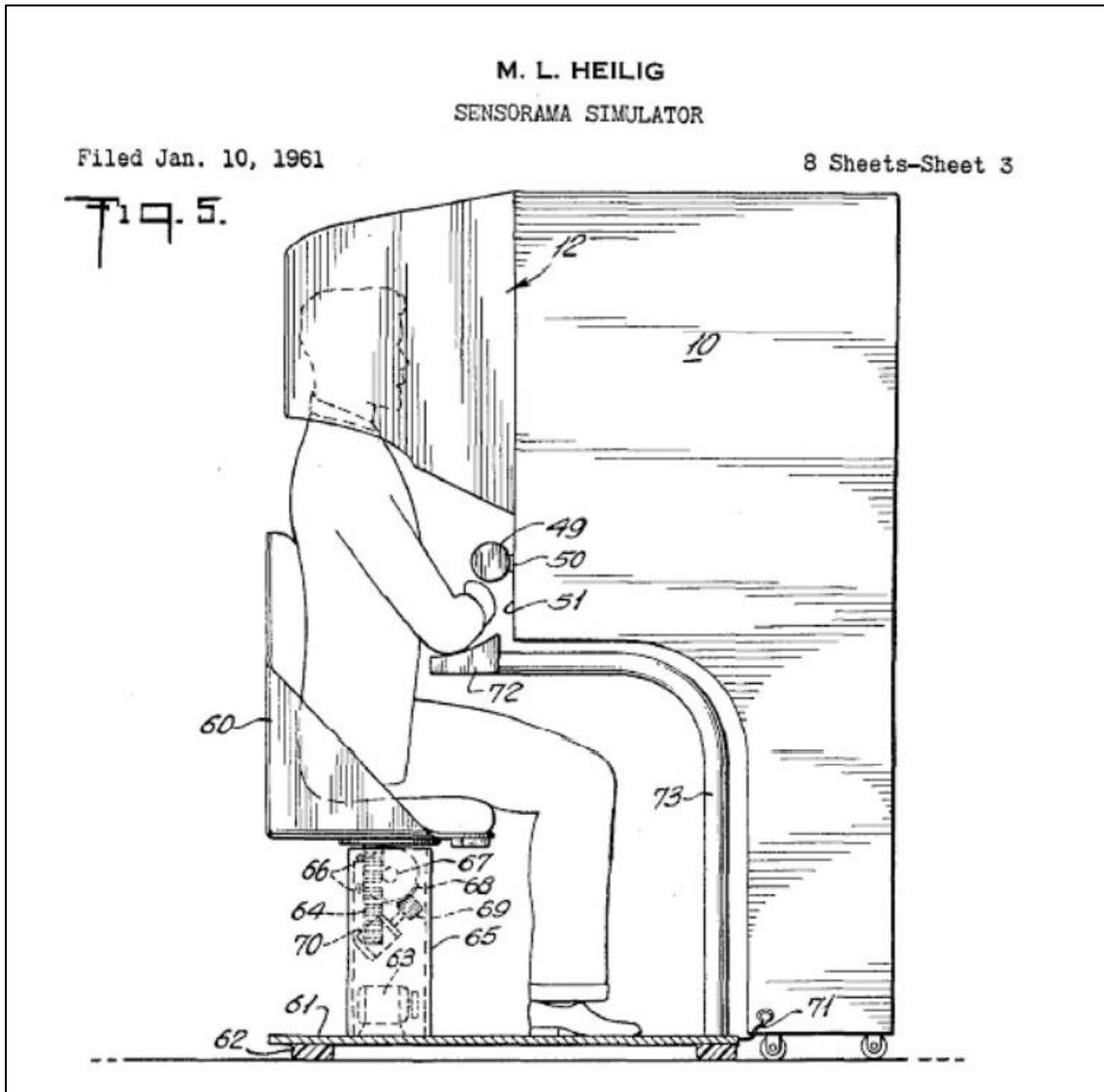


Ilustración 6. Sensorama (patente) [9].

En el año 1968, Iván Sutherland y David Cohen [2] crearon el primer prototipo de una máquina llamada «La Espada de Damocles» (Ilustración 7. Boceto de la espada de Damocles), considerado el primer visor de Realidad Aumentada y Realidad Virtual, consistía en dos tubos de rayos catódicos conectados a una computadora que representaba imágenes gráficas superpuestas en la escena real mediante un sistema de espejos. Además, el dispositivo estaba suspendido del techo mediante un brazo mecánico que permitía determinar la posición y orientación de la cabeza del usuario. En el siguiente [enlace](#) se puede ver un vídeo de demostración de este invento.

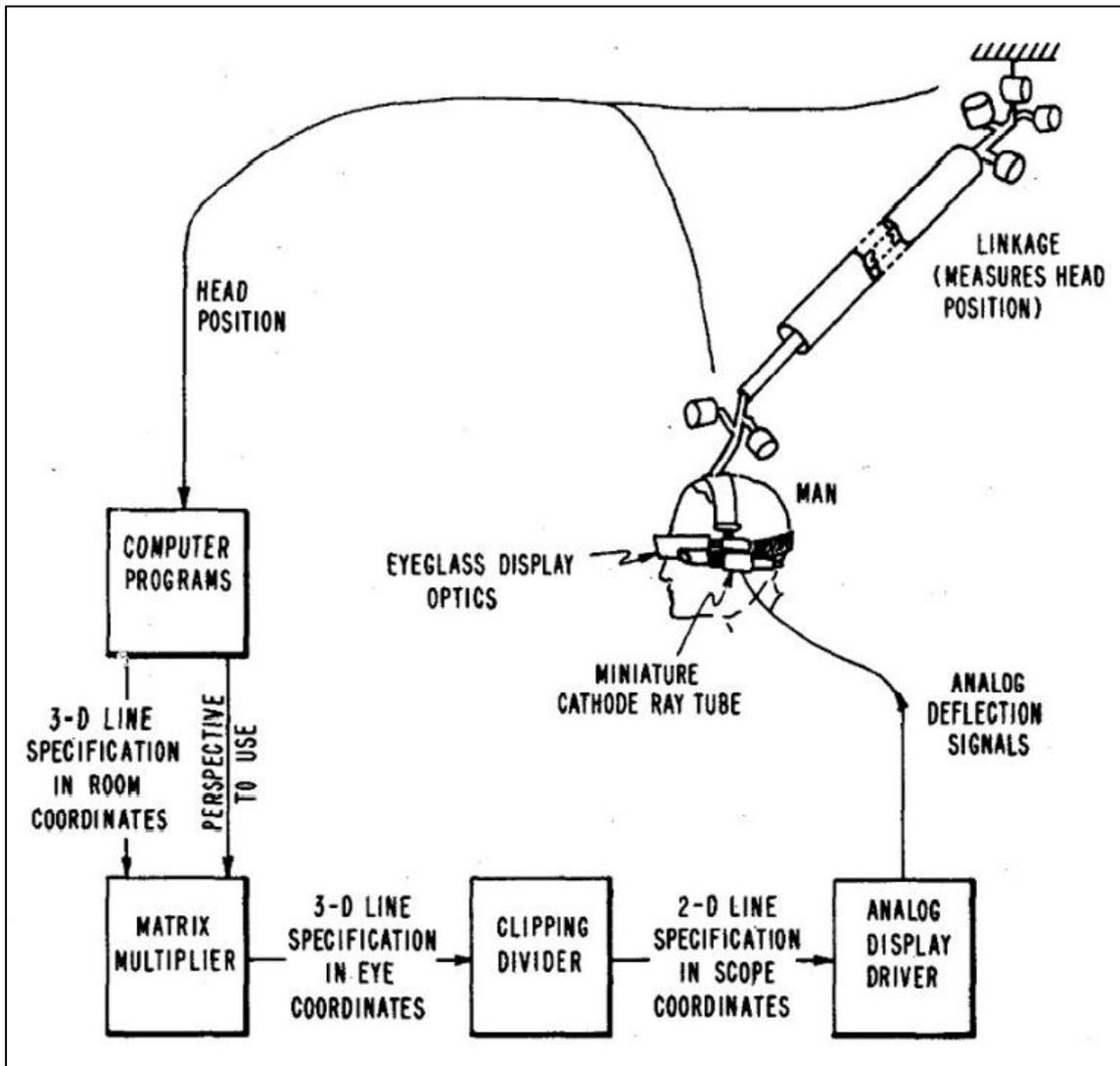


Ilustración 7. Boceto de la espada de Damocles [9].

A mediados de 1970 Myron Krueger inventó Videoplace [4] (Ilustración 8. Videoplace), un sistema de Realidad Artificial que respondía a los movimientos de los usuarios sin la necesidad de accesorios como gafas o guantes.

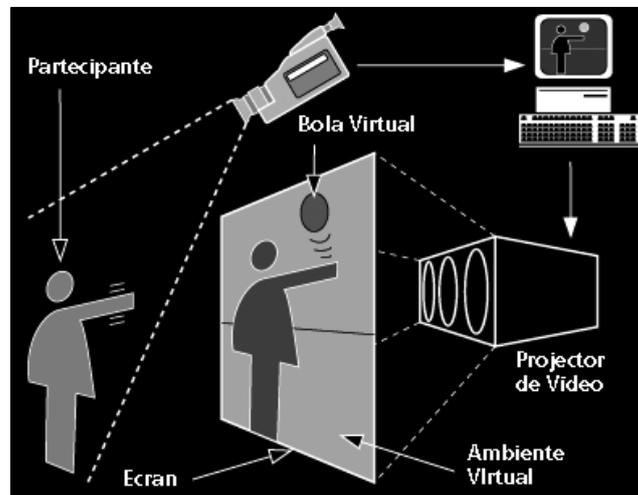


Ilustración 8. Videoplace, [9]

En 1982 se utilizó por primera vez la Realidad Aumentada en la televisión, gracias a la Realidad Aumentada interactiva creada por Dan Reitan, que fue usada para las retransmisiones meteorológicas.

A principios de la década de los 90, el inventor Louis Rosenberg [5] creó el primer sistema inmersivo de Realidad Aumentada: «Virtual Fixtures», que proyectaba unos brazos robóticos sobre el usuario y lo guiaban para la realización de algunas tareas.

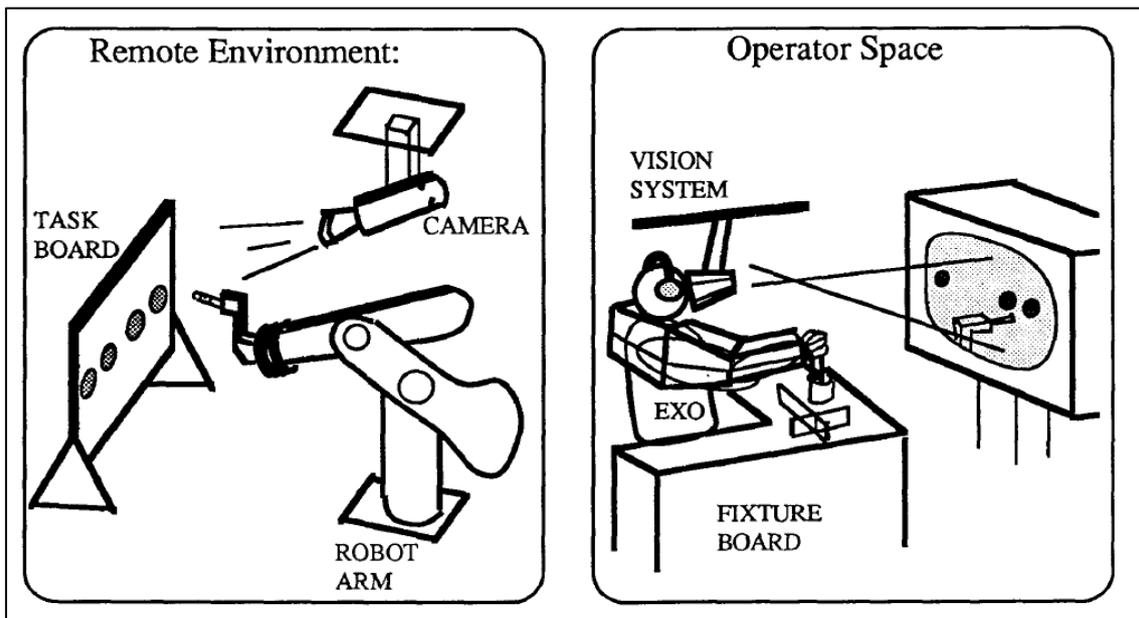


Ilustración 9. Virtual Fixtures [9]

En 1994, de la mano de Steve Feiner, Dore Seligmann y Blair Maclyntre[8] nace KARMA (Ilustración 10. KARMA), que es considerado el primer prototipo de Realidad Aumentada que se usaba para tareas de mantenimiento y reparación.



Ilustración 10. KARMA [9]

En el año 1999, Hirokazu Kato desarrolla ARToolkit, la primera librería de código abierto para el desarrollo de la Realidad Aumentada, mientras que la NASA empleó la Realidad Aumentada para mejorar la navegación de las pruebas de vuelo.

En el año 2000 se crea ARQuake [7] (Ilustración 11. ARQuake) el primer juego de Realidad Aumentada que permite jugar al aire libre, este juego estaba basado en el shooter de 8 bits

llamado Quake. En el siguiente [enlace](#) se puede ver una muestra del funcionamiento de este juego.



Ilustración 11. ARQuake, [9]

En el año 2008 se crea Wikitude, una plataforma de Realidad Aumentada que permite usar la cámara de los móviles para mostrar la Realidad Aumentada al aire libre. Un año más tarde se crea el logo oficial de Realidad Aumentada para facilitar la identificación de esta tecnología aplicada a cualquier medio.

En el año 2012 Google presenta el proyecto Google Glass, unas gafas de Realidad Aumentada basadas en Android destinadas al día a día de los usuarios [6].

En 2016, la empresa Niantic lanza el juego «Pokemon Go» un videojuego (Ilustración 12. PokemonGO) para dispositivos móviles Android e IOS que usa la Realidad Aumentada por geolocalización.



Ilustración 12. PokemonGO [9].

4.1.2. Niveles y Tipos de Realidad Aumentada

Niveles de Realidad Aumentada:

- **Nivel 0:** (también conocido como enlazado con el mundo físico): emplea imágenes en dos dimensiones como, por ejemplo, códigos de barras, que son utilizados como enlaces a otros contenidos.
- **Nivel 1:** (también conocido como RA con marcadores): se emplean aplicaciones que reconocen patrones de dos o tres dimensiones como, por ejemplo, figuras en blanco y negro, formas o dibujos.
- **Nivel 2:** (también conocido como Realidad Aumentada sin marcadores): en este nivel no es necesario el uso de marcadores, sino que utilizan sistemas como la brújula digital para conocer la ubicación del usuario y proyectar imágenes virtuales sobre la realidad cotidiana.
- **Nivel 3:** (también conocido como visión aumentada): dispositivos de alta tecnología que ofrecen una experiencia inmersiva y fusionada con la realidad.

Tipos de Realidad Aumentada:

- **Realidad aumentada basada en marcadores:** se usan marcadores visuales, ya sea un código QR o una imagen 2D para producir un resultado cuando se detecte el marcador. Para este tipo de Realidad Aumentada es necesario usar patrones distintos unos de otros para poder reconocerlos fácilmente (Ilustración 13. RA basada en marcadores.).



Ilustración 13. RA basada en marcadores.

- **Realidad Aumentada sin marcadores:** en este tipo de Realidad Aumentada se hace uso de elementos como el GPS, la brújula digital, medidor de velocidad..., con el fin de proporcionar información basada en la ubicación del usuario. Es frecuentemente usado para mapear direcciones, encontrar negocios cercanos entre otros (Ilustración 14. RA sin marcadores).



Ilustración 14. RA sin marcadores.

- **Realidad Aumentada a través de objetos tangibles:** este tipo de Realidad Aumentada se basa en el reconocimiento de objetos reales, por lo que una de sus principales desventajas es que necesita dispositivos con una alta capacidad de procesamiento (Ilustración 15. RA a través de objetos).



Ilustración 15. RA a través de objetos

- **Realidad aumentada basada en proyección:** consiste en la proyección de luz artificial sobre superficies del mundo real, y mediante la detección de interacción humana la máquina realiza acciones. También es normal el empleo de tecnología de plasma láser para proyectar un holograma (Ilustración 16. RA basada en proyección).



Ilustración 16. RA basada en proyección [9]

- **Smart Terrain:** este tipo de Realidad Aumentada consiste en la creación de mapas 3D en tiempo real. Esta capacidad proporciona un nuevo nivel de realismo, ya que detecta objetos reales y los integra en mapas 3D (Ilustración 17. Smart Terrain).



Ilustración 17. Smart Terrain

4.1.3. Usos de Realidad Aumentada

Usos de Realidad Aumentada Algunos de los usos de la Realidad Aumentada más comunes son [10]:

- **Videojuegos:** es el sector en el que más ha avanzado la Realidad Aumentada, ya que las grandes empresas de este sector tienen grandes desarrollos y lanzamientos de videojuegos que usan RA. Un ejemplo es Pokemon Go.
- **Visión de Realidad Aumentada:** herramientas como Google Glass, HoloLens de Microsoft o los cascos de realidad virtual de Samsung utilizan ya la Realidad Aumentada, ya que permiten que el usuario tenga la sensación de tocar objetos virtuales.
- **Moda:** existen tiendas que permiten que los usuarios puedan probarse ropa o maquillajes virtualmente, o para probarnos gafas.
- **Publicidad:** este sector fue uno de los pioneros en el uso de la Realidad Aumentada, ya que ven en la realidad ella un gran elemento para crear anuncios que consigan llamar la atención de los consumidores. Un ejemplo de esto es la aplicación Blippar que permite al usuario escanear productos, folletos, anuncios con el móvil para tener experiencias interactivas.
- **Reforma de interiores:** empresas como IKEA es pionera en el uso de Realidad Aumentada en los catálogos, ya que permite a los usuarios escanear el mueble y colocarlo en su hogar.

4.2. Tratamiento de la imagen

El tratamiento de la imagen es una técnica que nos permite modificar las imágenes con la finalidad de obtener mayor calidad, obtener una composición totalmente distinta a la original, etc.

4.2.1. Elementos de la imagen

Los elementos que componen a una imagen digital son [11]:

- **El píxel:** es la parte más pequeña que compone a una imagen y aporta información sobre el color, la saturación y el brillo. El color que compone un píxel se obtiene, a su vez, mediante tres colores (Rojo, verde y azul). En el mundo de la fotografía se usa más el concepto megapíxel, que equivale a un millón de píxeles.
- **Resolución:** hace referencia a la cantidad de filas y columnas de píxeles que tiene de ancho y de alto una imagen digital. La resolución suele medirse en píxeles por pulgadas o densidad por pantalla, esto quiere decir que cuantos más píxeles tenga una imagen por pulgada, estos serán más pequeños y la imagen tendrá mayor resolución.
- **Tamaño:** esta característica está relacionada directamente con los píxeles, por lo que cuantos más píxeles posea una imagen digital, más grande podrá hacerse. Cada píxel está formado por bits (unidad de almacenamiento más pequeña de información digital). A mayor cantidad de píxeles, más colores puede albergar una imagen.

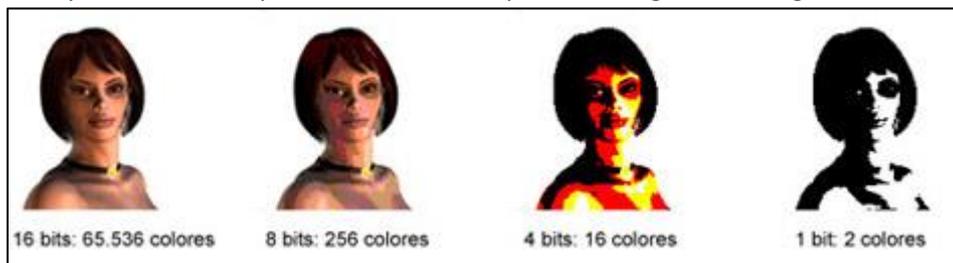


Ilustración 18. imagen con diferente número de Bits [9]

- **Profundidad de color:** es el número de bits utilizados para almacenar información sobre el color de cada píxel en una imagen digital.
- **Compresión:** capacidad de una imagen digital de comprimirse para posteriormente poder enviarla o guardarla de una manera más rápida, pero con la desventaja de que durante este proceso se va perdiendo la calidad de esta.

4.2.2. Espacios de color

Los espacios de color son organizaciones específicas de los colores de una imagen que nos permiten analizar los píxeles desde varios puntos de vista para poder aprovechar la información que contiene al completo. Algunos de los modelos de espacios de color son:

- **Modelo RGB:** este modelo es el más utilizado por la informática para crear y reproducir colores en monitores y pantallas. Está basado en la «síntesis aditiva», en la que se suman la intensidad de luz relativas al rojo, verde y azul para conseguir el resto de los colores. Su sistema de coordenadas es cúbico, en la que las coordenadas (0,0,0) representan el negro y las coordenadas (1,1,1) el blanco.

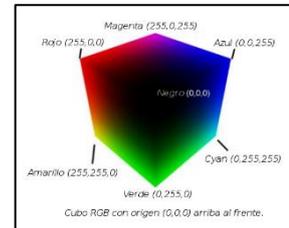


Ilustración 19. Modelo RGB

- **Modelo HSV:** sus siglas corresponden a Tono (Hue), Saturación (Saturation) y Valor (Value). Su sistema de coordenadas es cilíndrico y el subconjunto de este espacio donde se definen los colores es una pirámide con base hexagonal. Este modelo tiene la ventaja de no variar ante las condiciones de la luz, pero es muy complejo a nivel computacional.

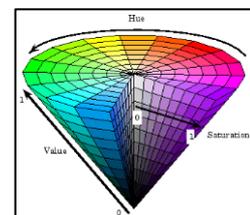


Ilustración 20. Modelo HSV

- **Modelo YcbCr:** en este modelo, el color se representa mediante tres parámetros: la luminancia (que es la cantidad lineal de luz que es directamente proporcional a la intensidad física) y dos valores diferentes de color (características del color). Este modelo se suele usar en los estudios de televisión europeos y en la compresión de imágenes.



Ilustración 21. Modelo YcbCr

4.3. CMS

CMS [12], también conocido como Content Management System o Sistema de Gestión de Contenidos, es un sistema online que hace la creación de una página Web mucho más sencilla. Es un software que te ayuda a administrar el contenido dinámico de la página.

Gracias a los CMS, casi cualquier persona sin ningún tipo de conocimiento sobre programación, puede administrar la página web desde una interfaz gráfica, desde crear y editar contenido hasta agregar imágenes y videos.

Los CMS permiten al usuario utilizar un editor de contenido para crear desde publicaciones y paginas hasta tiendas online(ecommerce). Lo que los hace tan fundamentales es que administra todo el contenido subido por ti, tan solo tendrás que publicarlo y aparecerá automáticamente en la web. Una de las mayores ventajas de los CMS es que no debes preocuparte por la parte técnica, es decir solo tienes que alimentar a la base de datos con el contenido que quieres publicar.

Los más comunes y con mejor puntuación son:

- JOOMLA
- DRUPAL
- WordPress

En mi caso, no tener acceso a la parte técnica hace más difícil el desarrollo de los controladores. Pero hay sistemas de manejo de todo tipo, algunos solo te manejan una pequeña parte del proyecto que sea común en muchos proyectos, como los usuarios, la herramienta utilizada se explicara en el apartado de Técnicas y Herramientas.

Estos no son reconocidos como CMS, pero también son manejadores de contenido, incluso hay CMS con la tecnología de estos, como Firebase.

4.4. APIS RESTful

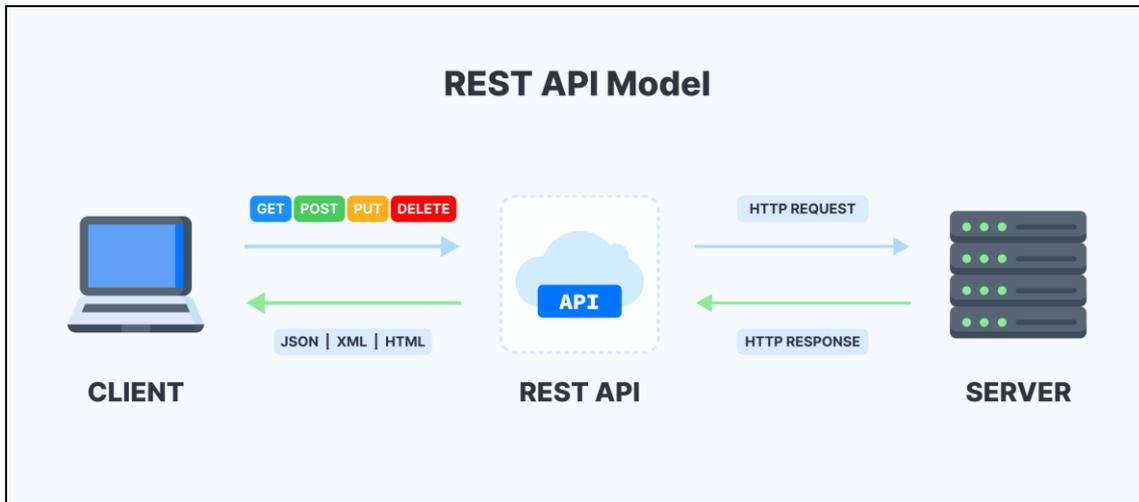


Ilustración 22. API RestFul [14]

API [13], Application Programming Interface, es un conjunto de funcionalidades que nos facilita un sistema para poder interactuar con él desde otro sistema, sin importar los lenguajes de programación ni la tecnología, debido a que trabaja con una capa de abstracción.

API Restful se ajusta a los límites de la arquitectura REST, que permite la iteración con servicios web de RESTful.

Roy Fielding, informático, es el creador de la transferencia de estado representacional (REST).

REST [14], Representational State Transfer, es un tipo de arquitectura de software para realizar la comunicación cliente-servidor, esta se apoya en el protocolo HTTP para la comunicación y los mensajes pueden estar en XML o JSON.

Resto es la mejora de SOAP, mejor, más sencillo y versátil.

Las peticiones que le hacemos al servidor mediante HTTP son:

- **GET:** Obtener recursos del servidor.
- **POST:** Insertar recursos nuevos en el servidor.
- **PUT:** Actualizar un recurso en el servidor.
- **PATCH:** Actualización temporal, a diferencia del PUT, no hace falta rellenar todos los campos, si uno no se rellena, no se actualizará.
- **DELETE:** Borrar un recurso en el servidor.

4.5. CLIENTE-SERVIDOR

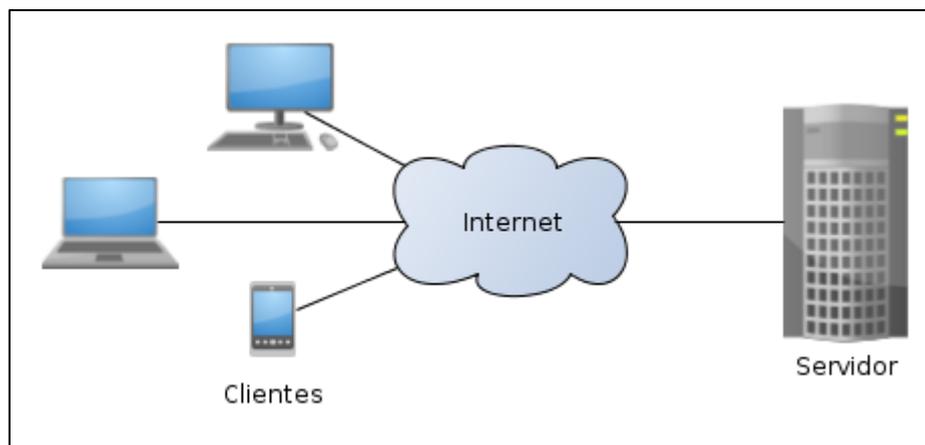


Ilustración 23. Cliente-Servidor

Es uno de los modelos más usados en servicios y protocolos de internet [21]. Tiene dos partes, por un lado, el cliente o grupo de clientes, y por otro el servidor, que normalmente debe ser una máquina potente con hardware y software específico que actúa como depósito de datos y gestor de base de datos o aplicaciones.

En esta arquitectura los clientes suelen ser estaciones de trabajo que solicitan servicios al servidor, mientras que el servidor es una máquina que actúa como depósito de datos y como sistema gestor de bases de datos, y, por tanto, se encarga de dar la respuesta demandada por el cliente.

El ejemplo más claro de la arquitectura cliente-servidor, es la propia red de internet, donde existen diferentes ordenadores conectados alrededor del mundo, los cuales se conectan a través de los servidores y de esta manera la información de los servicios requeridos viaja a través de internet dando respuesta a la solicitud demandada.

La principal importancia de este modelo es que permite conectar varios clientes a los servicios del servidor de manera simultánea. Y actualmente la gran mayoría de sistemas, por no decir todos, necesitan dicha propiedad.

Componentes principales:

- **Red:** Conjunto de clientes, servidores y bases de datos unidos, entre los que existen protocolos de transmisión de información.
- **Ciente:** Demandante de servicios.
- **Servidor:** Proveedor de servicios.
- **Protocolo:** Conjunto de normas o reglas sobre el flujo de información en una red.
- **Servicios:** Conjunto de información que busca responder las necesidades de un cliente.
- **Base de datos:** Bancos de información.

4.6. MVP

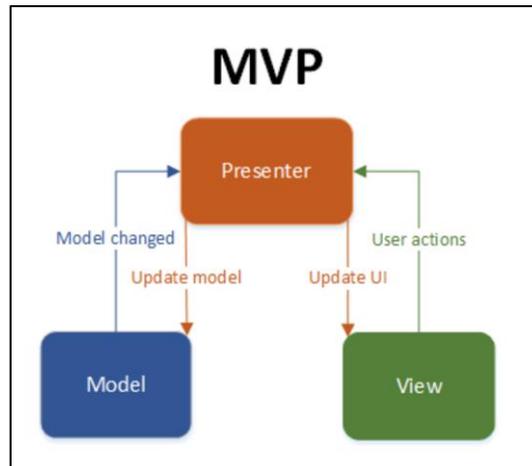


Ilustración 24. MVP

La arquitectura MVP, Modelo-Vista-Presentador, trata de delegar toda la funcionalidad en el presentador, que será el encargado de actualizar el modelo cuando ocurre un evento en la vista, y también de actualizar la vista en caso de que el modelo lo indique, es decir, actúa como un intermediario.

La vista se encargará de mostrarle una interfaz al usuario y transmitir al presentador los eventos y datos necesarios.

El modelo define los datos sobre los que actúa el presentador, en este caso serán los Targets de Vuforia y los usuarios de Firebase, cada uno en su respectiva escena.

Como podemos observar, MVP, es muy parecido MVC, Modelo-Vista-Controlador, con la diferencia que el presentador hace de intermediario, lo cual entre vista y modelo no va a haber flujo de información.

Recalcar que el presentador realmente es un controlador, pero se le llama así para diferenciar entre MVP y MCV.

5. Técnicas y Herramientas

Dentro de este apartado explicaremos las técnicas y herramientas empleadas en el desarrollo del proyecto.

5.1. Herramientas case

Las herramientas CASE (*Computer Aided Software Engineering*), son aquellas asociadas a la ingeniería del software, se han utilizado para la creación de tablas, diagramas, estimaciones de tiempos... durante todo el ciclo de vida del proyecto, son las siguientes:

5.1.1. EzStimate

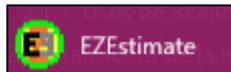


Ilustración 25. EzEstimate.

Es un programa que tiene como finalidad evaluar el esfuerzo de desarrollo de los casos de uso, incluyendo a los actores y medido en horas de trabajo. Ha sido de gran ayuda en la estimación del esfuerzo y planificación de tareas, detallado en el Anexo I.

5.1.2. Microsoft Office Project



Ilustración 26. M.Office Project

Es un software de administración de proyectos, comercializado por Microsoft para ayudar a administradores de proyectos en el desarrollo de planes, asignación de tareas, seguimiento del progreso, presupuesto...

En resumen, es un software muy útil para la gestión de proyectos.

En nuestro caso, ha sido muy útil para la planificación temporal.

5.1.3. VisualParadigm



Ilustración 27. VisualParadigm

Es una herramienta Case de UML capaz de proporcionar soporte de modelado, capacidades de ingeniería de código y genera informes, entre otras. Ha sido necesaria a lo largo de todo el proyecto para la realización de diagramas, presentes en todos los anexos y memoria.

5.1.4. Git



Ilustración 28. Git

Git es un proyecto de código abierto y con mantenimiento activo, creado por Linus Torvalds, creador también del kernel de Linux en 2005. Se caracteriza por una arquitectura distribuida y es el sistema de control de versiones más utilizado.

Su funcionamiento trata de permitir el acceso a todos los integrantes del proyecto, mediante una copia en forma de repositorio, además de poder ver todo el historial de cambios.

Las principales características de Git son:

- **Rendimiento:** Los algoritmos de Git aprovechan al máximo los atributos comunes de los árboles de archivos, generando así la máxima optimización en la confirmación de nuevos cambios, ramificación y comparación de versiones.
- **Seguridad:** El contenido se encuentra protegido mediante un algoritmo hash seguro (SHA1).
- **Flexibilidad:** eficiencia en trabajo tanto de flujo lineal como no lineal, y eficiencia tanto en proyectos grandes como pequeños.

Algunos ejemplos pueden ser GitHub o GitLab...

5.1.5. GitLab



Ilustración 29. GitLab

GitLab es una suite que permite la gestión, creación, administración y conexión de repositorios con diferentes aplicaciones y hacer todo tipo de integración con ellas.

Creado por Dmitriy Zaporozhets y Valery Sizov, el lenguaje de programación es Ruby.

Preferible usar GitLab antes que GitHub ya que la versión free esta mucho menos limitada.

5.1.6. REM

REM, siglas de REquirements Management, es una herramienta de Gestión de Requisitos diseñada para soportar la fase de Ingeniería de Requisitos de un proyecto de desarrollo software. Facilita la creación de tablas, proporcionando plantillas para la documentación de requisitos.

Sigue la metodología de Duran Y Bernárdez.

5.1.7. Doxygen



Ilustración 30. Doxygen.

Se trata de un generador de documentación para lenguajes como C, Java, Python....

Funciona en la mayoría de sistemas operativos.

En el proyecto se ha utilizado para generar la documentación, presente en el anexo IV.

5.1.8. Procreate



Ilustración 31. Procreate.

Aplicación de edición de gráficos rasterizados, desarrollada y publicada por Savage Interactive para iOS y iPadOS.

Usado para el diseño de la interfaz, anexo III y para los diagramas de secuencia a suicio, predecesores a visual paradigm.

5.1.9. JSDoc



Ilustración 32. JSDoc.

Se trata de un generador de documentación para JavaScript y TypeScript.

Funciona en la mayoría de sistemas operativos.

En el proyecto se ha utilizado para generar la documentación, presente en el anexo IV.

5.2. Técnicas y herramientas desarrollo

En este apartado hablaremos de las técnicas y herramientas utilizadas en el desarrollo del proyecto, divididas en dos partes:

- APP Unity.
- APP Web.

Primero hablaremos de las técnicas y herramientas generales, utilizadas en las dos aplicaciones:

GENERALES:

5.2.1. Vuforia



Ilustración 33. Vuforia

Vuforia es un kit de desarrollo software de Realidad Aumentada para dispositivos móviles. Actualmente proporciona una interfaz de programación de aplicaciones en Java, C++ y PHP(HTTP). Ofrece:

- Reconocimiento de texto y de imágenes.
- Rastreo robusto
- Detección y rastreo simultaneo de marcadores.
- Detección rápida de objetivos.

Al registrarse en la Vuforia como desarrolladores, nos permitirá crear nuevos marcadores, o también conocidos en el proyecto como Targets, para nuestra aplicación.

Add Target

Target Image File:

Choose File

.jpg or .png (max file 2mb)

Width:

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Metadata Package: (Optional)

max file 1mb

Name:

Ilustración 34. Add Target

Funcionamiento: La cámara del dispositivo captura en una imagen en vivo, Vuforia crea un frame y convierte la imagen a otra con diferente resolución, la analiza y busca similitudes en la base de datos.

Herramientas similares:

- OpenCV.
- Wikitude.
- ArCore.
- Arkit.
- Kudan.
- MaxST.

Todas ellas son herramientas para la Realidad Aumentada, la decisión estuvo igualada entre OpenCV y Vuforia, pero al final me decidí por Vuforia, aunque tenga menos información en internet sobre su programación y manejo, la interfaz que nos ofrece Vuforia es muy interesante a la hora de manejar los Targets. A continuación, muestro una comparativa entre todas las opciones:

	Vuforia	OpenCV	Wikitude	ARCore	Kudan	MaxST
Precio versión de pago	42€/mes	0€	2490€	0€	0€	49.9€/mes
Tipo licencia	>Gratis >Comercial	>Gratis	>Comercial	>Gratis >Comercial	>Gratis	>Comercial
Plataformas	Windows	Sí	Sí	Sí	No	Sí
	IOS	Sí	Sí	Sí	Sí	Sí
	Android	Sí	Sí	Sí	Sí	Sí
Marcadores	Reconocimiento facial	No	Sí	No	Sí	No
	Imágenes	Sí	Sí	Sí	Sí	Sí
	Objetos 3D	Sí	Sí	Sí	Sí	No
	Texto	Sí	No	No	No	No
	Múltiples Marcadores	Sí	Sí	Sí	No	Sí
	Reconocimiento en la nube	Sí	Sí	Sí	No	No
	GPS	Sí	Sí	Sí	Sí	Sí

Ilustración 35. Comparacion RA.

5.2.2. Drive, API Google



Ilustración 36. Drive

Google Drive es un servicio, introducido por la empresa Google en 2012, de alojamiento de archivos. Antes se conocía como Google Docs, lo que ahora es un editor de texto.

Drive es uno de los alojamientos en la nube más usado en el mundo y disponen de aplicación para todo tipo de dispositivos.



Ilustración 37. Google Cloud Platform.

Gracias a Google Cloud Platform, podemos enlazar el proyecto con la parte de Google Drive que queramos, para ello usaremos Google Drive API, en el siguiente enlace se encuentra la explicación del proceso:

<https://es.stackoverflow.com/questions/166054/c%C3%B3mo-subir-archivos-desde-mi-web-a-la-nube>

En el proyecto es de mucha necesidad, ya que la imagen a superponer por la Realidad Aumentada es interpretada como un link de descarga, de donde descargará y superpondrá. Para ello cuando un usuario suba las dos imágenes, de manera transparente, habrá un proceso de subida a drive, y generación del link de descarga.

5.2.3. Firebase



Ilustración 38. Firebase

Firebase es una plataforma para el desarrollo de aplicaciones web y móviles, lanzado en 2011 y comprada por Google en 2014.

Es plataforma ubicada en la nube, también integrada con Google Cloud Platform, que hace el manejo de usuarios y de bases de datos de tiempo real mucho más sencillo, acercándose al funcionamiento de un CMS pero con más posibilidad de modificación de los algoritmos de manejo.

En resumen, es como una base de datos de Google, con la posibilidad de interfaz de manejo de usuarios, todo ello preparado para la creación de aplicaciones.

Ventajas:

- **Sincronizar** fácilmente los datos de sus proyectos sin tener que administrar conexiones o escribir lógica de sincronización compleja.
- Usa un conjunto de **herramientas multiplataforma**: se integra fácilmente para plataformas web como en aplicaciones móviles.
- Usa la **infraestructura de Google** y escala automáticamente para cualquier tipo de aplicación, desde las más pequeñas hasta las más potentes.
- Crea proyectos **sin necesidad de un servidor**: Las herramientas se incluyen en los SDK para los dispositivos móviles y web, por lo que no es necesario la creación de un servidor para el proyecto.

En el proyecto se usa para el manejo de usuarios y administradores, está implementada tanto en la parte web como en la parte unity.

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
gcalvosu@gmail.com	✉	1 jun 2022	7 jun 2022	LkVE6LnsqLVgTnXXYk9Jp98syEJ3
yiye@usal.es	✉	26 may 2022	7 jun 2022	bsNFumcYwsQWVOI0IPACQTXeo...
dialgacs@hotmail.com	✉	23 may 2022	23 may 2022	9VfPQaV4ZOSi2UBWnEkxWsO2C...
guillermocalvosu@gmail.c...	✉	28 abr 2022	7 jun 2022	mjWVFL8Jq1PuMrd6qdG56Hx83F...

Ilustración 39. Tabla usuarios Firebase

5.2.4. Visual Studio

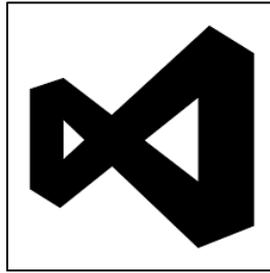


Ilustración 40. VisualStudio

Es el entorno de desarrollo utilizado para todos los lenguajes del proyecto, sus extensiones(addons) hacen que la programación sea más divertida.

Visual Studio es un entorno de desarrollo integrado para Windows y macOS, iniciado en 2003.

Visual Studio que permite crear sitios y aplicaciones web, así como cualquier servicio web en cualquier entorno compatible con la plataforma.

Es compatible con la mayoría de los lenguajes de programación y con la mayoría de entornos de desarrollo.

UNITY:

5.2.5. Unity



Ilustración 41.Unity.

Es un motor de videojuegos multiplataforma, está disponible para Linux, macOS y Microsoft.

Algunas de las funcionalidades que presenta son:

- programación de scripts
- Creación de Inteligencia Artificial
- Creación de animaciones
- Uso de motor grafico que imita las leyes de la física.

Tiene la capacidad de integrar Realidad Aumentada, debido al SDK de Vuforia comentado anteriormente.

5.2.6. UnitySimpleFileBrowser

Es un plugin que permite a los usuarios manejar archivos creados por el programa de una manera sencilla en tiempo de ejecución.

Enlace: [GitHub - yasirkula/UnitySimpleFileBrowser: A uGUI based runtime file browser for Unity 3D \(draggable and resizable\)](https://github.com/yasirkula/UnitySimpleFileBrowser)

WEB:

5.2.7. Node.JS



Ilustración 42.nodejs

Node.js es un entorno en tiempo de ejecución multiplataforma para la capa del servidor basado en el lenguaje JavaScript.

Es de código abierto, asíncrono, y con Entrada/Salida de datos en una arquitectura orientada a eventos. Está basado en el motor de Google V8.

Su objetivo es ser útil en la creación de programas escalables.

Fue creado en 2009 por Ryan Dahi.

Tiene una peculiaridad, se ejecuta en el servidor en vez de en el navegador.

5.2.8. NPM



Ilustración 43.NPM

Es el sistema de gestión de paquetes por defecto para Node.js.

5.2.9. Ajax



Ilustración 44. Ajax.

Asynchronous JavaScript And XML, es una técnica de desarrollo web para la creación de aplicaciones web asíncronas.

Estas aplicaciones se ejecutan en el navegador del cliente, mientras se tiene una comunicación asíncrona con el servidor en segundo plano.

Lo más interesante es el segundo plano, las funciones se van a ejecutar de manera transparente sin interferir en la visualización y comportamiento de la página.

Usamos Ajax en el proyecto para las llamadas a funciones de Vuforia, necesario que se hagan de manera asíncrona y en segundo plano.

5.2.10. Xampp



Ilustración 45.xampp.

Es un paquete de software libre, que consiste en:

- Sistema de gestión de bases de datos MySQL.
- Servidor web Apache
- Interpretes para lenguajes de PHP y Perl.

El nombre realmente es un acrónimo que viene de:

- X: cualquier sistema operativo
- Apache
- MariaDB/MySQL
- Php
- Perl.

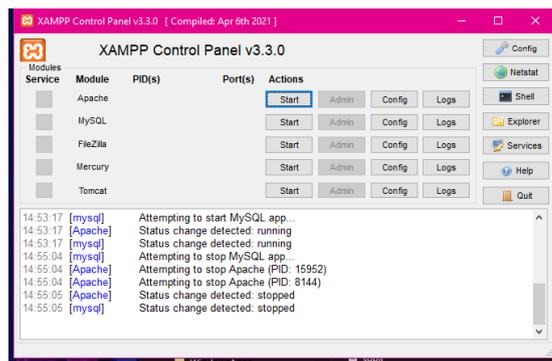


Ilustración 46.interfaz xampp

5.2.11. Apache Server

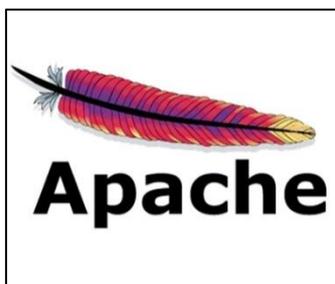


Ilustración 47.Apache.

Es un servidor web HTTP de código abierto, para Unix, Windows, Macintosh y alguna más.

Es desarrollado y mantenido por la comunidad de usuarios, pero bajo la supervisión de la Apache Software Foundation.

Características:

- Altamente configurable.
- Bases de datos de autenticación.
- Negocio de contenido.

Tiene una gran desventaja, no tiene una interfaz gráfica que ayude a la configuración de este.

5.2.12. Google Chrome y Microsoft Edge

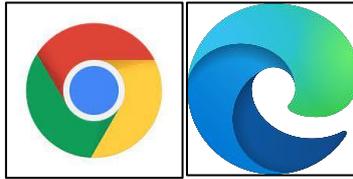


Ilustración 48. Navegadores

Son los navegadores donde se ha comprobado la funcionalidad de la aplicación web.

Google Chrome es desarrollado por Google, y Microsoft Edge es desarrollado por Microsoft.

5.2.13. Bootstrap



Ilustración 49. Bootstrap.

Es una biblioteca multiplataforma de código abierto para el diseño de sitios o aplicaciones web; ofrece a los usuarios plantillas de diseño, formularios, botones, menús de navegación.... Está enfocado solo en el front-end

Es el segundo proyecto más destacado en Git-Hub y es utilizado por organizaciones de gran prestigio, como la Nasa, MSNBC....

Dentro de Bootstrap, los componentes utilizados en el proyecto son:

5.2.13.1. Jasny Bootstrap



Librería de Bootstrap para la visualización de botones, paneles desplegables, menús....

Hace más fácil la creación de una interfaz sencilla y bonita.

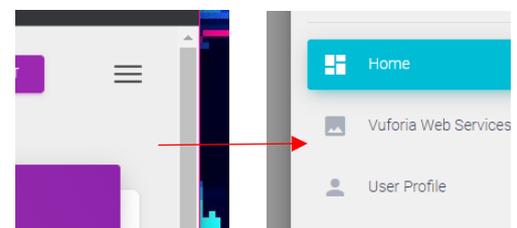


Ilustración 51. navbar

Ilustración 50. menu navbar

4.2.13.1. Bootstrap-material-design

Librería de Bootstrap que ayuda a mejorar la calidad del diseño de una interfaz web, mejorando los botones, imágenes, alertas, formularios, modales.....

LOGOUT

Es de las librerías más usadas dentro de Bootstrap.

A continuación, dejo un enlace donde se puede visitar para ver todas las posibilidades que tiene: <https://demos.creative-tim.com/material-dashboard/docs/2.1/getting-started/introduction.html>

5.2.14. JQuery



Ilustración 52. jQuery.

Es una biblioteca multiplataforma de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones, agregar interacción con la técnica AJAX a páginas web.

El uso principal en este proyecto la utilización de AJAX con jQuery.

```
$.ajax({
  url: "VuforiaWebServices/API_DeleteTarget.php",
  type: "POST",
  data: {targetID: targetId},
  dataType: "json",
  beforeSend: function (xhr){
  },

```

Ilustración 53. Ejemplo Ajax jQuery

5.2.15. SweetAlert



Ilustración 54. sweetalert

Es un plugin de jQuery que nos ayuda a darle un aspecto profesional a los mensajes lanzados al usuario, más conocidos como alertas.

Tiene muchas posibilidades de configuración diferentes.

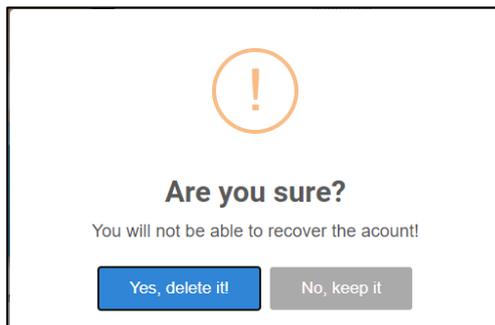
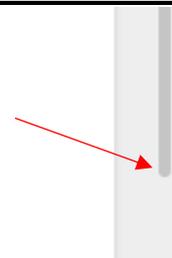


Ilustración 55. alerta sweetalert

5.2.16. Perfect scrollbar jQuery

Es un plugin de jQuery que hace la barra de navegación más atractiva y ayuda en caso de usar la web en una plataforma móvil.



5.2.17. DevTols



Ilustración 56. DevTools.

Conjunto de herramientas para desarrolladores web integrado en el navegador Google Chrome. Con las que se puede inspeccionar el HTML renderizado (DOM) y la actividad de red de sus páginas.

4.3 Lenguajes

A continuación, se mostrarán todos los lenguajes utilizados en el proyecto:

4.3.1. C#



Ilustración 57. C#.

Lenguaje de programación multiparadigma desarrollado y estandarizado por Microsoft. Es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común. Usado en toda la parte de Unity.

4.3.2. JavaScript



Ilustración 58. JavaScript

Lenguaje de programación interpretado, definido como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Usado para los controladores principales de la interfaz web.

4.3.3. Html



Ilustración 59. html.

HyperText Markup Language. Lenguaje de marcado que nos permite indicar la estructura que va a tener nuestro documento mediante etiquetas.

Nos ofrece:

- Gran adaptabilidad.
- Estructuración lógica.
- Fácil interpretación tanto por humanos como por máquinas.

Usado para toda la interfaz web.

4.3.4. PHP



Ilustración 60.php

Lenguaje de código abierto adecuado para el desarrollo web y que puede ser incrustado en HTML.

La principal diferencia con java script es que php se inicia desde el servidor, en cambio javascript lo hace desde el navegador del usuario. Así que para que sea más cómodo, las comunicaciones del servidor con vuforia, se hacen con php.

Se usa en todas las Apis y controladores de Vuforia.

4.3.5. JSON

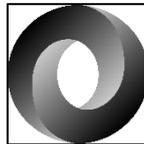


Ilustración 61. Json.

Lenguaje de intercambio de datos. Se hizo popular gracias a su texto legible por humanos, el cual lo hace mucho más fácil, y requiere menos codificación.

4.3.6. CSS



Ilustración 62. css

También conocido como “Hojas de estilo en cascada”, es un lenguaje de diseño gráfico que define y crea la parte visual de un documento estructurado escrito en un lenguaje marcado.

6. Aspectos relevantes del desarrollo del proyecto

En este apartado se va a exponer los aspectos más importantes relacionados con durante las distintas fases del desarrollo del proyecto. Primero se llevará a cabo una explicación de la documentación técnica desarrollada. Posteriormente, en los apartados 6.2, 6.3 y 6.4 se explicarán la metodología usada para el desarrollo y la estimación de esfuerzo llevada a cabo. La sección 6.5 se recoge lo más relevante en cuanto a la especificación de requisitos. La sección 6.6 y 6.7 contiene el análisis y diseño del sistema, y para finalizar, la sección 6.8 recoge las pruebas realizadas para comprobar el correcto funcionamiento del sistema.

6.1. Metodología

Para la realización del proyecto se han seguido las directrices de las metodologías ágiles.

Estas metodologías [23] permiten adaptar la forma de trabajo a las condiciones del proyecto, lo cual aporta mayor flexibilidad e inmediatez en la respuesta al cliente, dando así la posibilidad de amoldar el desarrollo del proyecto a las circunstancias del entorno. Mejorando así en gran escala la satisfacción del cliente.

Dentro de las metodologías ágiles [22] nos podemos encontrar diferentes tipos:

- Extreme Programming XP.
- SCRUM.
- Kanban.

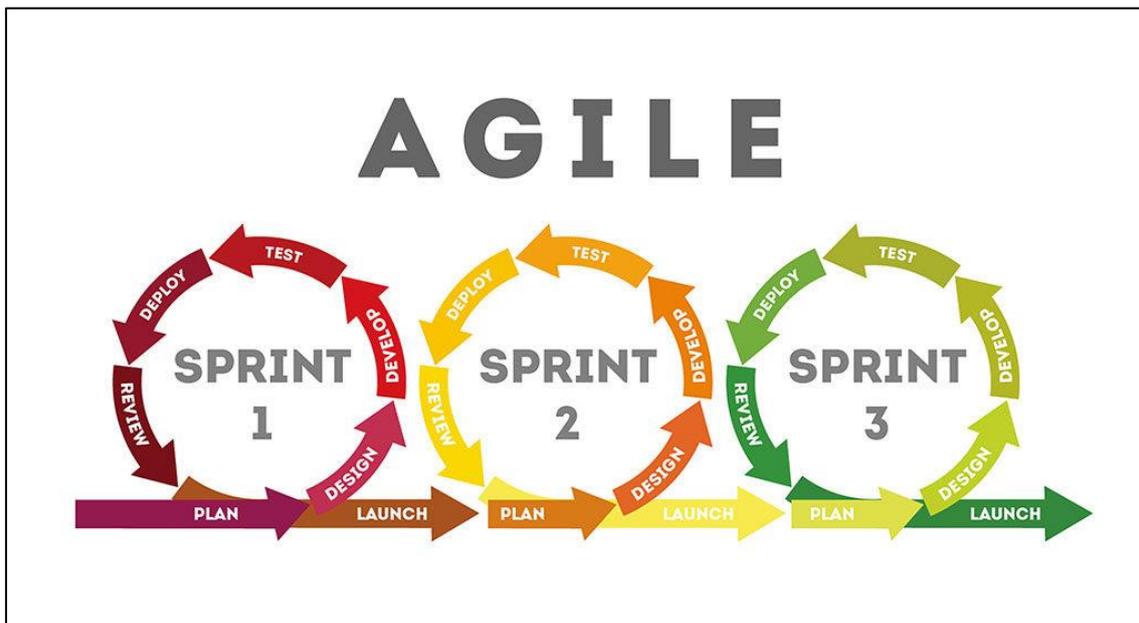


Ilustración 63. Metodología ágil

6.2.1 SCRUM

La metodología **SCRUM**, se basa en partir el proyecto en pequeños proyectos, divididos en 3 etapas: análisis, desarrollo y testing. En la parte de desarrollo nos encontraremos con iteraciones o Sprints, que son entregas regulares de partes del proyecto final al cliente.

La idea principal es darle un enfoque de proceso unificado, dado que es el más trabajado en la carrera, pero debido a la flexibilidad en el proceso de desarrollo, el pequeño equipo de trabajo y los cambios a lo largo del proceso, la mejor solución es enfocarlo con una metodología ágil, en concreto, la que acabamos de ver, metodología **SCRUM**. Marcando Sprints de 3 semanas, comenzando el primer Sprint el 1 de diciembre del 2021 y acabando el último Sprint la última semana de junio del 2022. Quedando los Sprints:

- Sprint 1: Fase de iniciación.
- Sprint 2: Fase de Planificación.
- Sprint 3: Fase de Planificación II.
- Sprint 4: Fase de implementación I. Unity.
- Sprint 5: Fase de implementación II. Web.
- Sprint 6: Fase de implementación III. Web.
- Sprint 7: Fase de Revisión.
- Sprint 2: Fase de Lanzamiento.

Dentro de la metodología **SCRUM**, existen 3 roles:

- **Product Owner:** Es el encargado de optimizar y maximizar el valor del producto, controlando el flujo de valor del proyecto. En este caso el tomador del papel es Guillermo Calvo, al cual le resultara más fácil sabiendo el resultado final del proyecto.
- **SCRUM Master:** Tienen dos tareas principales que son:
 - Gestionar el proceso Scrum.
 - Ayudar a eliminar impedimentos que puedan afectar a la entrega del producto.En este caso los **SCRUM Master** del proyecto serán Sara Rodríguez, Luis Martín y Fernando Rodríguez.
- **Develop Team:** Se trata del equipo de desarrollo, que realiza las tareas y entrega el producto final. En este caso el **Develop Team** estará compuesto por Guillermo Calvo.

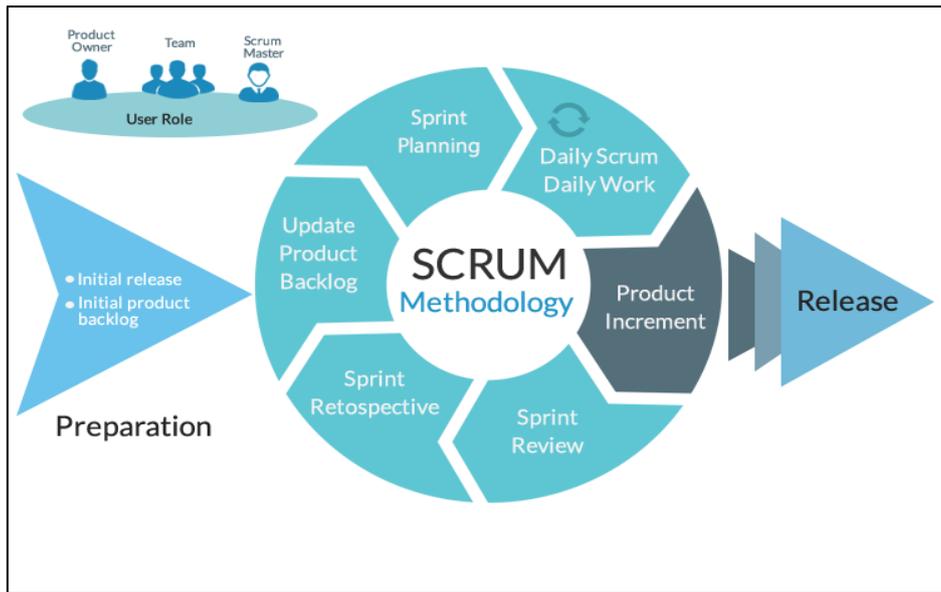


Ilustración 64. SCRUM

6.2. Estimación de esfuerzo

Es la parte esencial para la estimación de costes del proyecto [24]. Esta estimación se va a realizar con la métrica UCP, análisis de puntos de casos de uso, compuesta por una serie de cálculos y estimaciones que veremos a continuación.

Dicha métrica utiliza actores, escenarios y factores técnicos y de entorno para calcularla. Las variables utilizadas para el cálculo son:

- **UUCP**: Puntos de casos de uso desajustados. Dicho valor es igual a la suma de las siguientes variables:
 - **UUCW**: Peso de los casos de uso desajustado.
 - **UAW**: Peso de los actores desajustado.
- **TCF**: Factor de complejidad técnica.
- **EF**: Factor de complejidad del entorno.

UCP viene dado por la siguiente formula:

$$UCP = UCCP * TCF * EF$$

Donde:

$$UCCP = UUCW + UAW$$

Estos cálculos se realizan asignando un peso a cada factor y a cada caso de uso del sistema, este peso con relación a su complejidad. [25]

6.2.1. Complejidad de los Casos de Uso

Transacción: Conjunto de actividades atómicas. Desde que el usuario inicia una acción hasta que el sistema espera a la nueva acción del usuario.

CASOS DE USO	RAZÓN	PESO
Simple	El número de transacciones es 3 o menor.	1
Medio	El número de transacciones esta entre 4 y 7	2
Complejo	El número de transacciones es mayor de 7	3

Tabla 5. complejidad según las transacciones.

	CASOS DE USO	CATEGORIA	PESO
UUCW	Iniciar sesión con Google	Simple	5
	Registrarse	Medio	10
	Iniciar sesión	Simple	5
	Recuperar contraseña	Medio	10
	Cerrar sesión	Simple	5
	Registrar usuario/administrador	Medio	10
	Listar usuarios	Medio	10
	Ver información usuario	Simple	5
	Modificar datos del usuario	Simple	5
	Dar de baja usuario	Medio	10
	Re autenticar	Simple	5
	Cambiar contraseña	Medio	10
	Cambiar Rol	Simple	5
	Listar Targets	Medio	10
	Filtrar por Nombre	Medio	10
	Filtrar por Correo	Medio	10
	Filtrar por Ciudad	Medio	10
	Mostrar/Ocultar Activos	Medio	10
	Mostrar/Ocultar Pendientes	Medio	10
	Borrar Target	Medio	10
	Activar/Desactivar Target	Simple	5
	Crear Nuevo Target	Medio	10
	Ver Target	Medio	10
	Editar Target	Medio	10
	Subir img a Drive	Simple	5
	Subir ficheros Servidor	Simple	5
	Coger ficheros Servidor	Simple	5
	Activar/Desactivar AR	Simple	5
	Visualizar Realidad Aumentada	Medio	10
	Reconocer Monumento	Medio	10
	Superposición de Imagen	Medio	10
	Mostrar información	Medio	10
	Visitar Web con Información	Medio	10
Realizar Captura De Pantalla.	Simple	5	
Total			275

Tabla 6. Pesos Casos de Uso.

6.2.2. Complejidad de Actores

ACTORES	RAZÓN	PESO
Simple	Es un sistema que se comunica con la aplicación mediante una API.	1
Medio	Es un sistema que se comunica con la aplicación mediante un protocolo (internet)	2
Complejo	Persona que interactúa con el sistema mediante una interfaz	3

Tabla 7. Complejidad de Actores.

	CASOS DE USO	CATEGORIA	PESO
UAW	Administrador	Complejo	3
	Usuario No Registrado	Complejo	3
	Usuario Registrado	Complejo	3
Total			9

Tabla 8. Pesos Actores.

Por tanto, si hacemos cuentas:

$$UUCP = 275 + 9 = 284$$

6.2.3. Factores

En este apartado veremos la complejidad percibida de los factores de complejidad técnica (TCF), que son trece, y de los factores de complejidad del entorno (EF). Cada factor tendrá asignado un peso acorde a su repercusión.

6.2.3.1. Factores de complejidad Técnica

Veremos los factores de complejidad técnica, una vez calculados, obtendremos TCF mediante la siguiente formula:

$$TCF = C_1 + C_2 \sum_{i=1}^{13} W_i F_i$$

Factor	Nombre	Peso	Valor	Factor
T1	sistema distribuido	2	2	4
T2	Rendimiento	2	2	4
T3	Eficiencia del usuario final	1	4	4
T4	Procesamiento interno complejo	1	2	2
T5	Reusabilidad	1	2	2
T6	Facilidad de instalación	0.5	1	0.5
T7	Facilidad de uso	0.5	4	2
T8	Portabilidad	2	1	2
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	0	0
T11	Características especiales de seguridad	1	3	3
T12	Acceso directo a terceras partes	1	3	3
T13	Entrenamiento especial del usuario	1	0	0
TOTAL (TFactor)				30

Tabla 9. Factores de complejidad Técnica.

$$TCF = 0.6 + (0.01 * TFactor) = 0.9$$

6.2.3.2. Factores de complejidad del Entorno

En este apartado veremos los valores de los factores de complejidad del entorno. Una vez calculados podremos calcular ECF, mediante la siguiente ecuación:

$$TCF = C_1 + C_2 \sum_{i=1}^8 W_i F_i$$

Factor	Nombre	Peso	Valor	Factor
E1	Familiaridad con UML	1.5	2	3
E2	Trabajadores a tiempo parcial	-1	0	0
E3	Capacidad de los analistas	0.5	2	1
E4	Experiencia en la aplicación	0.5	3	1.5
E5	Experiencia en la orientación a objetos	1	4	4
E6	Motivación	1	5	5
E7	Dificultad del lenguaje de programación	-1	1	-1
E8	Estabilidad de los requisitos	2	3	6
TOTAL (EFactor)				20

Tabla 10. Factores de complejidad del Entorno.

$$TCF = 1.4 + (-0.03 * RFactor) = 0.8$$

Ahora con todas las variables calculadas podemos hacer los cálculos de los que hablamos al principio:

$$UCP = 284 * 0.9 * 0.8 = 204,48$$

Multiplicando por la constante *esfuerzo F*:

$$TPH = 20 * UCP = 20 * 204,48 = 4.089,6horas/UCP$$

Este último valor, TPH, indica las horas necesarias para completar el proyecto, lo cual este proyecto, una persona trabajando a jornada completa, tardaría casi 2 años:

EZEstimate - C:\Users\dialg\Desktop\TFG\MEMORIA\MEMORIA Y ANEXOS\Anexo 1\EZEstimate.ezp

File Settings Help

Module

Casos de Uso

Add Module Delete

Summary

Total Modules 1 **Excel Report** Generate Report

Use cases Simple 13 Average 21 Complex 0

Actors Simple 0 Average 0 Complex 3

Add Actor / Use case

Actor / Use case Name Select Type Complexity Add

Actor Complex

Tech / Env Factors

Set Tech Factor

Set Env Factors

Estimation Summary

UAW 9

UUCW 275

UUPC = UAW + UUCW 284

TFactor 30

EFactor 20

TCF = 0.6 + (.01*TFactor) 0,9

EF = 1.4 + (-0.03*EFactor) 0,8

UCP = UUPC*TCT*EF 204,48

Total Effort@ 20 Hrs/UCP 4089,6

Use case / Actor List (Double click to delete)

Id	Module	Type	Name	complexity
1	Casos de Uso	Usecase	Iniciar sesión c...	Simple
10	Casos de Uso	Usecase	Dar de baja usu...	Average
11	Casos de Uso	Usecase	Re autentificar	Simple
12	Casos de Uso	Usecase	Cambiar contra...	Average
13	Casos de Uso	Usecase	Cambiar Rol	Simple
14	Casos de Uso	Usecase	Listar Targets	Average
15	Casos de Uso	Usecase	Filtrar por Nombre	Average
16	Casos de Uso	Usecase	Filtrar por Correo	Average
17	Casos de Uso	Usecase	Filtrar por Ciudad	Average
18	Casos de Uso	Usecase	Mostrar/Ocultar...	Average
19	Casos de Uso	Usecase	Mostrar/Ocultar...	Average
2	Casos de Uso	Usecase	Registrarse	Average
20	Casos de Uso	Usecase	Borrar Target	Average
21	Casos de Uso	Usecase	Activar/Desacti...	Simple
22	Casos de Uso	Usecase	Crear Nuevo T...	Average
23	Casos de Uso	Usecase	Ver Target	Average
24	Casos de Uso	Usecase	Editar Target	Average
25	Casos de Uso	Usecase	Subir ima a Drive	Simple

Ilustración 65. EzEstimate

6.3. Planificación temporal

La planificación temporal es imprescindible en proyectos de desarrollo software. Nos sirve antes, durante y después de la realización del proyecto.

Antes, nos ayudará a realizar una estimación del tiempo que se necesitará para su correcta elaboración, incluyendo las subtareas (planificación macroscópica).

Durante, nos dará organización a la hora de los tiempos transcurridos en cada bloque.

Después, nos permitirá hacer una comparación entre el tiempo real y el supuesto al principio del proyecto, dándonos oportunidad de mejorar nuestros tiempos estimados para futuros proyectos.

En la [Ilustración 49: Planificación temporal Resumida](#) se muestra la planificación temporal de los bloques generales, indicando a gran escala los tiempos de ejecución y de finalización...

En la [Ilustración 51: Planificación temporal](#) se muestran las tareas a realizar para alcanzar los objetivos del proyecto, con sus respectivos tiempos de ejecución y de finalización.

Resumida por bloques:

	Nombre	Duración	Inicio	Finalización
1	> Sprint 1. Fase de iniciación	20 días	2/11/2021	29/11/2021
9	> Sprint 2. Fase de Planificación I	23 días	1/12/2021	31/12/2021
15	> Sprint 3. Fase de Planificación II	21 días	3/1/2022	31/1/2022
20	> Sprint 4. Fase de Implementación I, Unity	22 días	1/2/2022	2/3/2022
28	> Sprint 5. Fase de Implementación II. Web	18 días	1/3/2022	24/3/2022
32	> Sprint 6. Fase de Implementación III. Web	21 días	1/4/2022	29/4/2022
38	> Sprint 7. Fase de Revisión	17 días	3/5/2022	Hoy
45	> Sprint 8. Fase de Lanzamiento	18 días	1/6/2022	24/6/2022

Ilustración 66. planificación temporal Resumida.

Extendida:

Nombre	Duración	Inicio	Finalización	Depende de	Asignado a
1 <input type="radio"/> ▼ Sprint 1. Fase de iniciación	20 días	2/11/2021	29/11/2021		
2 <input type="radio"/> Identificación del grupo de trabajo	3 días	2/11/2021	4/11/2021		GC Guillermo calvo
3 <input type="radio"/> Especificación de requisitos	5 días	4/11/2021	10/11/2021		GC Guillermo calvo
4 <input type="radio"/> Identificación de los objetivos del proyecto	4 días	9/11/2021	12/11/2021		GC Guillermo calvo
5 <input type="radio"/> Planteamiento de los recursos necesarios	4 días	15/11/2021	18/11/2021		GC Guillermo calvo
6 <input type="radio"/> Planificación temporal	2 días	18/11/2021	19/11/2021		GC Guillermo calvo
7 <input type="radio"/> Detección de prioridades	2 días	22/11/2021	23/11/2021		GC Guillermo calvo
8 <input type="radio"/> Anexo I	4 días	24/11/2021	29/11/2021	6 2	GC Guillermo calvo
9 <input type="radio"/> ▼ Sprint 2. Fase de Planificación I	23 días	1/12/2021	31/12/2021		
10 <input type="radio"/> Definición de objetivos	2 días	1/12/2021	2/12/2021		GC Guillermo calvo
11 <input type="radio"/> Planteamiento Casos de Uso	6 días	3/12/2021	10/12/2021	10	GC Guillermo calvo
12 <input type="radio"/> Definición de las funcionalidades del sistema	2 días	13/12/2021	14/12/2021		GC Guillermo calvo
13 <input type="radio"/> Estudio de las herramientas a utilizar	3 días	15/12/2021	17/12/2021	12	GC Guillermo calvo
14 <input type="radio"/> Estudio del proyecto predecesor	10 días	20/12/2021	31/12/2021		GC Guillermo calvo
15 <input type="radio"/> ▼ Sprint 3. Fase de Planificación II	21 días	3/1/2022	31/1/2022		
16 <input type="radio"/> Definición de casos de uso	8 días	3/1/2022	12/1/2022		GC Guillermo calvo
17 <input type="radio"/> Definición de las herramientas a utilizar	7 días	13/1/2022	21/1/2022		GC Guillermo calvo
18 <input type="radio"/> Definición de la capa de Datos	4 días	24/1/2022	27/1/2022		GC Guillermo calvo
19 <input type="radio"/> Anexo II	6 días	24/1/2022	31/1/2022	16	GC Guillermo calvo
20 <input type="radio"/> ▼ Sprint 4. Fase de Implementación I, Unity	22 días	1/2/2022	2/3/2022		
21 <input type="radio"/> Identificación de capas	1 día	1/2/2022	1/2/2022		GC Guillermo calvo
22 <input type="radio"/> Diagrama de despliegue general	1 día	2/2/2022	2/2/2022		GC Guillermo calvo
23 <input type="radio"/> Preparación del entorno	5 días	3/2/2022	9/2/2022		GC Guillermo calvo
24 <input type="radio"/> Despliegue y desarrollo de la capa de datos y visualización	5 días	10/2/2022	16/2/2022	23	GC Guillermo calvo
25 <input type="radio"/> Unión de la capa de datos con la app	2 días	16/2/2022	17/2/2022	23 22	GC Guillermo calvo
26 <input type="radio"/> Puesta a punto de la app unity	5 días	17/2/2022	23/2/2022	24 23	GC Guillermo calvo
27 <input type="radio"/> Anexo III	5 días	24/2/2022	2/3/2022	25 24	GC Guillermo calvo
28 <input type="radio"/> ▼ Sprint 5. Fase de Implementación II. Web	18 días	1/3/2022	24/3/2022		
29 <input type="radio"/> Preparación del entorno	7 días	1/3/2022	9/3/2022		GC Guillermo calvo
30 <input type="radio"/> Desarrollo de la capa de visualización	7 días	9/3/2022	17/3/2022		GC Guillermo calvo
31 <input type="radio"/> Desarrollo y despliegue de la capa de datos	7 días	16/3/2022	24/3/2022		GC Guillermo calvo
32 <input type="radio"/> ▼ Sprint 6. Fase de Implementación III. Web	21 días	1/4/2022	29/4/2022		
33 <input type="radio"/> Desarrollo de la capa de control	7 días	1/4/2022	11/4/2022		GC Guillermo calvo
34 <input type="radio"/> Despliegue de la capa de control	5 días	12/4/2022	18/4/2022	33	GC Guillermo calvo
35 <input type="radio"/> Despliegue de la capa de visualización	3 días	18/4/2022	20/4/2022		GC Guillermo calvo
36 <input type="radio"/> Finalización y puesta a punto de la app web	7 días	21/4/2022	29/4/2022		GC Guillermo calvo
37 <input type="radio"/> Anexo IV	6 días	22/4/2022	29/4/2022	35 34	GC Guillermo calvo
38 <input type="radio"/> ▼ Sprint 7. Fase de Revisión	17 días	3/5/2022	Hoy		
39 <input type="radio"/> Revisión, organización y perfeccionamiento de los scripts	5 días	3/5/2022	9/5/2022		GC Guillermo calvo
40 <input type="radio"/> Consolidación de la parte funcional	1 día	10/5/2022	10/5/2022		GC Guillermo calvo
41 <input type="radio"/> Revisión y refinamiento de la parte de visualización	3 días	11/5/2022	13/5/2022		GC Guillermo calvo
42 <input type="radio"/> Revisión y feedback de los tutores	5 días	16/5/2022	20/5/2022		GC Guillermo calvo
43 <input type="radio"/> Manual usuario	2 días	23/5/2022	24/5/2022		GC Guillermo calvo
44 <input type="radio"/> Manual programador	2 días	24/5/2022	Hoy		GC Guillermo calvo
45 <input type="radio"/> ▼ Sprint 8. Fase de Lanzamiento	18 días	1/6/2022	24/6/2022		
46 <input type="radio"/> Realización de pruebas	5 días	1/6/2022	7/6/2022		GC Guillermo calvo
47 <input type="radio"/> Anexo V	4 días	8/6/2022	13/6/2022		GC Guillermo calvo
48 <input type="radio"/> Memoria	7 días	14/6/2022	22/6/2022	47	GC Guillermo calvo
49 <input type="radio"/> Refinamiento de las documentaciones	2 días	23/6/2022	24/6/2022	47 48	GC Guillermo calvo

Ilustración 67. planificación temporal

Como se puede apreciar en la figura anterior, el proyecto ha sido dividido en 8 Sprints:

- **Fase de iniciación:** Donde se va a inicializar el proyecto y se va a hacer una planificación general.
- **Fase de Planificación I:** Planteamos algunas de las partes principales del proyecto, como los casos, funcionalidades del sistema, herramientas... También estudiaremos el proyecto desde el cual partimos.
- **Fase de Planificación II:** Remataremos los conocimientos procesados en el anterior Sprint, definiendo correctamente todos ellos.
- **Fase de Implementación I, Unity:** Prepararemos el entorno y construiremos las capas necesarias para la parte de Unity y haremos la puesta a punto.
- **Fase de Implementación II, Web:** Prepararemos el entorno Web y desarrollaremos la capa de visualización y de datos.
- **Fase de Implementación III, Web:** Acabaremos el desarrollo de la parte Web y haremos la puesta a punto.
- **Fase de Revisión:** Realizaremos una revisión al proyecto para corregir errores y sesgos.
- **Fase de Lanzamiento:** Prepararemos el proyecto para el lanzamiento, junto con una serie de feedbacks, para saber la calidad de este.

Hay muchas maneras de representar la planificación temporal, dentro de las empresas, con cualquier metodología; la dos más usada siempre son:

- **Diagramas de Gantt.**
- **Redes de tareas.**

Entre estas, la mejor opción son las Redes de tareas ya que pueden representar las actividades que se realizan en paralelo, que no tienen dependencia entre ellas y las que se realizan secuencialmente.

A pesar de esta ventaja, la planificación temporal se representa mediante un diagrama Gant, ya que el número de tareas paralelas ha sido muy reducido, casi nulo. El diagrama se puede observar en las ilustraciones [Diagrama Gantt 1](#) e [Ilustración 52. Diagrama Gantt 2](#)

6.3.1. Diagrama Gantt

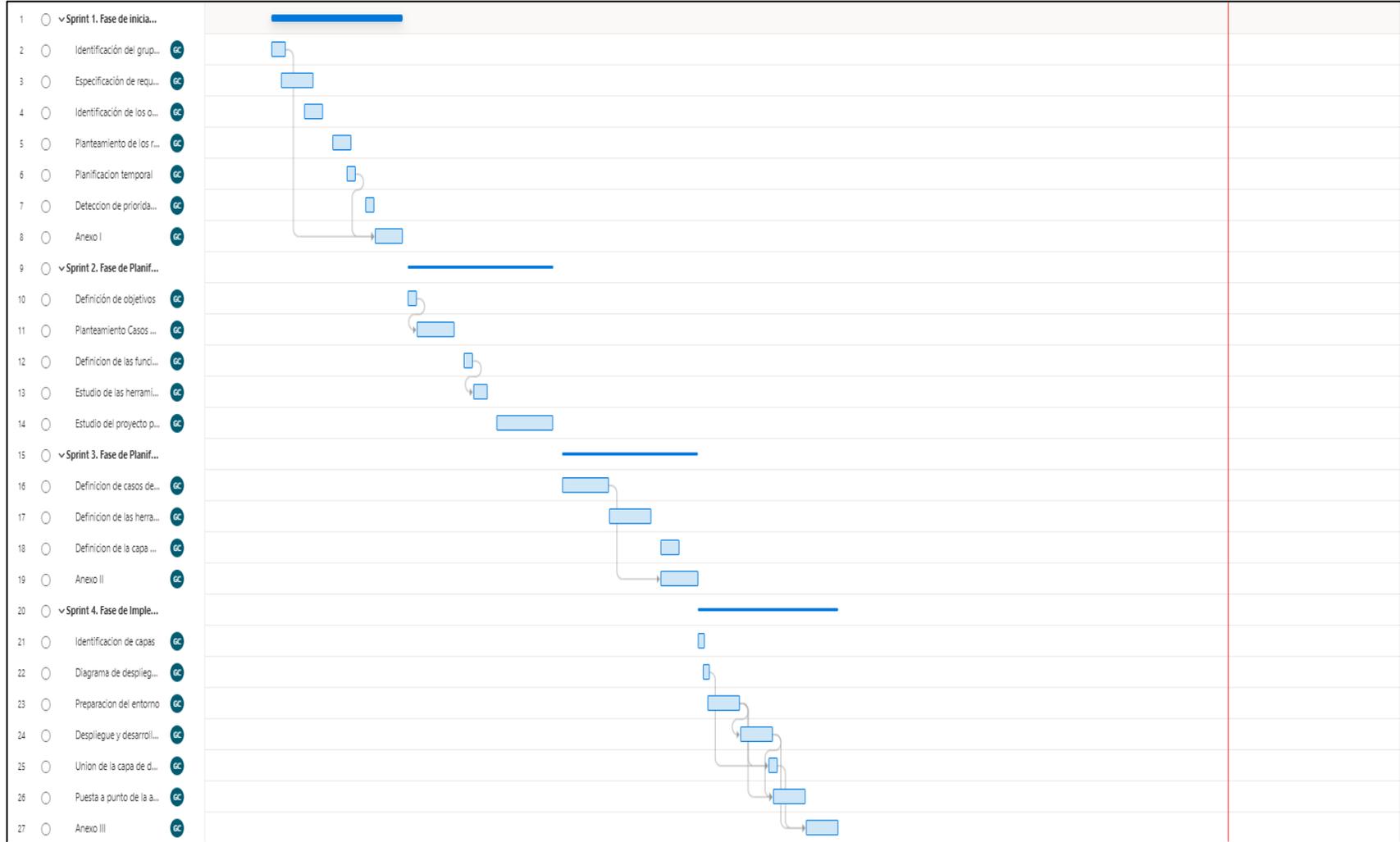


Ilustración 68. Diagrama Gantt 1

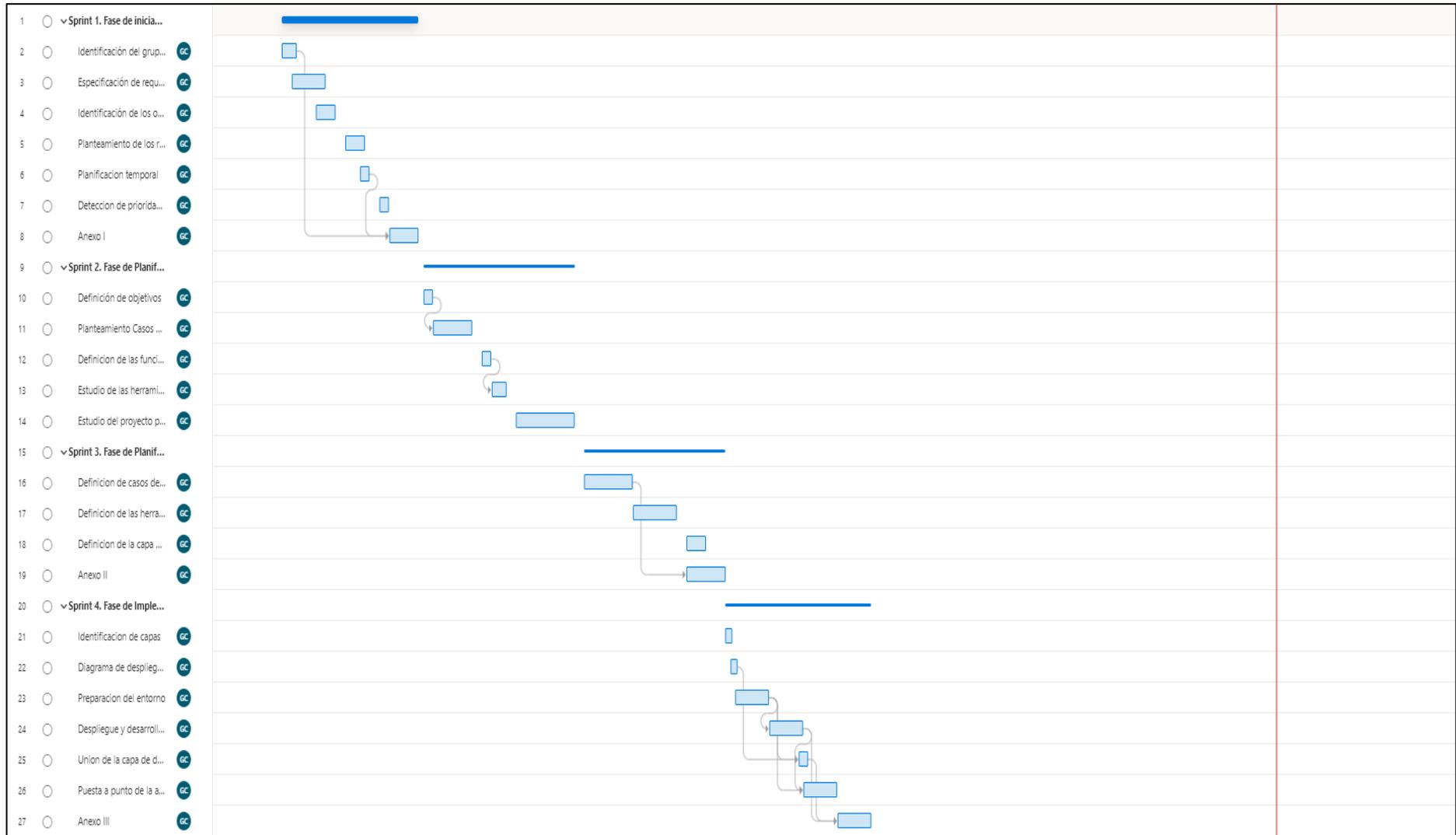


Ilustración 69. Diagrama Gantt 2

6.4. Especificación de requisitos

En esta sección se resume en gran escala el contenido del anexo II, donde se explica detalladamente:

- Organizaciones en el proyecto.
- Participantes en el proyecto.
- Objetivos del sistema.
- Catálogo de requisitos.
 - Requisitos Funcionales → RF
 - Diagrama de paquetes.
 - Definición de los actores.
 - Casos de uso.
 - Requisitos de Información.
 - Requisitos No Funcionales.

Los requisitos del sistema nos permiten estructurar el sistema conociendo las necesidades y condiciones del cliente, y así dar la mejor versión de este. Estos requisitos se han realizado a partir de los objetivos expuestos en el apartado [Objetivos del sistema](#).

Estos requisitos se presentan mediante tablas que siguen la metodología de Durán y Bernárdez, se muestra un ejemplo de cada uno en su respectivo apartado.

6.4.1. Paquetes. RF

Para hacer más fácil el desarrollo del proyecto, facilitando el manejo de los requisitos funcionales, se ha dividido en paquetes, correspondientes con los objetivos del sistema, excepto tiempo real y almacenamiento en la nube que se hacen de manera transparente en las funciones del proyecto. Véase la [Ilustración 53. Paquetes](#).

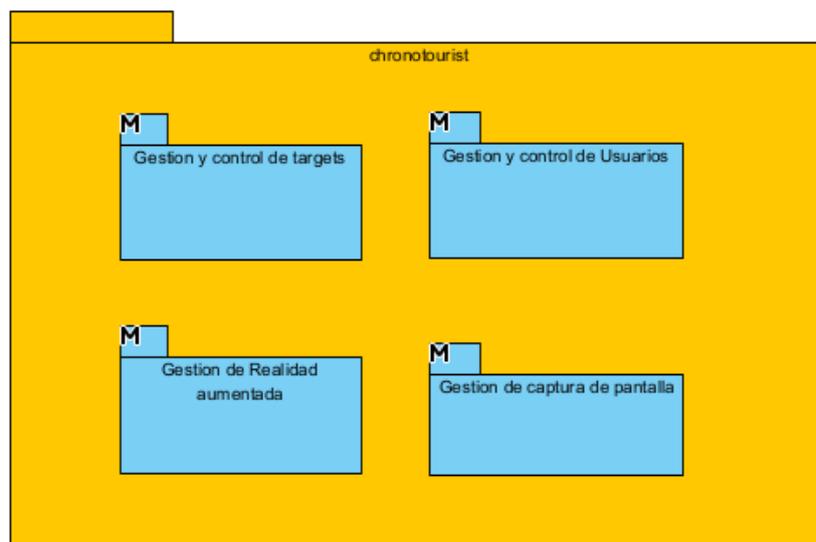


Ilustración 70. Paquetes.

6.4.2. Actores. RF

Se cuenta con 3 actores que interactúan con el sistema, en la [Ilustración 54. Actores](#) se puede ver la herencia de estos:

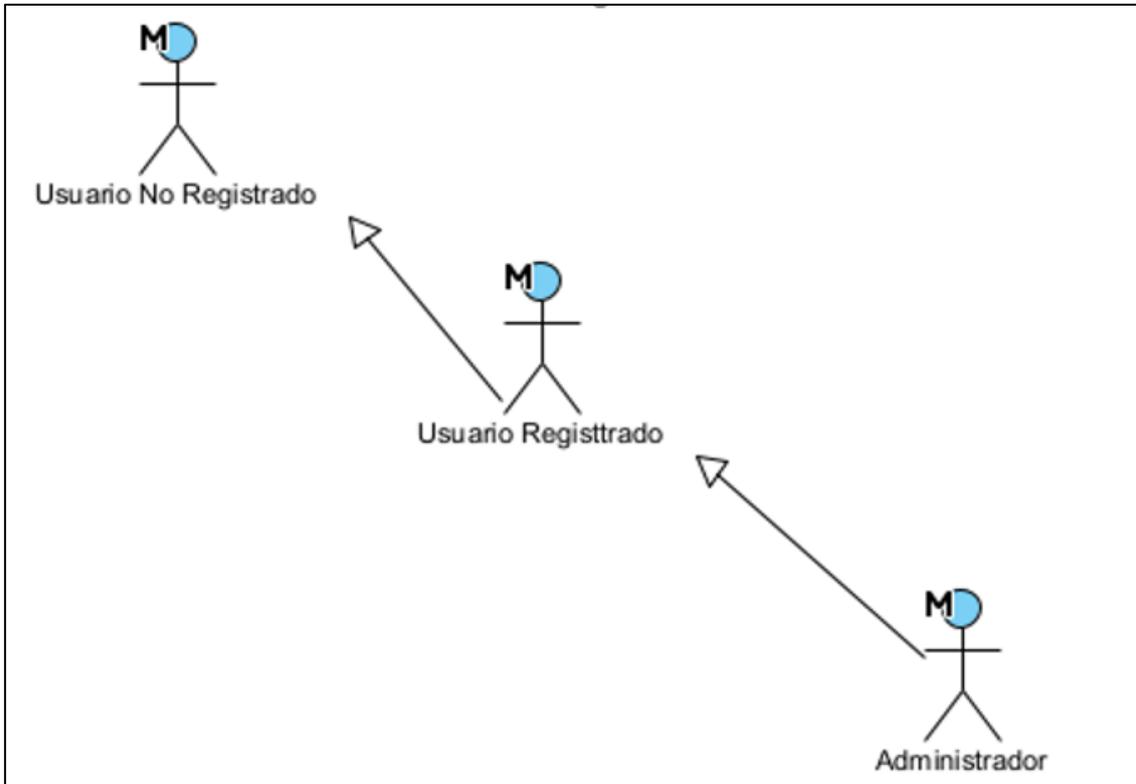


Ilustración 71. Actores

- **Usuario No Registrado:** representa a un usuario que aún no forma parte de la base de datos de la aplicación, es decir, no está registrado y no tiene acceso a gran funcionalidad.
- **Usuario Registrado:** representa a un usuario que forma parte de la base de datos de la aplicación, es decir, está registrado y tiene acceso a gran parte de la funcionalidad.
- **Administrador:** representa a un usuario que forma parte de la base de datos de la aplicación, es decir, está registrado y tiene acceso a toda la funcionalidad de la aplicación.

ACT-0003	Administrador
Versión	Versión-fecha
Autores	<ul style="list-style-type: none">• Guillermo Calvo Suarez
Fuentes	<ul style="list-style-type: none">• Luis Francisco Martín Liras.• Fernando Rodríguez Vicente• Sara Rodríguez González
Descripción	Dicho actor representa a un usuario que forma parte de la base de datos de la aplicación, es decir, está registrado y tiene acceso a toda la funcionalidad de la aplicación.
Comentarios	Ninguno

Ilustración 72. Ejemplo tabla actores

6.4.3. Casos de uso. RF

Se definen los casos de uso mediante diagramas de casos de uso y las tablas mencionadas anteriormente, a continuación, se mostrarán dichos diagramas y un ejemplo de tabla de caso de uso.

Gestión y control de Usuarios.

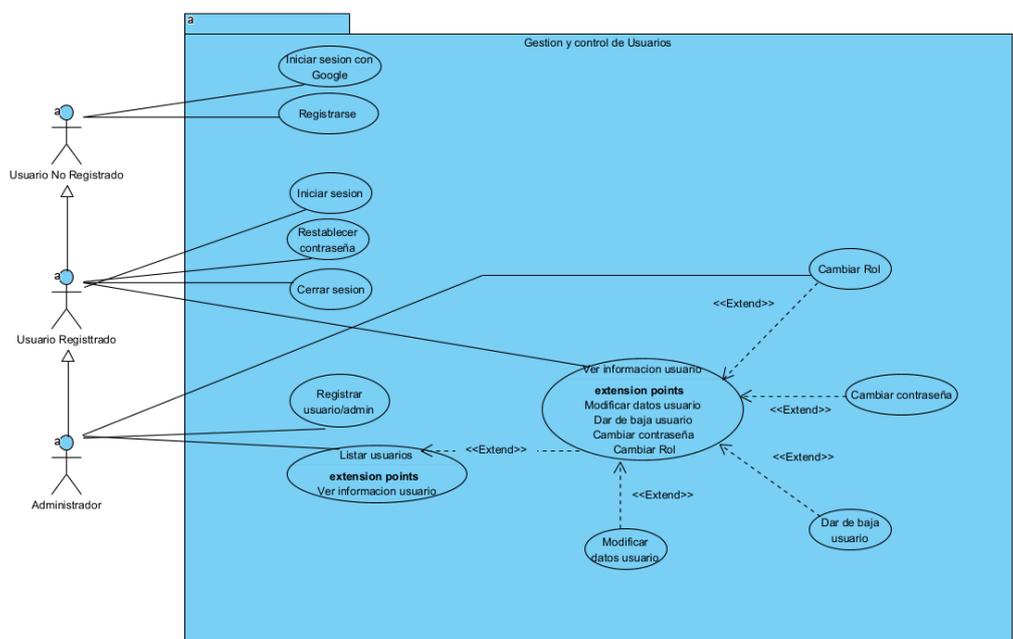


Ilustración 73. Gestión de usuarios

Gestión y control de Targets.

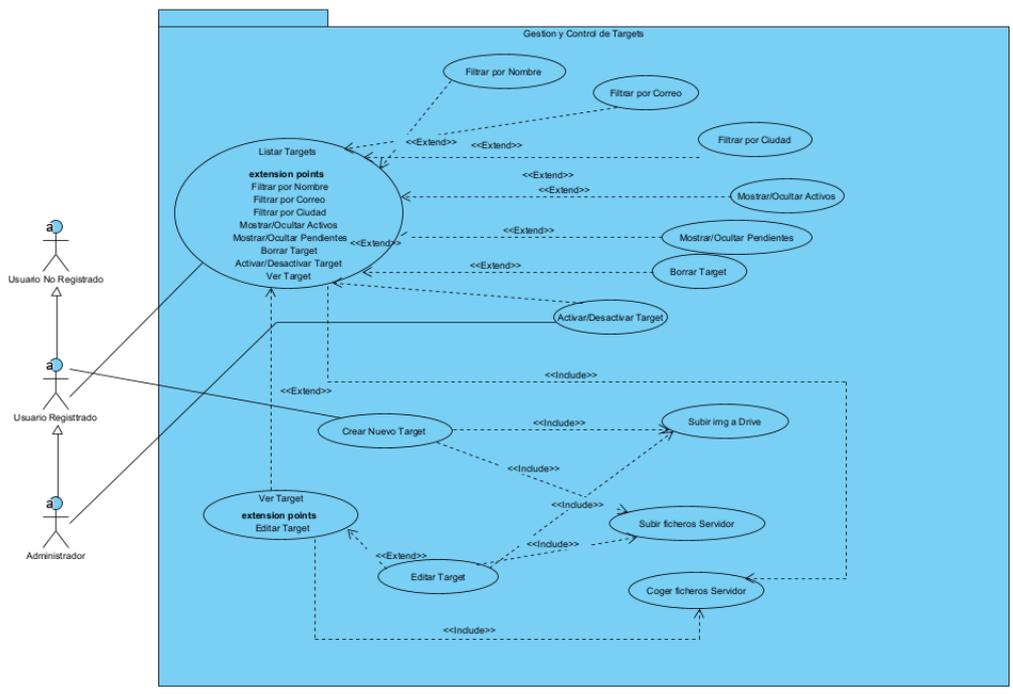


Ilustración 74. Gestión de Targets

Gestión de Realidad Aumentada.

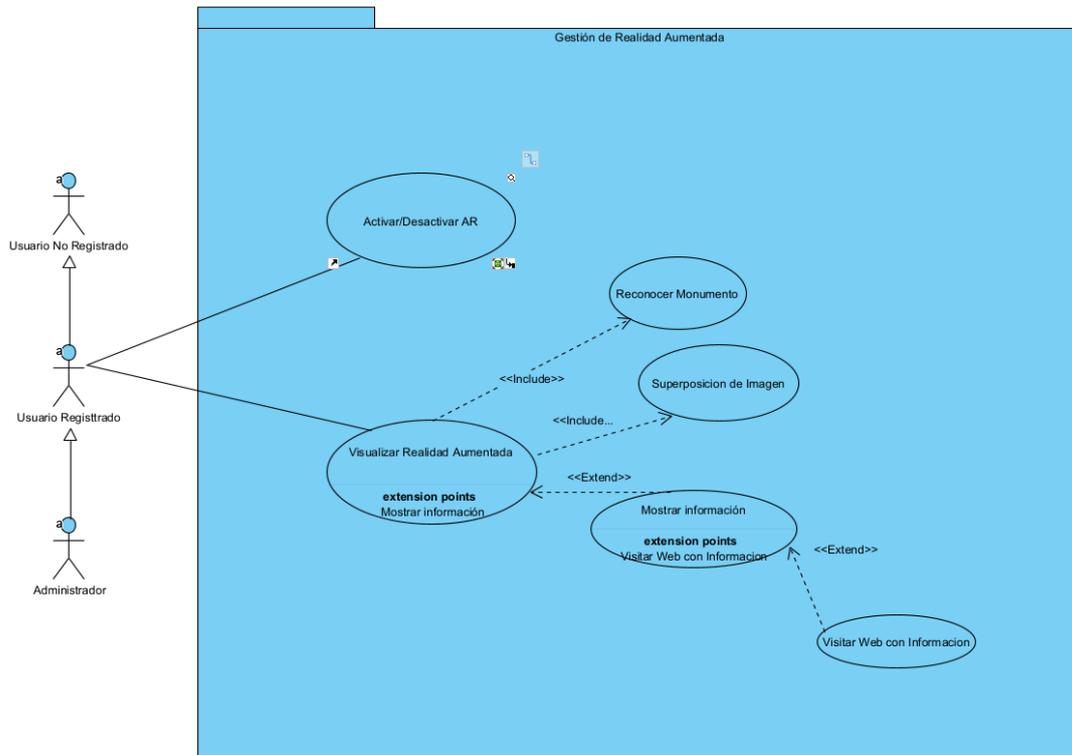


Ilustración 75. Gestión de Realidad Aumentada.

Gestión de captura de pantalla.

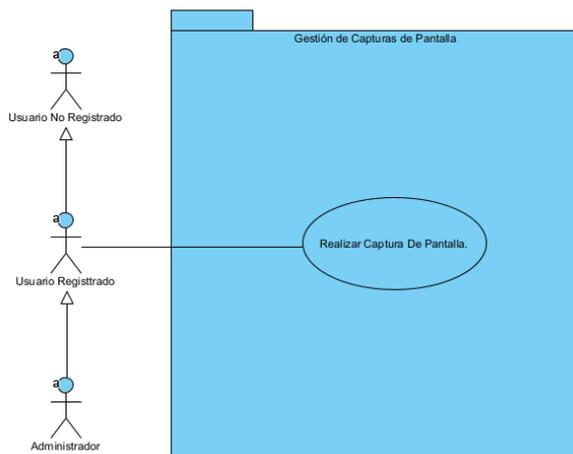


Ilustración 76. Gestión de captura de pantalla.

UC-0016	Filtrar por Correo						
Versión	Versión-fecha						
Autores	<ul style="list-style-type: none"> Guillermo Calvo Suarez 						
Fuentes	<ul style="list-style-type: none"> Luis Francisco Martín Liras, Fernando Rodríguez Vicente Sara Rodríguez González 						
Dependencias	[OBI-0002] Gestión y Control de Targets [IRQ-0001] Información sobre Targets						
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el Usuario Registrado quiere filtrar por correo.						
Precondición	El usuario debe estar logueado .						
Secuencia normal	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El actor Usuario Registrado [ACT-0002] inicia el caso el caso de uso solicitando filtrar por correo la lista de targets.</td> </tr> <tr> <td>2</td> <td>El sistema recoge y muestra la lista de targets filtrada.</td> </tr> </tbody> </table>	Paso	Acción	1	El actor Usuario Registrado [ACT-0002] inicia el caso el caso de uso solicitando filtrar por correo la lista de targets.	2	El sistema recoge y muestra la lista de targets filtrada.
Paso	Acción						
1	El actor Usuario Registrado [ACT-0002] inicia el caso el caso de uso solicitando filtrar por correo la lista de targets.						
2	El sistema recoge y muestra la lista de targets filtrada.						
Postcondición	El usuario visualiza la lista de targets de targets filtrada.						
Excepciones	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>-</td> <td>-</td> </tr> </tbody> </table>	Paso	Acción	-	-		
Paso	Acción						
-	-						
Rendimiento	<table border="1"> <thead> <tr> <th>Paso</th> <th>Tiempo máximo</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	Paso	Tiempo máximo				
Paso	Tiempo máximo						
Frecuencia esperada	5 vez por hora						
Importancia	Vital						
Urgencia	Hay presión						
Estado	Valido						
Estabilidad	Alta						
Comentarios	ninguno						

Ilustración 77. Ejemplo tabla casos de uso

6.4.4. Requisitos de Información

Este apartado presenta los requisitos que nos indican los datos que debe almacenar el sistema, esos datos son los siguientes:

- **Usuarios.**
- **Targets.**
- **Imágenes Drive.**
- **Ficheros.**

Nos ayudaran también a la correcta definición de las Bases de Datos.

IRQ-0002		Información de los Targets	
Versión	Versión-fecha		
Autores	<ul style="list-style-type: none"> • Guillermo Calvo Suarez 		
Fuentes	<ul style="list-style-type: none"> • Luis Francisco Martín Liras. • Fernando Rodríguez Vicente • Sara Rodríguez González 		
Dependencias	<ul style="list-style-type: none"> • Caso de uso Listar Targets. • Caso de uso Filtrar por Nombre. • Caso de uso Filtrar por Correo. • Caso de uso Filtrar por Ciudad. • Caso de uso Mostrar/Ocultar Activos. • Caso de uso Mostrar/Ocultar Pendientes. • Caso de uso Borrar Target. • Caso de uso Activar/Desactivar Target. • Caso de uso Crear Nuevo Target. • Caso de uso Crear Nuevo Target. • Caso de uso Ver Target. • Caso de uso Editar Target. 		
Descripción	El sistema deberá almacenar la información correspondiente a los targets que se registren en el sistema, en concreto:		
Datos específicos	<ul style="list-style-type: none"> • ID del target • Nombre del target • Imagen a reconocer por RA. • Correo del creador del target • Target Width • Ciudad • Calle • Activo/Pendiente • Fecha de creación • Fecha de última modificación • JSON con los siguientes parámetros: <ul style="list-style-type: none"> ○ Posición ○ URL de la imagen a superponer ○ Escala ○ Descripción del monumento ○ Link información del monumento 		
Tiempo de vida	Medio	Máximo	
	-	-	
Ocurrencias simultaneas	Medio	Máximo	
	-	-	
Importancia	Vital		
Urgencia	Hay presión		
Estado	Valido		
Estabilidad	Alta		

Ilustración 78. Ejemplo tabla Requisito Información

6.4.5. Requisitos No Funcionales

En este apartado se detallan los requisitos no funcionales, los cuales se encargan de buscar la calidad del sistema software. Son los siguientes:

- Usabilidad
- Seguridad
- Portabilidad
- Compatibilidad
- Rendimiento
- Almacenamiento
- Fiabilidad
- Escalabilidad

Si se cumplen dichos requisitos obtendremos una aplicación segura y competente.

NFR-0001	Usabilidad
Versión	Versión-fecha
Autores	<ul style="list-style-type: none">• Guillermo Calvo Suarez
Fuentes	<ul style="list-style-type: none">• Luis Francisco Martín Liras.• Fernando Rodríguez Vicente• Sara Rodríguez González
Dependencias	[OBJ-0002] Gestión y Control de Targets [OBJ-0003] Gestión y Control de Realidad Aumentada [OBJ-0001] Gestión y Control de Usuarios [OBJ-0004] Gestión de Capturas de Pantalla
Descripción	El sistema deberá proporcionar simplicidad y fácil manejo a todos los <u>usuarios</u> , para ellos facilitara una interfaz sencilla e intuitiva sin sobrecarga de información.
Importancia	Vital
Urgencia	Hay presión
Estado	Valido
Estabilidad	Alta
Comentarios	Ninguno

Ilustración 79. Ejemplo tabla Requisito no Funcional

6.5. Análisis del sistema

El análisis, refinamiento y estructuración, es el paso que continúa a la especificación de requisitos del sistema. Gracias a ello se consigue una visión más clara del proyecto permitiendo una comprensión más precisa de este y, por tanto, un mejor diseño del sistema.

Para obtener información más detallada se debe consultar el Anexo II-b – *Análisis de Requisitos*.

6.5.1. Modelo de Dominio

El modelo de dominio es una representación donde se muestran las clases conceptuales significativas del dominio del problema planteado en el proyecto. Se centra en las abstracciones relevantes, vocabulario del dominio, etc. Para ello se representan las distintas entidades, atributos, papeles y relaciones, a parte de las restricciones que rigen el dominio del problema.

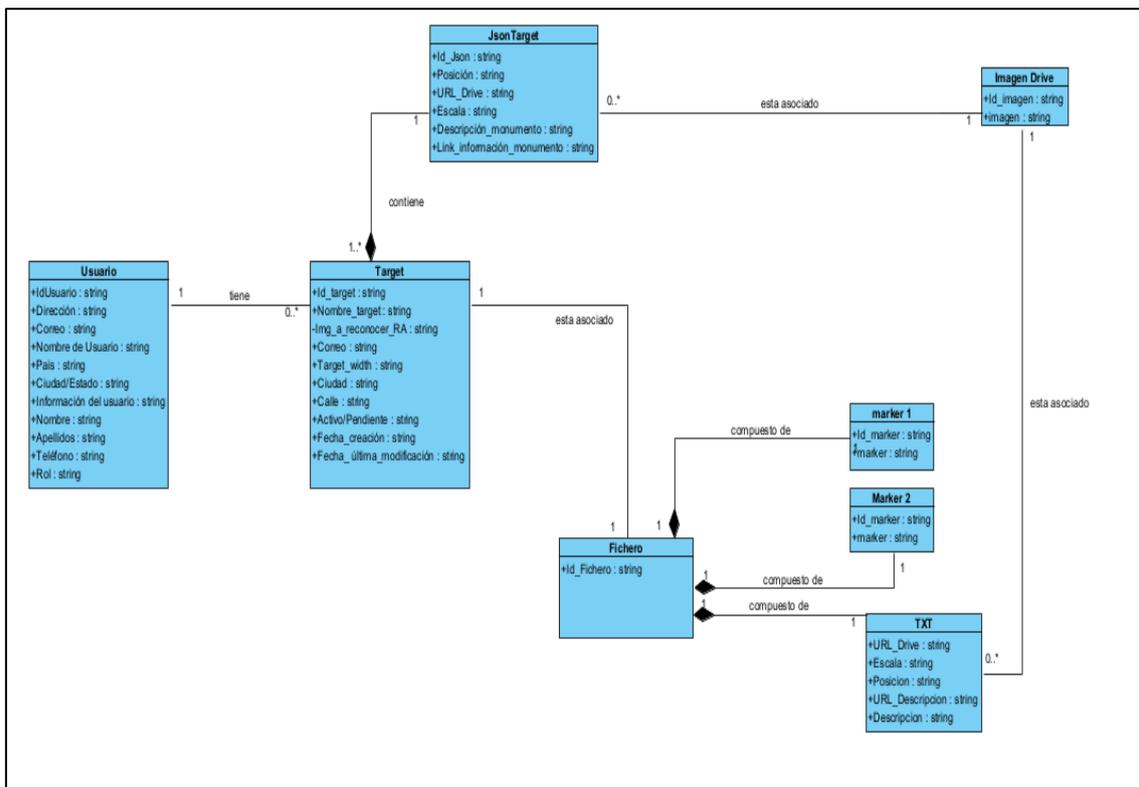


Ilustración 80. Modelo de Dominio

A continuación, se explica brevemente cada una de las clases:

Usuario: Representa a las personas registradas en el sistema, tanto administrador como usuario normal, y que tienen acceso a la información del sistema. No representa a un usuario no registrado.

Atributos:

- Dirección: Dirección donde reside.
- Nombre de Usuario: Nombre de usuario de la cuenta.

- Correo: correo electrónico.
- País: país de residencia.
- Ciudad/Estado: ciudad de residencia.
- Información del usuario: información general del usuario.
- Nombre: nombre real de la persona.
- Apellidos: apellidos de la persona.
- Teléfono: número de teléfono.
- Rol: tipo de cuenta, usuario o administrador.

Target: Representa al Target de la imagen que reconoce con Realidad Aumentada.

Atributos:

- ID del Target: identificador clave del Target.
- Nombre del Target: nombre del Target. Nombre +ciudad + calle también hacen clave.
- Imagen a reconocer por RealidadAumentada: imagen que reconocerá la aplicación, encima de la cual superpondrá otra.
- Correo del creador del Target: correo de la persona que ha subido el Target.
- Target Width: ancho de la imagen.
- Ciudad: ciudad donde se encuentra el monumento asociado al Target.
- Calle: calle donde se encuentra el monumento asociado al Target.
- Activo/Pendiente: indica si el Target esta activado para Realidad Aumentada o no.
- Fecha de creación: fecha de creación del Target.
- Fecha de última modificación: fecha de la última modificación del Target.
- JSON con los siguientes parámetros: Json con parámetros de la imagen a superponer.

JSONTarget: Representa el JSON que contiene el Target, y que lleva los datos necesarios para escalar y posicionar la imagen.

Atributos:

- Posición: posición donde se coloca la imagen.
- URL de la imagen a superponer: URL de descarga de la imagen a superponer.
- Escala: escala de la imagen a superponer.
- Descripción del monumento: descripción del monumento.
- Link información del monumento: link con más información del monumento.

Fichero: Representa la carpeta asociada al Target, que contiene 3 ficheros:

Marker 1: representa una copia de la imagen que reconoce Vuforia codificada y que tiene otros usos que no son el reconocimiento, como la visualización de esta en la interfaz web.

Marker 2: representa una copia de la imagen que superpone Vuforia y que tiene otros usos que no son la superposición, como la visualización de la misma en la interfaz web.

TXT: representa una copia de los atributos del JSONTarget:

Atributos:

- Url Drive: URL de descarga de la imagen a superponer.
- Escala: escala de la imagen a superponer.
- Posición: posición donde se coloca la imagen.

- Url descripción: Link con más información del monumento.
- Descripción: descripción del monumento.

Imagen Drive: Representa la imagen que se sube a Drive, la cual se superpone en la Realidad Aumentada.

Atributos:

- Imagen: imagen en formato JPG o PNG.
- ID de la imagen (URL): identificador Clave de la imagen.

6.5.2. Diagrama de paquetes

En este apartado veremos la descomposición en paquetes de análisis reducido. Muestra las dependencias que tienen los paquetes entre ellos.

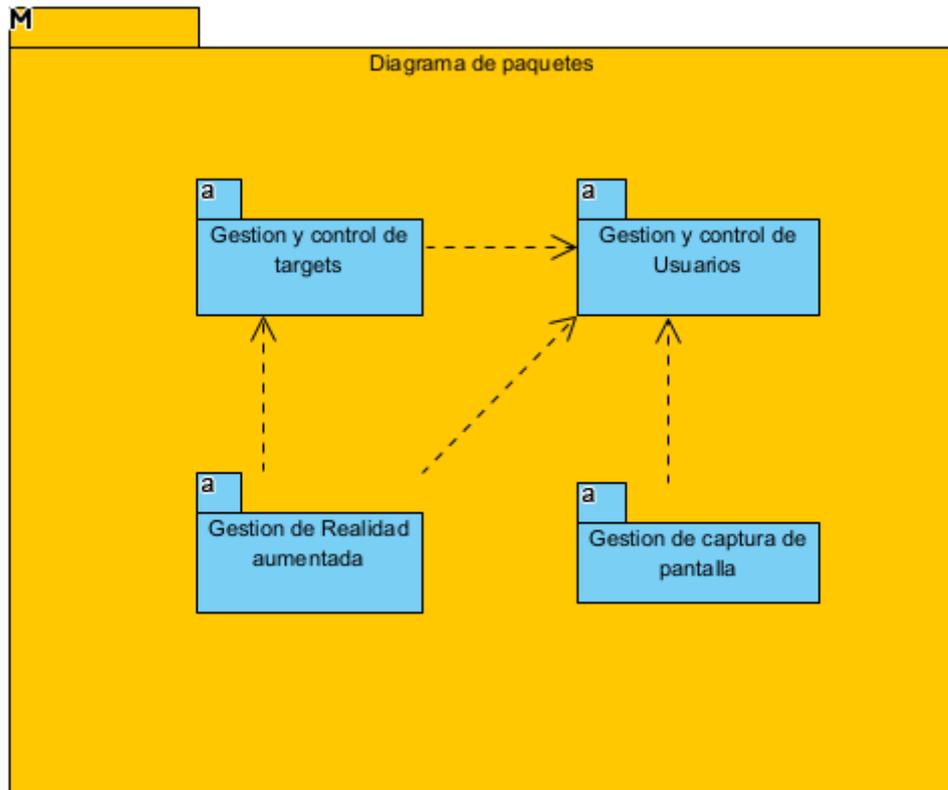


Ilustración 81. diagrama de paquetes reducido

Los paquetes de análisis son:

- **Gestión y control de Usuarios:** realiza la gestión de usuarios, como por ejemplo la baja o creación de estos.
- **Gestión y control de Targets:** realiza la gestión de Targets, como por ejemplo la creación o eliminación de estos.
- **Gestión de Realidad Aumentada:** realiza la gestión de la Realidad Aumentada, como por ejemplo el reconocimiento o superposición del monumento.
- **Gestión de captura de pantalla:** realiza la gestión de capturas de pantalla, como por ejemplo sacar captura de pantalla.

A continuación, se muestra el diagrama de paquetes más detallado, con los componentes de cada paquete, mostrando la interacción entre los elementos del análisis.

Se puede decir que es la primera vista de la arquitectura del sistema.

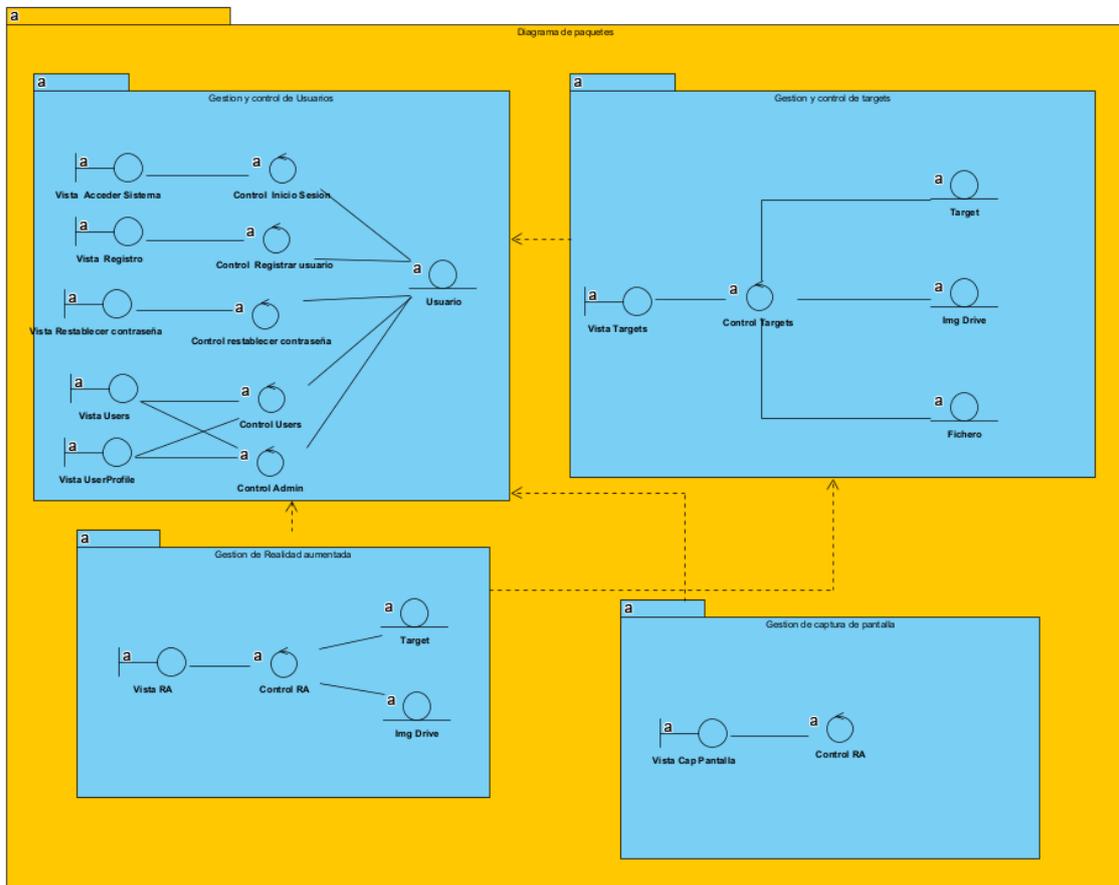


Ilustración 82. Diagrama de paquetes extenso.

6.5.3. Diagramas de casos de uso

En este apartado se mostrará la realización de un caso de uso a nivel de análisis, en el que se puede ver como interactúa el actor con la vista, la vista con el controlador y el controlador con el modelo. Más adelante estos diagramas tendrán su equivalencia a nivel de diseño, donde se verá la interacción con los componentes de los paquetes de diseño.

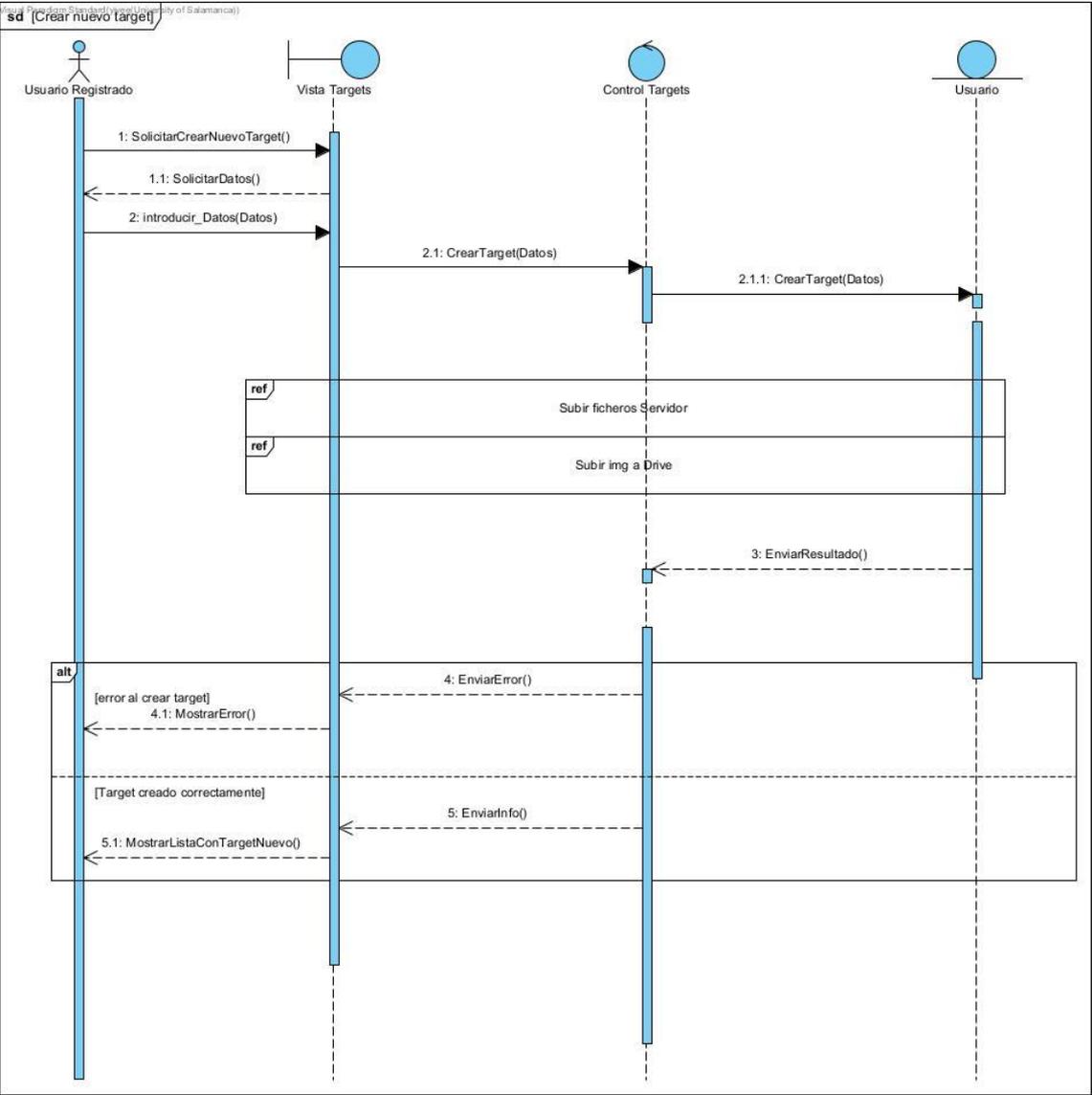


Ilustración 83. Ejemplo caso de uso análisis.

Para consultar la información completa de análisis del sistema consultar el documento Anexo II-b.

6.6. Diseño del sistema

Nos adentramos ya en el desarrollo del sistema software, es decir, le vamos dando la forma que queremos que tenga finalmente. Esta parte es muy importante, se van a tomar decisiones sobre la arquitectura, los datos, la interfaz y los detalles procedimentales del software.

En esta sección veremos los elementos más importantes del diseño del sistema, si se quiere ver la información más detallada debe consultar el *Anexo III- Diseño del sistema*.

6.7.1 Diseño arquitectónico

Partiendo de la arquitectura que obtuvimos en el proceso de análisis, en este apartado se realizará un estudio más detallado de la composición y estructura del sistema, mejorando así la calidad arquitectónica, y, por ende, el resultado final del proyecto.

El proyecto está dividido en dos partes:

UNITY:

Está formada por el entorno **Unity**, dicho entorno nos va a hacer tomar soluciones un poco alejadas de lo convencional, pero necesarias para el correcto funcionamiento, esto se debe a que es un entorno de desarrollo de videojuegos.

Para que sea más simple el desarrollo de la aplicación, Unity la divide en escenas, definiendo así el flujo del programa. En nuestro caso, la aplicación está formada por dos escenas, la principal, donde estará el manejo de toda la Realidad Aumentada, y una secundaria, donde estará el inicio de sesión y registro.

Dentro de cada una de las escenas, se usa el patrón MVP [26], separando así, la lógica de la vista con la cual interaccionamos, y dejando toda la funcionalidad al presentador:

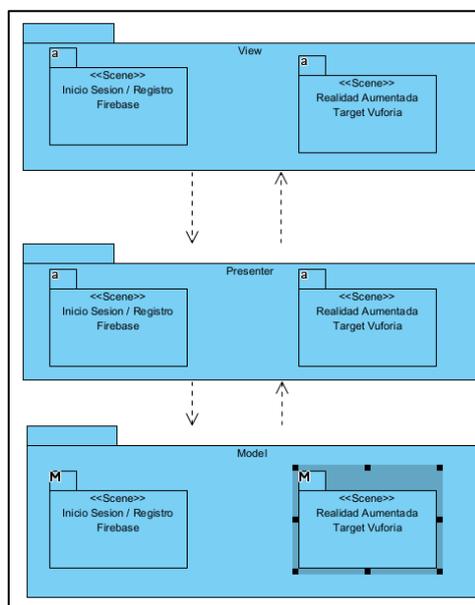


Ilustración 84. arquitectura unity

La arquitectura MVP, Modelo-Vista-Presentador, trata de delegar toda la funcionalidad en el presentador, que será el encargado de actualizar el modelo cuando ocurre un evento en la vista, y también de actualizar la vista en caso de que el modelo lo indique, es decir, actúa como un intermediario.

La vista se encargará de mostrarle una interfaz al usuario y transmitir al presentador los eventos y datos necesarios.

El modelo define los datos sobre los que actúa el presentador, en este caso serán los Targets de Vuforia y los usuarios de Firebase, cada uno en su respectiva escena.

Como podemos observar, MVP, es muy parecido MVC, Modelo-Vista-Controlador, con la diferencia que el presentador hace de intermediario, lo cual entre vista y modelo no va a haber flujo de información.

Para proyectos en Unity resulta mucho más cómodo utilizar esta arquitectura, para desacoplar por completo toda la lógica de la vista, ya que Unity no funciona con eventos, si no que todas las comprobaciones y validaciones van a nivel de código.

Recaltar que el presentador realmente es un controlador, pero se le llama así para diferenciar entre MVP y MCV [27].

WEB:

Formada por la interfaz web y sus controladores y clases.

También es muy interesante enfocar la parte web de la misma manera, con el patrón MVP, consiguiendo presentar los datos en la vista de una manera mucho mejor y más cómoda.

Cuando sea necesario el flujo de datos entre la vista y el modelo, el presentador se encargará de formatearlos de la manera correcta y presentarlos o almacenarlos. En este proyecto es muy importante tener un formateo correcto ya que hay estructuras muy complejas con sus respectivos JSON y mucho flujo de información de diferente tipo, añadiéndole también los requisitos de las herramientas utilizadas para la Realidad Aumentada.

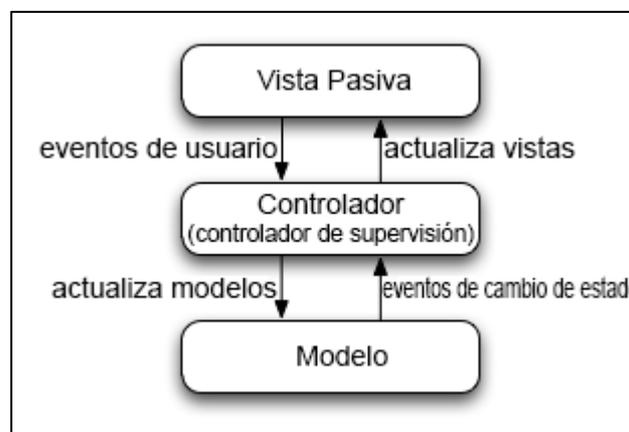


Ilustración 85. Arquitectura del sistema.

6.6.1.1. Subsistemas de diseño

Estos subsistemas representan la división del sistema en paquetes de diseño, en los cuales se especifican los componentes del sistema, mostrando también las relaciones entre ellos.

En este apartado veremos la descomposición en paquetes de diseño.

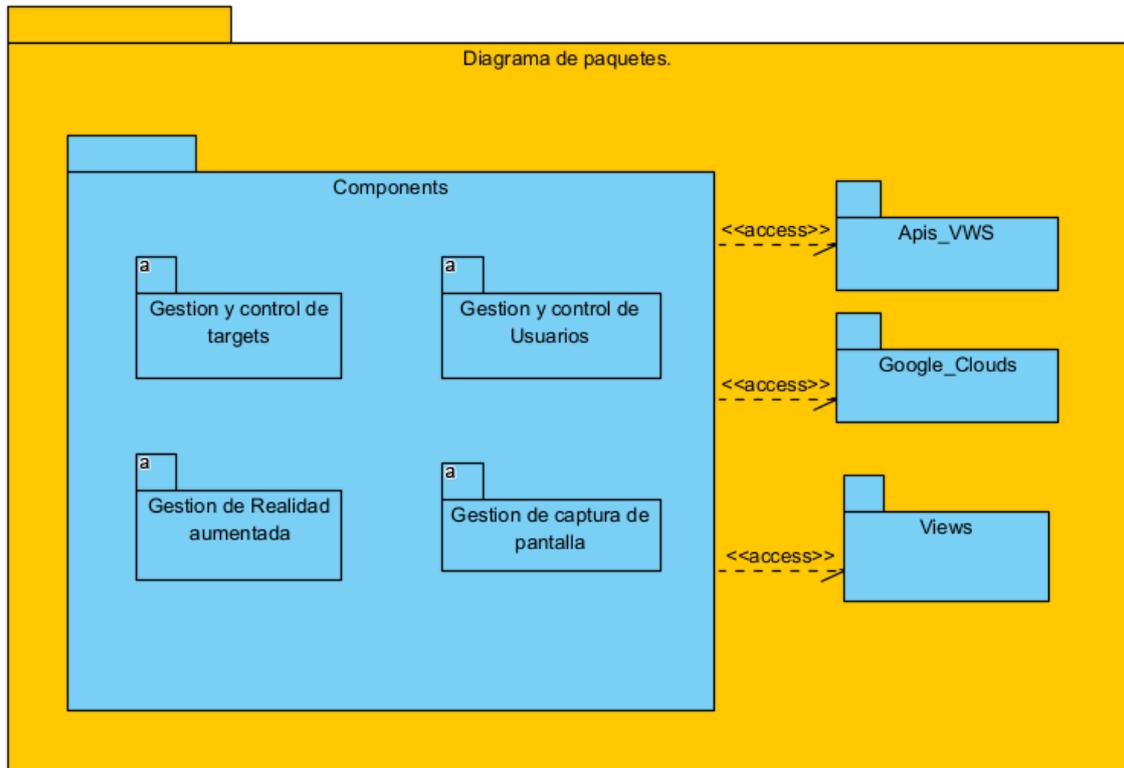


Ilustración 86. Diagrama de paquetes.

Paquete Views

Compuesta por 3 componentes, AdminView y UserView, encargadas de las homes del administrador y del usuario, y AboutUs, encargada de la información del proyecto y sus participantes.

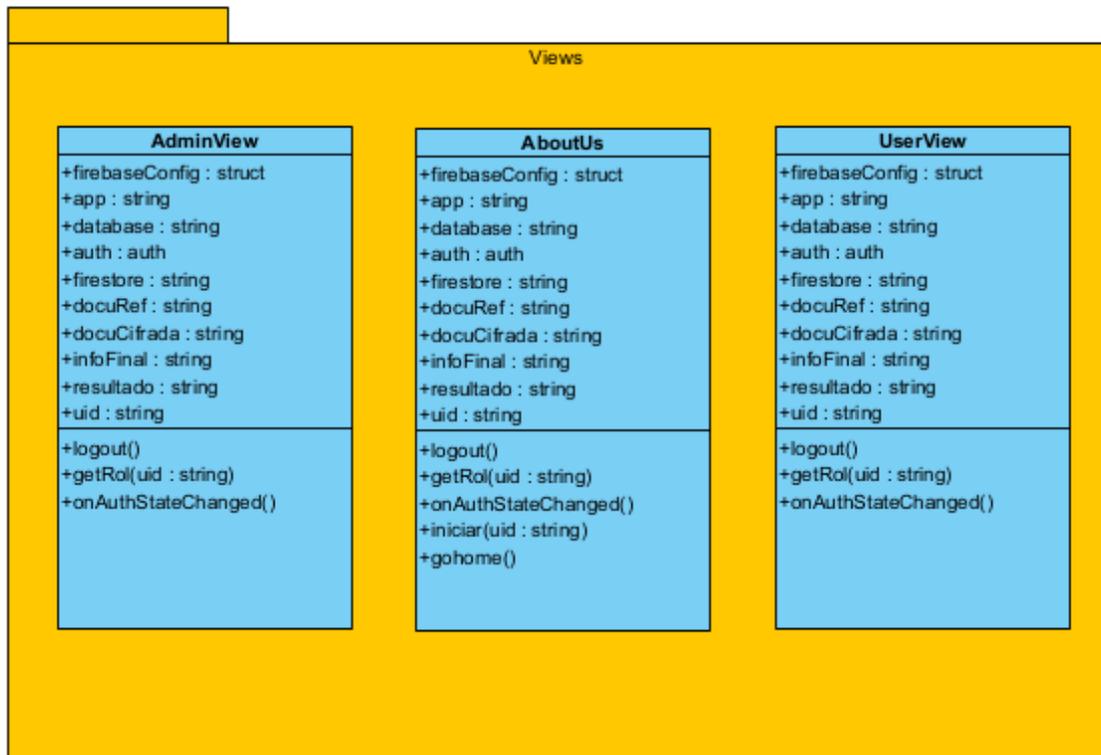
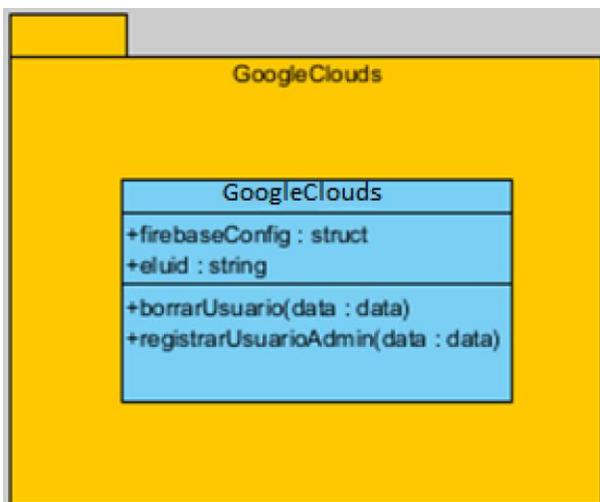


Ilustración 87. Paquete Views.

Paquete GoogleClouds



Formada por las GoogleClouds, encargadas de ejecutar las funciones necesarias desde el servidor.

Ilustración 88. Paquete GoogleClouds

Paquete Apis VWS

Conjunto de ficheros que contienen las diferentes funcionalidades de *VuforiaWebServices*, *PostNewTarget*, *DeleteTarget*, *SetTargetActive*, *GetTarget*, *GetAllTargets* y *UpdateTarget*, y sus respectivas APIS para el correcto formato y funcionamiento con Vuforia.

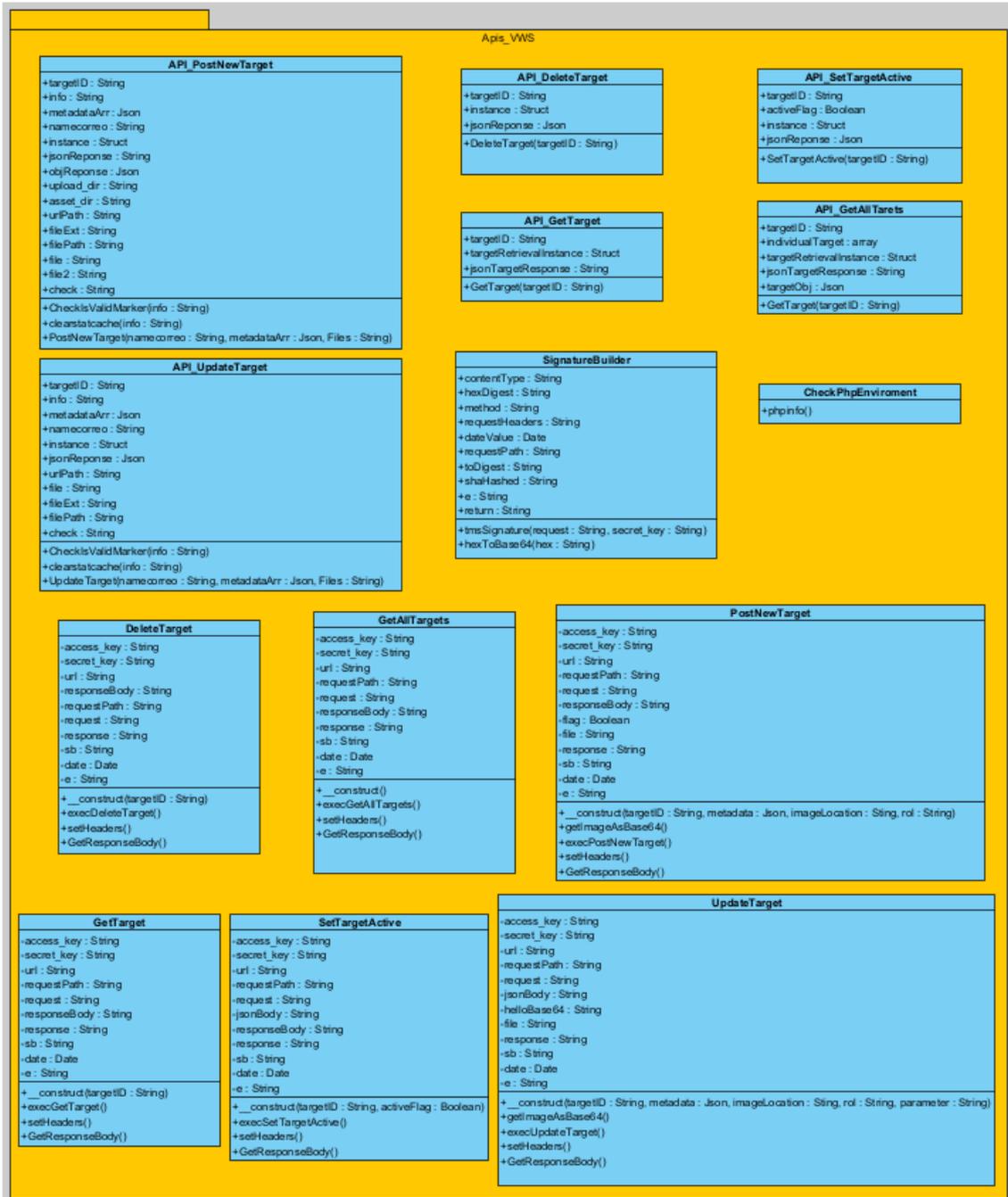


Ilustración 89. Paquete Apis_VWS

Paquete Componentes

Está compuesto por los elementos de la lógica del sistema como gestión y control de Targets, gestión y control de usuarios, gestión de Realidad Aumentada y gestión de captura de pantalla. También incluye el acceso al sistema. Se detallará cada uno a continuación.



Ilustración 90. Paquete componentes

Paquete Gestión y Control de Targets

Contiene index y se encarga de llevar el control de los Targets, incluyendo todas las funciones transparentes al usuario, como puede ser, subir la imagen a superponer a Drive, subir ficheros a servidor..., a parte de las principales como subir/borrar/editar/filtrar Targets, entre otras.

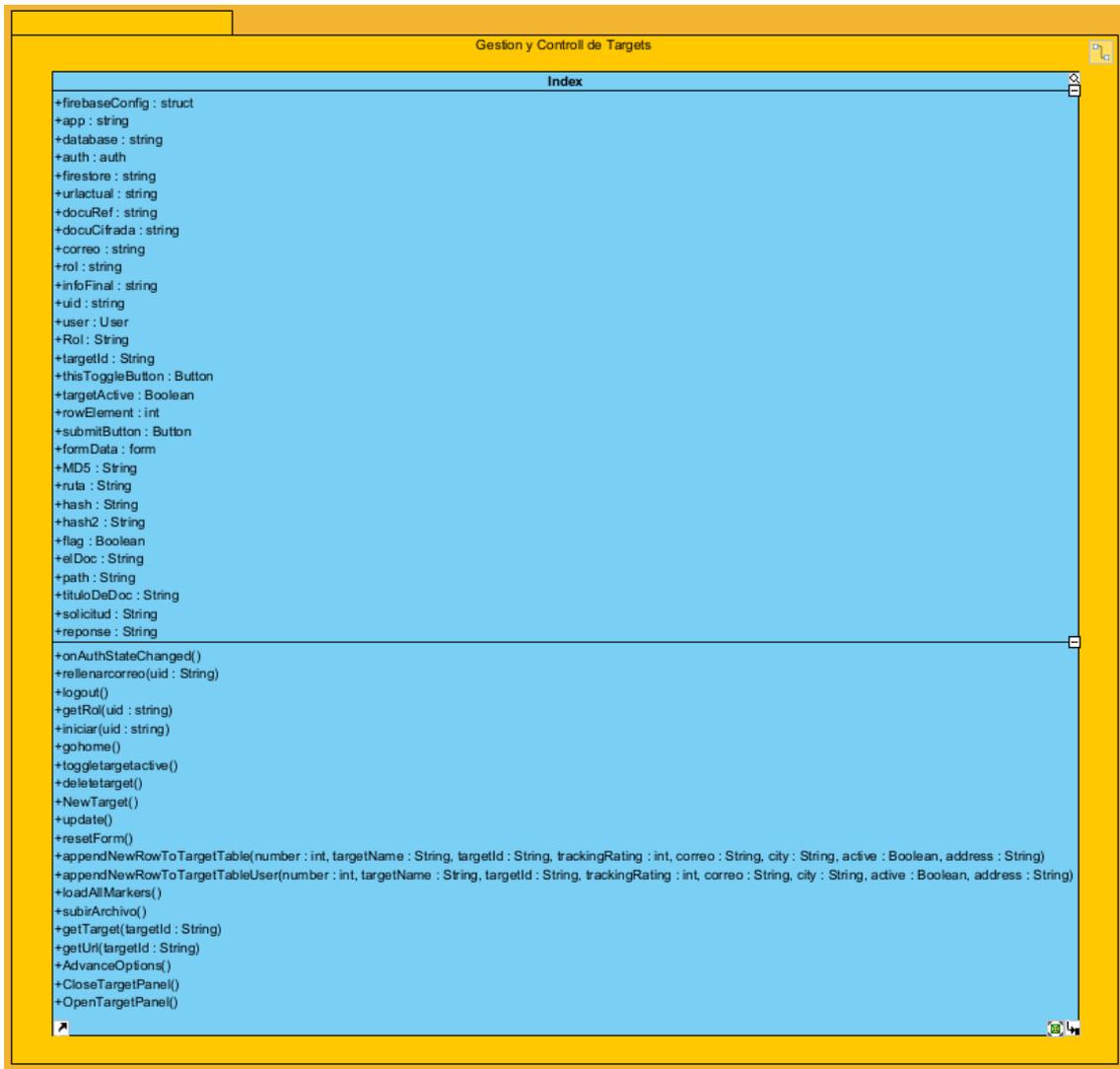


Ilustración 91. Paquete Gestión y control de Targets

Paquete Gestión y Control de Usuarios

Está compuesta por 6 componentes, index, Register, UIManager y AuthManager, encargados de la parte de autenticación del usuario tanto en la parte web (los dos primeros), como en la parte Unity (los otros dos). Users, da el control de los usuarios a los administradores, y UserProfile, controla los datos del perfil del usuario.



Ilustración 92. Paquete gestión y control de usuarios

Paquete Gestión de Realidad Aumentada

Está compuesto por ChronoTurist y se encarga de la Realidad Aumentada en la APP móvil, desde el reconocimiento del Target hasta la superposición sobre el monumento reconocido.



Ilustración 93. Gestión Realidad Aumentada.

Gestión de Captura de Pantalla

Compuesto por ChronoTurist, y es el encargado de las capturas de pantalla sobre la Realidad Aumentada.

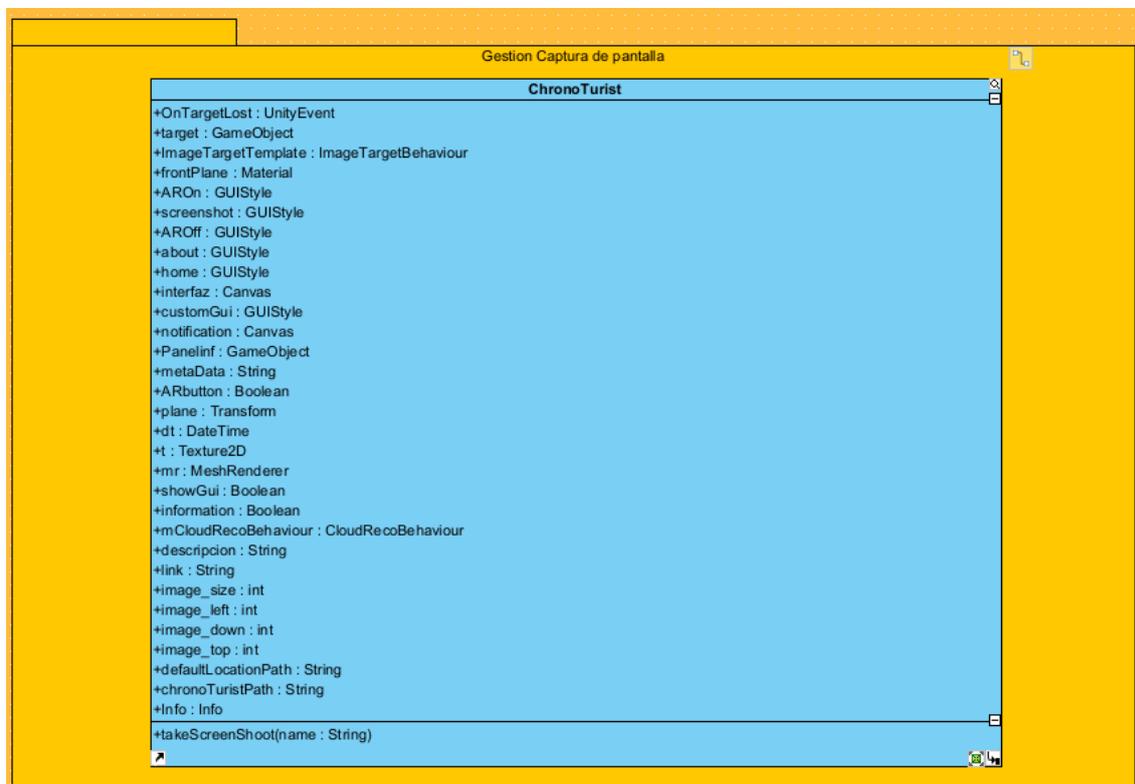


Ilustración 94. Gestión de captura de pantalla

Diagrama de Paquetes completo

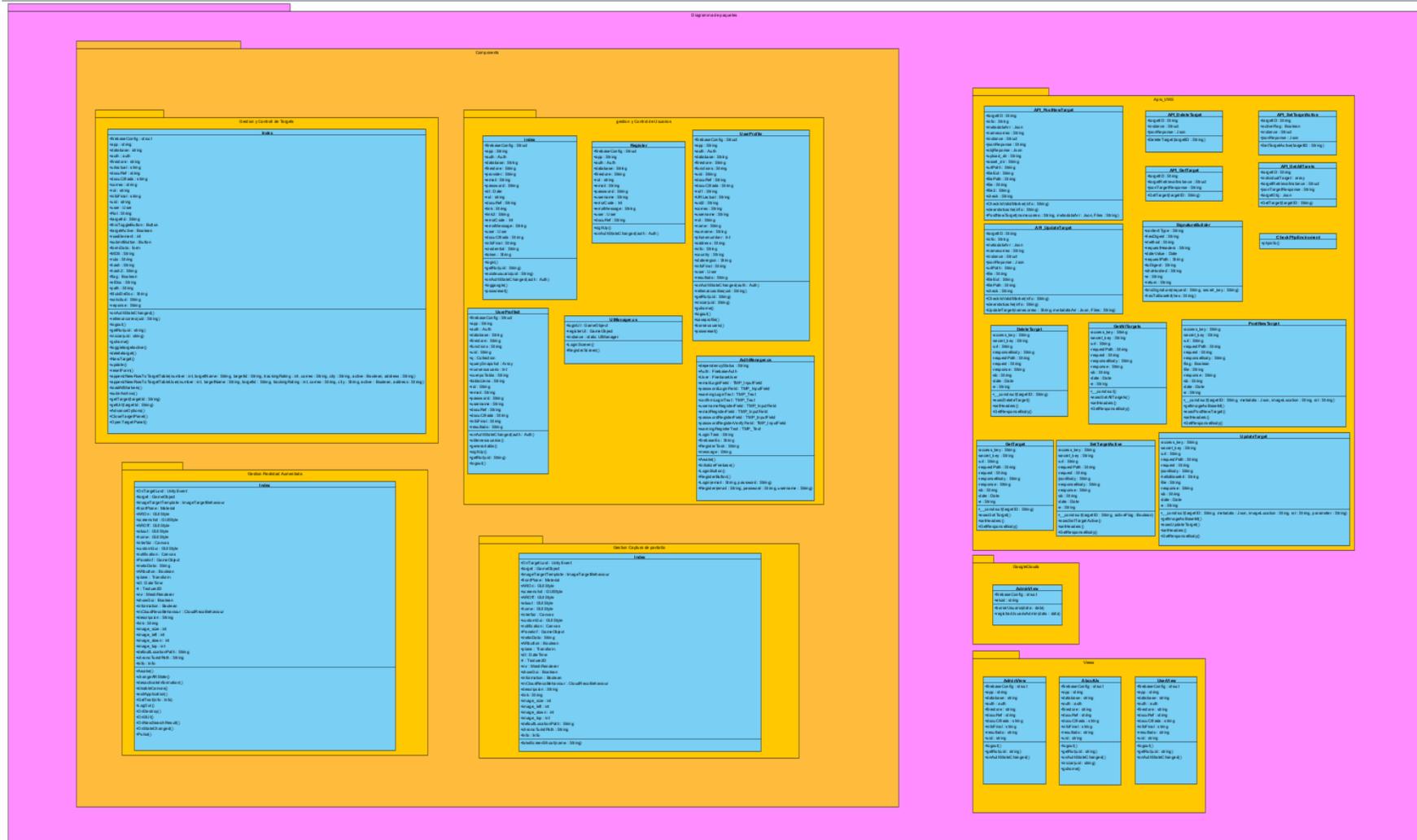


Ilustración 95. diagrama de paquetes completo

6.6.1.2. Diagrama de despliegue

El diagrama de despliegue es un tipo de diagrama que se utiliza para modelar la distribución física de los artefactos software. Cada uno de los nodos que representa el diagrama es un elemento hardware o software que forma parte del sistema:

- **Móvil, Tableta, PC:** dispositivos desde los cuales se accede al servidor.
- **Servidor:** servidor de alojamiento de la aplicación web.
- **APIS:** nodo que comunica el servidor con la funcionalidad de Vuforia.
- **Vuforia, Firebase, Drive:** bases de datos del sistema.
- **Aplicación móvil:** aplicación de Realidad Aumentada.

Muestra la arquitectura del sistema:

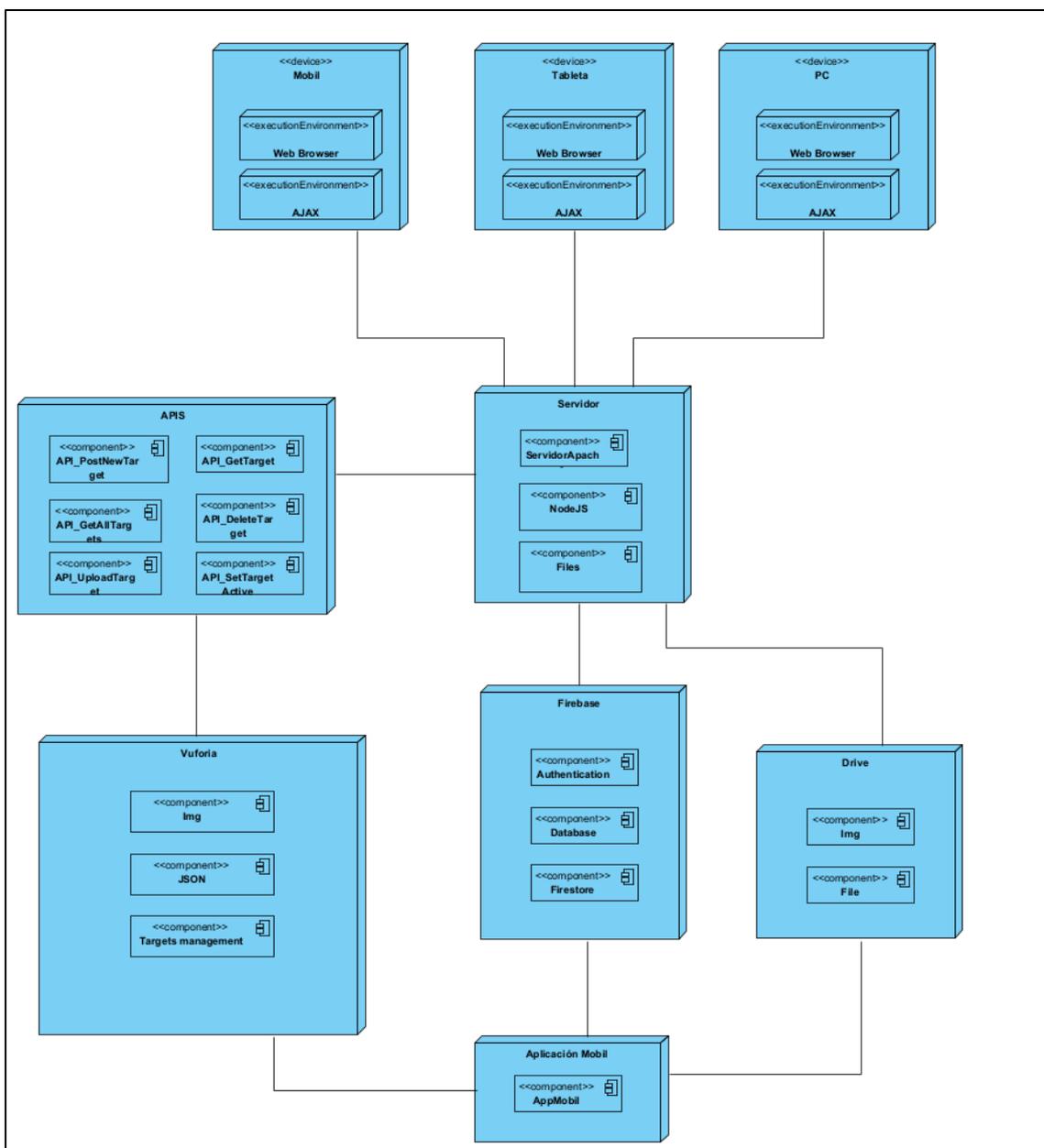


Ilustración 96. Diagrama de Despliegue.

6.6.1.3. Diagramas de Secuencia

En este apartado se muestra como interactuar el actor con los distintos componentes del sistema y los mensajes con los que se comunican, y como interactúan entre si dichos componentes.

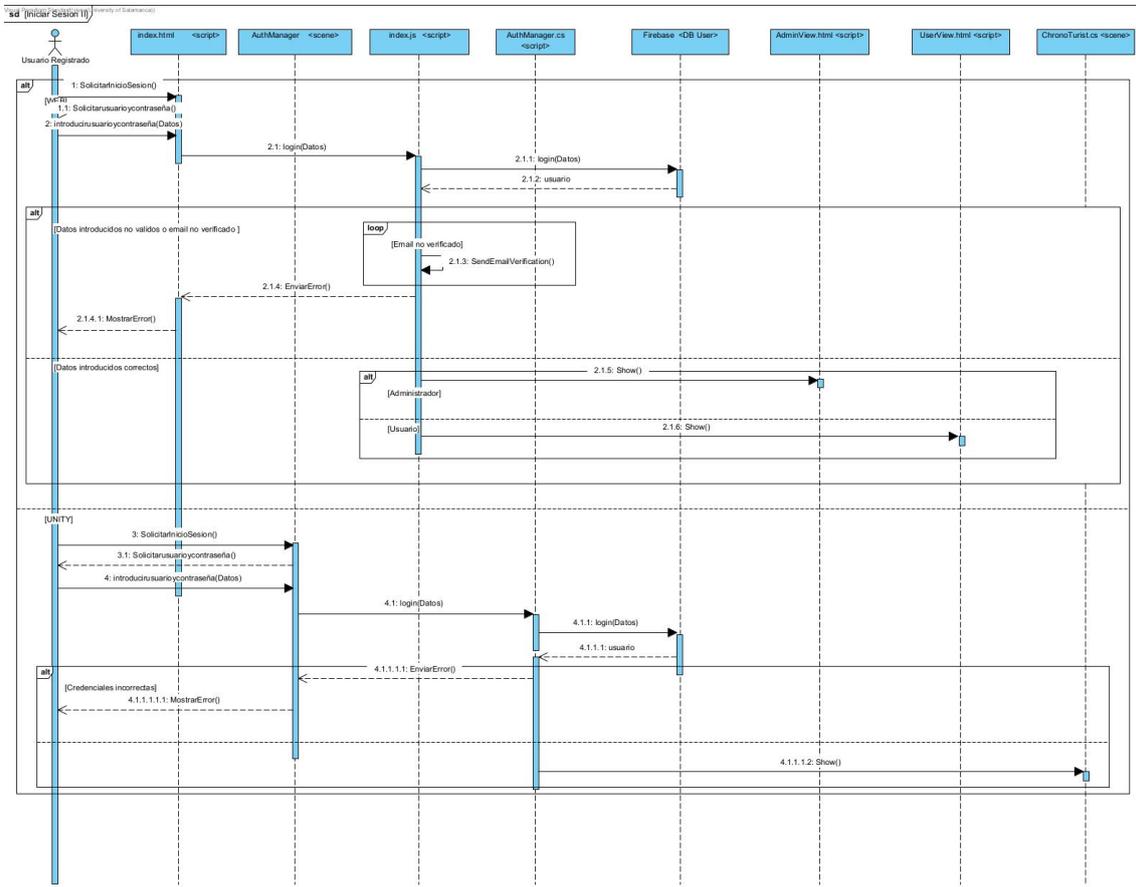


Ilustración 97. Ejemplo caso de uso diseño.

6.6.2. Diseño de los datos

En este apartado veremos la estructura del almacenamiento de los datos del proyecto, los principales datos a guardar, ya definidos en los requisitos de información y como se consigue el modelo relacional entre ellos.

Dada la funcionalidad del proyecto, y los requisitos de ciertas herramientas, los datos se guardan en diferentes bases de datos, cada una tendrá su API funcional para lograr este almacenamiento. A continuación, explicaré cada uno de ellos:

- **Usuarios:** los usuarios se guardarán en una Base de Datos a tiempo real, Firebase. Firebase es una Base de Datos adquirida por Google, en 2011, con la que pretenden facilitar el desarrollo de aplicaciones web y móviles, ofreciendo diferentes facilidades a la hora de manejarla. Además, tiene la posibilidad de desarrollo de aplicaciones de forma gratuita hasta cierto uso.

Atributos:

- Dirección
- Nombre de Usuario
- Correo
- País
- Ciudad/Estado
- Información del usuario
- Nombre
- Apellidos
- Teléfono
- Rol

- **Targets:** los Targets se guardarán en la Base de Datos propia de Vuforia. Vuforia es una herramienta (SDK), que va de la mano con Unity, muy útil a la hora de la creación de Realidad Aumentada. Es capaz de realizar el reconocimiento de imágenes y superposición de imágenes, consiguiendo así la famosa Realidad Aumentada. El almacenamiento en dicha Base de Datos, al estar orientado a la Realidad Aumentada, nos brinda facilidad a la hora del tratamiento de imágenes y sus respectivos JSON.

Atributos:

- ID del Target
- Nombre del Target
- Imagen a reconocer por RA.
- Correo del creador del Target
- Target *Width*
- Ciudad
- Calle
- Activo/Pendiente
- Fecha de creación
- Fecha de última modificación
- JSON con los siguientes parámetros:
 - Posición
 - URL de la imagen a superponer
 - Escala

- Descripción del monumento
 - Link información del monumento
- **Imágenes Drive:** Vuforia solo nos da la capacidad de subir la imagen a reconocer, en cambio, la imagen a superponer debe estar definida con un link de descarga, lo cual genera problemas como, por ejemplo, que el usuario publique un Target con un link de su propio Google drive y más adelante borre la foto, quedando un Target sin imagen a superponer en Vuforia.
Dado que todos los datos del proyecto van a estar en la nube, las imágenes a superponer se guardarán automáticamente en un drive de la propia aplicación, generando el url de descarga para incluirlo en el JSON que subirá a Vuforia, este proceso se hace de forma transparente al usuario.
Atributos:
 - Imagen en formato JPG o PNG
 - ID de la imagen (URL)
 - **Ficheros:** Vuforia, como todo, tiene inconvenientes. Si hacemos una solicitud a la base de Datos, no nos devuelve toda la información que nosotros subimos, por tanto, se ha desarrollado una funcionalidad para tener a mano esa información, y entregársela al sistema cuando sea necesario.
Esta información se almacenará en el servidor, accesible por todos los usuarios.
Atributos:
 - Marker 1, imagen a reconocer por RA codificada.
 - Marker 2, imagen a superponer codificada.
 - Url.txt, contiene información correspondiente a los Targets:
 - Url Drive
 - Scala
 - Posición
 - Url descripción
 - Descripción

La aplicación está dividida en 4 bases de datos para mejorar la funcionalidad. El funcionamiento es sencillo, el objetivo es hacer visible al cliente la información necesaria, para ello, es el propio programa quien convierte las 4 bases de datos en un modelo relacional.

Cuando subimos un Target, el sistema no sólo lo subirá a Vuforia. La imagen que se superpone no es subida con formato JPG o PNG, sino que es referenciada mediante un link de descarga, para ello, se crea la funcionalidad de subir una imagen a drive consiguiendo así dicho link (procedimiento más detallado en el apartado de Diseño Procedimental), con el cual se consigue la relación entre Google Drive y Vuforia.

Por otro lado, no podemos hacer las solicitudes de imagen y Json de todos los Targets cada vez que solicitamos la lista, además la versión gratuita de Vuforia está limitada. Para ello se ha diseñado un mecanismo consiguiendo la funcionalidad de almacenar la imagen a reconocer, la imagen a superponer y algunos atributos de la Realidad Aumentada. La manera de relacionar estos tres ficheros con el Targets es sencilla, simplemente se crea una carpeta en la Base de Datos del servidor, con el nombre del Target, el cual, se busca cada vez que Vuforia devuelve la

lista de los ids de los Targets, sin ser necesaria la devolución de las dos imágenes y el archivo Json de cada Target desde Vuforia.

En resumen, se consigue la unión de estas 4 bases de datos:

- Guardando el correo del usuario que subió el Target en Vuforia, relacionando Targets (Vuforia) con usuarios (Firebase).
- Guardando el Url de descarga de la imagen de Drive en Vuforia y en el servidor, relacionando Targets (Vuforia) con Url de las imágenes (Drive) y ficheros (BD Servidor) con Url de las imágenes (Drive).
- Guardando el id del Target en el Server, relacionando Targets (Vuforia) con ficheros (Server).

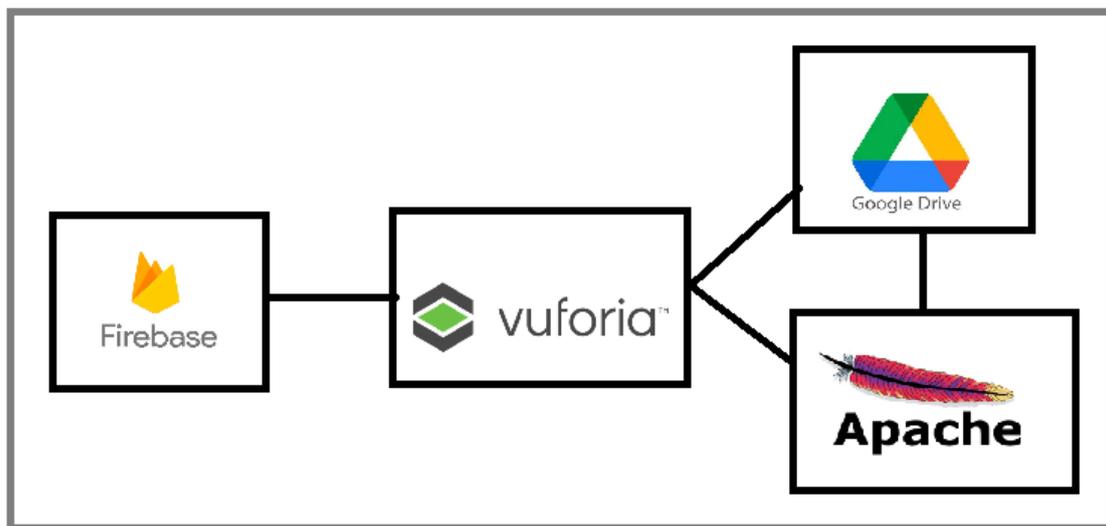


Ilustración 98. Esquema Bases de Datos.

Explicaré el procedimiento de forma más detallada en el apartado de Diseño Procedimental, viendo cómo funciona secuencialmente.

6.6.2.1. Diagrama de base de datos

Como se ha explicado en varios puntos anteriores, los datos están en 4 bases de datos separadas, pero se consigue la base de datos relacional que se muestra en el siguiente diagrama de bases de datos del sistema:

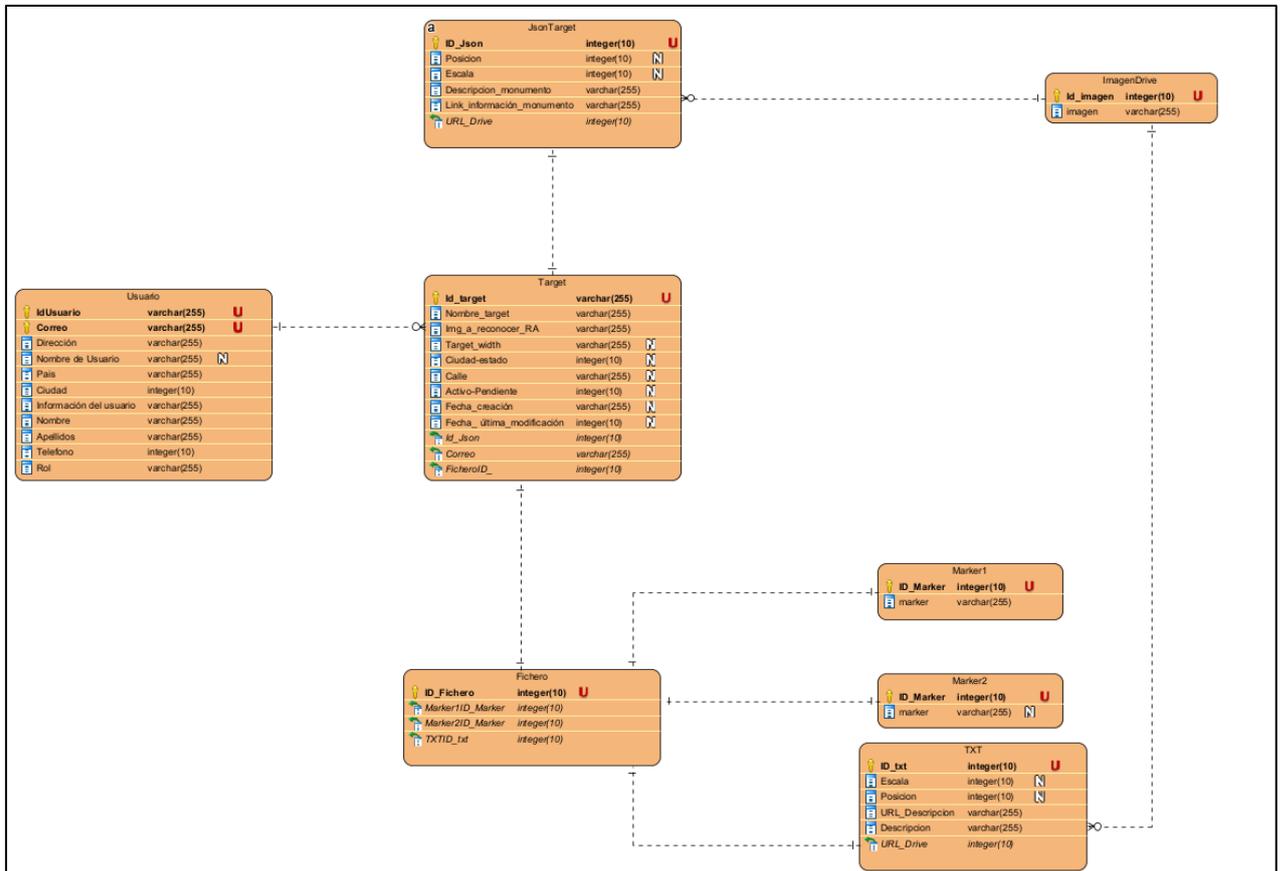


Ilustración 99. Diagrama de bases de datos.

6.6.3. Diseño de la Interfaz

En este apartado se muestra el diseño del interfaz obtenido en el apartado de diseño del sistema. Se representa el diseño con un mockup, realizado con *Procreate*, donde se enseña la vista y el flujo básico de los botones:

UNITY:

En esta primera parte se muestra la interfaz de Unity, donde los dos primeros recuadros representan a la parte de autenticación (inicio de sesión y registro). El tercer recuadro corresponde a la interfaz una vez logueados, desde donde podemos consultar información del proyecto o adentrarnos a la Realidad Aumentada.

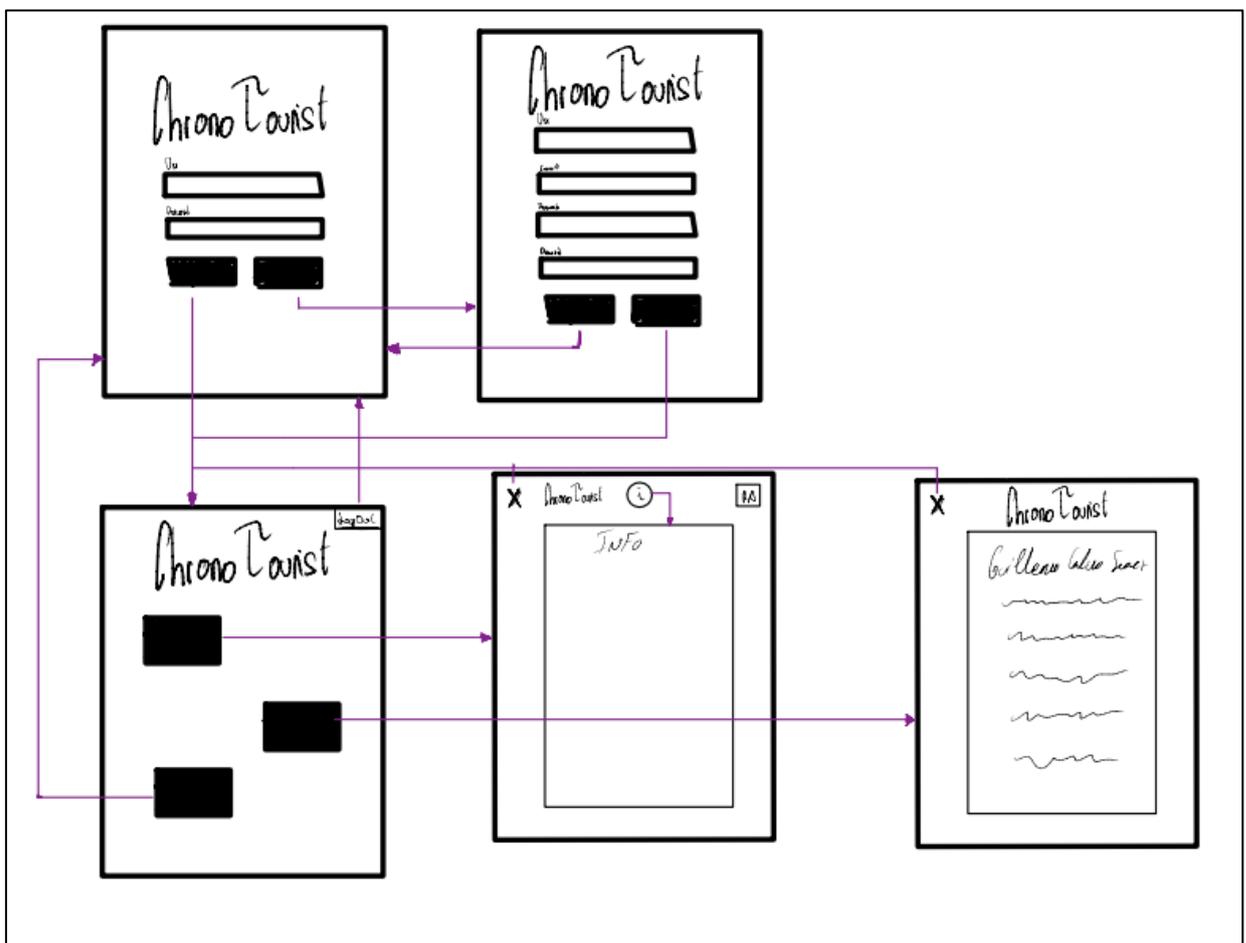
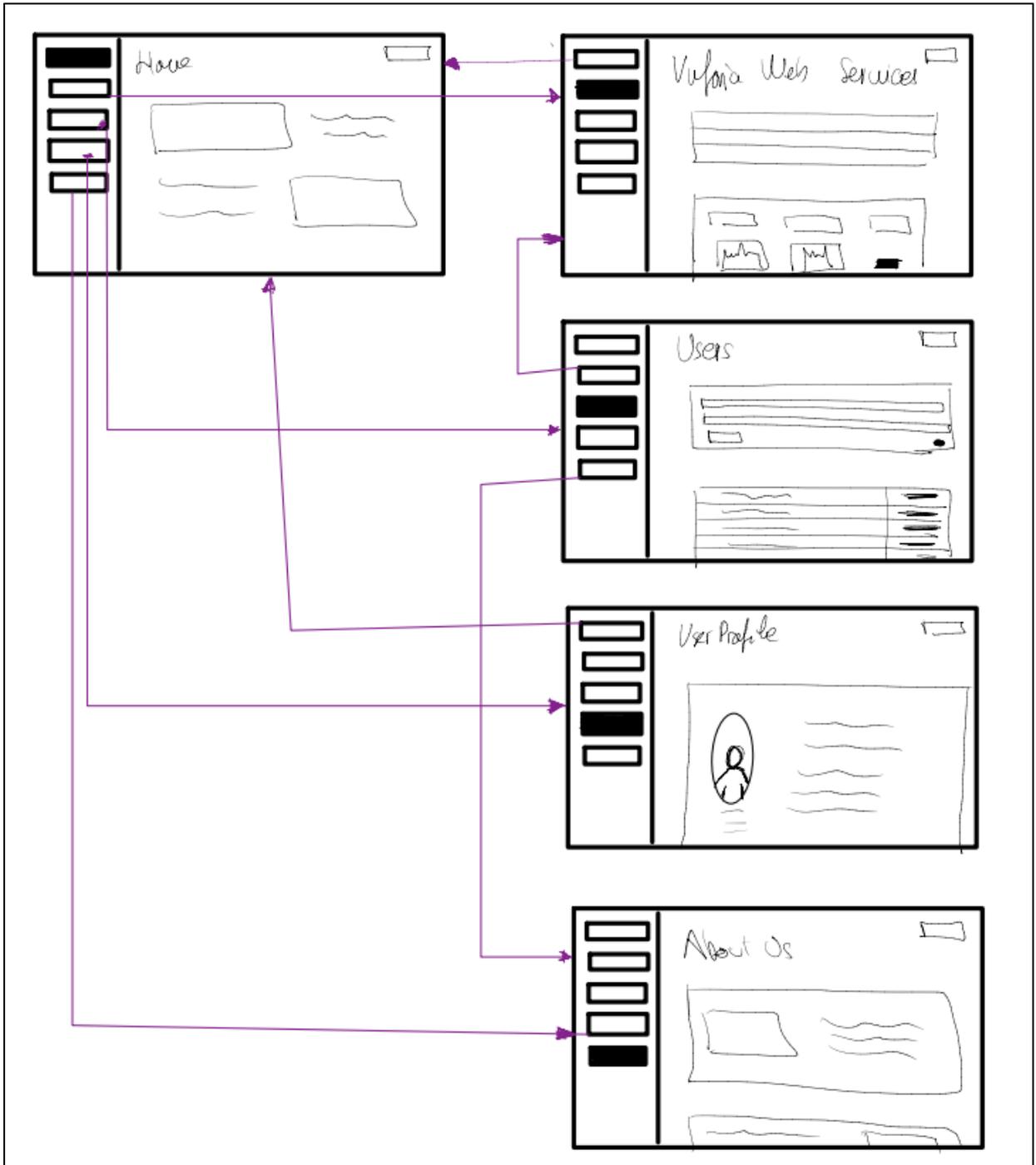


Ilustración 100mockup 1

WEB:

En esta segunda parte tenemos la interfaz web, desde donde se llevará el control de Targets y usuarios. El primer rectángulo corresponde con la Home, que nos aparecerá una vez nos autentiquemos con nuestras credenciales. Una vez logueados, nos podremos mover entre los diferentes apartados de la Web con el menú de la izquierda.



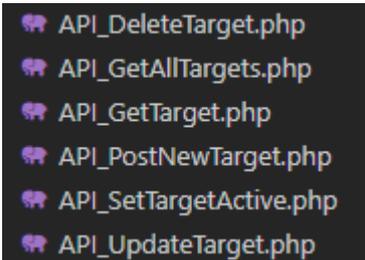
6.6.4. Diseño procedimental

A continuación, se exponen algunos de los procedimientos algorítmicos que han sido diseñados por mí, procedimientos concretos que resultan difícil encontrar en internet:

Vuforia Web Services.

Es la parte del proyecto que más trabajo ha llevado, la información en los foros es muy reducida, ya que es una tecnología que está empezando a estar en auge actualmente y, por tanto, hay pocos proyectos, aplicaciones o información con dicha tecnología.

Dentro de VWS se han diseñado diferentes algoritmos y procedimientos, entre ellos destacar los más exclusivos:



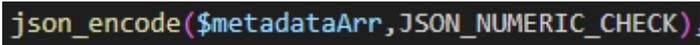
```
API_DeleteTarget.php
API_GetAllTargets.php
API_GetTarget.php
API_PostNewTarget.php
API_SetTargetActive.php
API_UpdateTarget.php
```

- Diseño de una API para cada función relacionada con el control de Targets, comunica el controlador con el propio SDK que nos proporciona Vuforia, para la correcta gestión de su Base de Datos. Estas APIS son capaces de transformar los datos al formato correcto con el que se comunica el sistema con Vuforia.

Ilustración 102. APIS

- Uso Ajax (muy importante en mi proyecto) para el envío asíncrono de información, la cual recibo y codifico con el parámetro `JSON_NUMERIC_CHECK` para poder crear un Json con algunos de sus parámetros int, consiguiendo que Vuforia los interprete correctamente. Al hacer una codificación todos los parámetros son de tipo String, y esto da problemas cuando Vuforia lo interpreta.

Las imágenes se reciben codificadas desde el controlador de Targets.



```
json_encode($metadataArr, JSON_NUMERIC_CHECK);
```

Ilustración 103. json_encode.

- Creación de una carpeta con el ID del Target en la Base de Datos del servidor, consiguiendo una Base de Datos orientada a ficheros donde se almacenarán tres archivos, imagen a reconocer, imagen a superponer y un txt con información del Target, los cuales sería muy costoso pedir a la Base de Datos de Vuforia cada vez que listamos los Targets. Estos ficheros se van a buscar cuando se solicite la lista de Targets o cuando se solicite ver un Target, para mostrar las dos imágenes y los atributos visualmente.

- `CheckIsValidMarker ()`: algoritmo que indica que todos los parámetros estén en el rango y sean válidos para la base de datos de Vuforia, comprobando tamaño, tipo de archivo, relleno de todos los campos obligatorios y su correcto formato (links, correo, num de tlf...) ... y devolviendo las instrucciones necesarias al usuario.

Estos procedimientos que acabamos de explicar se encuentran en las 6 APIS desarrolladas.

IMG DRIVE.

Vuforia no recibe la imagen a superponer codificada o como PNG o JPG, sino que necesita un URL de descarga, para ello se diseñara un algoritmo que cuando se crea o se modifica un Target, sube automáticamente la imagen a Drive y con el id de la imagen devuelto crea el URL de descarga. También es capaz de omitir este proceso si se inserta manualmente un URL de descarga de una imagen ya subida.

CACHÉ

Normalmente cuando entramos a una web, la información de las bases de datos se almacena en la caché, reduciendo así los tiempos, esto puede dar ciertos problemas a la hora de bases de datos en tiempo real, ya que, si almacenamos en el servidor, la caché del navegador tendrá información diferente. Para ello se ha diseñado un algoritmo que comprueba que el hash de los ficheros de la caché no cambia, una vez cambie, se fuerza el volcado de los ficheros del servidor de nuevo a la caché.

INTERFAZ.

La interfaz es la misma para el cliente que para el administrador, es el propio sistema el que hace que cada uno vea diferentes cosas, llevando en todo momento la sesión activa. La cual tanto si se cierra redirige al inicio de sesión, como si el usuario intenta acceder a apartados del administrador le redirige a su home.

CREACION DE USUARIOS.

Algoritmo que permite a los usuarios que solo quieren usar la APP Android, sin la necesidad de subir sus propias imágenes, no ocupar espacio innecesario en la firestore de Firebase, en el momento que quieran acceder a la interfaz web, los datos se crean automáticamente y de manera transparente en la firestore.

FUNCIONES CLOUD.

Firebase no es capaz de diferenciar entre usuarios administradores y usuarios normales. Aunque cada usuario tenga un atributo que sea Rol, y con ese atributo, los administradores puedan acceder a funcionalidades que los usuarios no, hay ciertas funciones que van a tener problemas. Las dos más importantes son:

- **Registrar usuario como administrador:** El problema viene al crear un usuario siendo administrador, no hay ningún problema a la hora de la creación, pero una vez creamos el usuario, nos vamos a autenticar con él, y por tanto la sesión activa pasara a ser la del usuario creado y dejaremos de estar en nuestra cuenta de administrador.
- **Borrar usuario:** Borrar el usuario con el que hemos iniciado sesión, no tienen ningún problema, ya que hemos autenticado con el recientemente, y la sesión la tenemos activa, el problema viene cuando somos administradores y queremos borrar un usuario, tendríamos que autenticarnos con las credenciales del usuario, y

necesitaríamos la contraseña, y no sería útil hacerlo así, ya que no se está ejecutando desde la cuenta del administrador.

La solución de estos problemas está en las funciones CLOUD. Instalamos las *tools* de Firebase con nodeJS, para ello, tenemos que cambiar algunos parámetros de la Shell para que pueda leer los scripts del sistema.

Set-ExecutionPolicy RemoteSigned -Scope CurrentUser

El objetivo de todo esto es que estas funciones problemáticas se ejecuten desde el propio servidor, y no desde la máquina del cliente.

Para poder ejecutar funciones cloud, Firebase solicita un método de pago, para que no sea necesario hemos diseñado una emulación, de manera que, en localhost, podemos emular dichas funciones sin ser necesario este pago. Esta emulación correrá en el puerto 5001.

firebase emulators:start --only functions

```
D:\xampp\htdocs\firebase\mio\functions>firebase emulators:start --only functions
i emulators: Starting emulators: functions
! functions: The following emulators are not running, calls to these services from the Functions emulator will affect p
roduction: auth, firestore, database, hosting, pubsub, storage
+ functions: Using node@16 from host.
i ui: Emulator UI logging to ui-debug.log
i functions: Watching "D:\xampp\htdocs\firebase\mio\functions\functions" for Cloud Functions...
+ functions[us-central1-borrarUsuario]: http function initialized (http://localhost:5001/user-manager-42dcc/us-central1
/borrarUsuario).
+ functions[us-central1-registrarUsuarioAdmin]: http function initialized (http://localhost:5001/user-manager-42dcc/us
-central1/registrarUsuarioAdmin).
```

📌 All emulators ready! It is now safe to connect your app.
i View Emulator UI at <http://localhost:4000>

Emulator	Host:Port	View in Emulator UI
Functions	localhost:5001	http://localhost:4000/functions

```
Emulator Hub running at localhost:4400
Other reserved ports: 4500

Issues? Report them at https://github.com/firebase/firebase-tools/issues and attach the *-debug.log files.
```

Ilustración 104. Emulador CloudFunctions.

6.7. Pruebas

Durante el desarrollo del proyecto es de suma importancia realizar pruebas incrementales para comprobar el correcto funcionamiento de la plataforma o solucionar problemas en caso de encontrarnos con ellos. También nos ayudara a encontrar errores ocultos y solucionarlos.

Estas pruebas se han realizado durante el desarrollo de cada componente y en la finalización, incluyendo también pruebas cuando unos componentes interactúen con otros.

Una vez acabado el desarrollo del sistema, se han realizado pruebas completas de todas las funcionalidades que ofrece. De todos los escenarios y desde todos los tipos de usuarios posibles, comprobando así el correcto funcionamiento de este.

A continuación, se muestran ejemplos de las pruebas de las funcionalidades más importantes del sistema

6.7.1. Tipos Usuarios y Pruebas

El actor Usuario no registrado apenas podrá acceder a las funcionalidades del sistema, es una aplicación de uso restringido a usuarios registrados.

Las únicas funcionalidades a las que puede acceder están relacionadas con la autenticación y el registro.

Las interfaces a las que puede acceder son sencillas y atractivas a primera vista, dándole intenciones de ser un usuario más del sistema.

Las demás funcionalidades solo pueden ser accedidas por un usuario registrado en el sistema. Se indicará en las diferentes pruebas las posibilidades siendo administrador o usuario.

Siempre que se hable de usuario, también podrá ser accedido por el administrador, excepto la página home.

A continuación, se muestran las principales pruebas utilizadas para comprobar el correcto funcionamiento del sistema, antes de ser lanzado.

6.7.1.1. Prueba registrar usuario nuevo e iniciar sesión.

Se ha realizado esta prueba para comprobar el correcto funcionamiento del sistema al crear un usuario e iniciar sesión.

El usuario introduce las credenciales en el formulario que aparece en la siguiente imagen:

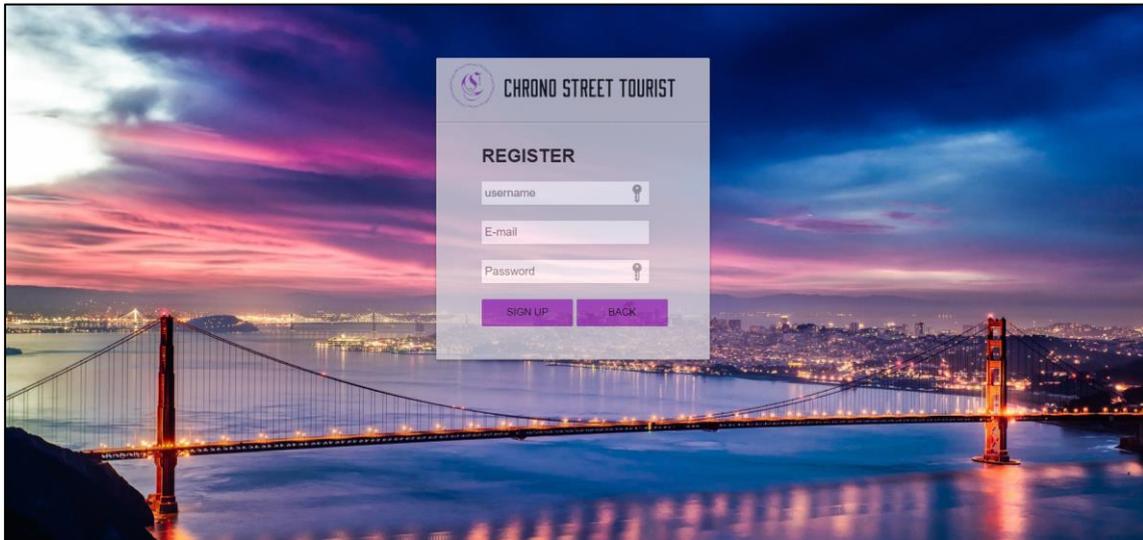


Ilustración 105. Registro

Una vez rellenados todos los campos y pulsado el botón *sign up*, el usuario será avisado y recibirá un email de confirmación en la dirección de correo introducida:



Ilustración 106. Email confirmación correo

Si el usuario intenta acceder con las credenciales correctas, pero no ha confirmado el correo, el sistema volverá a enviar un correo de confirmación y se lo indicará al usuario.

Una vez el usuario confirme la dirección de correo, introducirá las credenciales y pulsará el botón *login*, si son erróneas, el sistema se lo indicará, y si son correctas accederá al sistema:

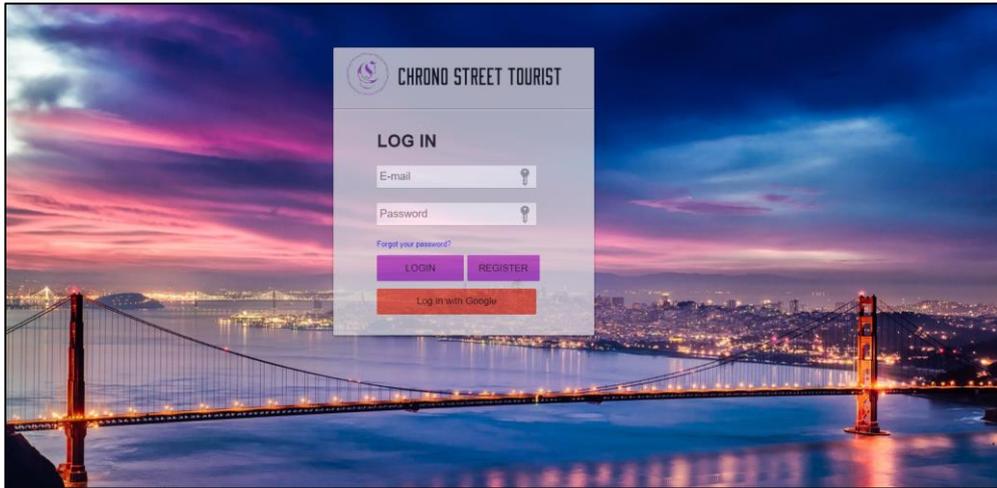


Ilustración 107. Inicio de sesión

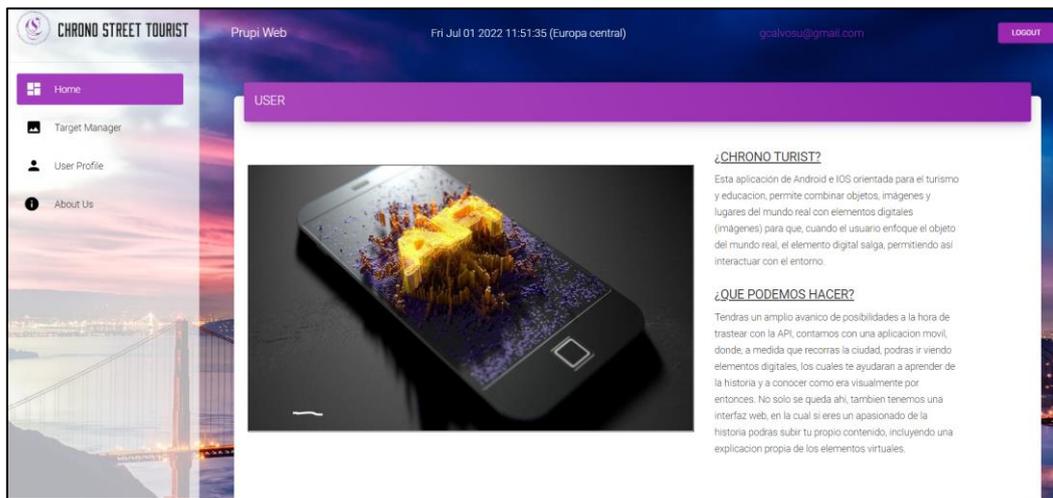


Ilustración 108. Home usuario

Si es administrador, se le mostrara diferente home:

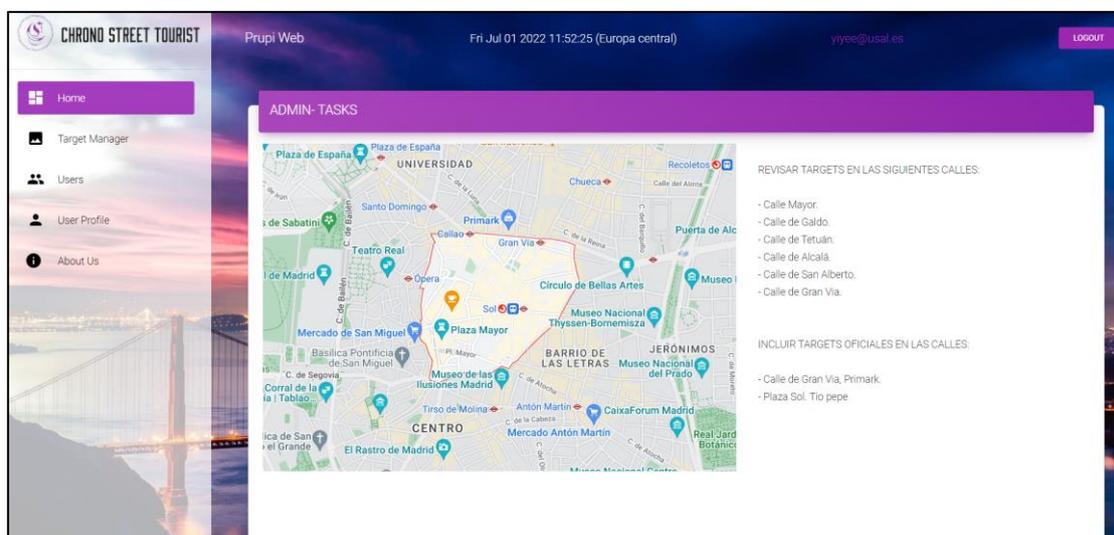
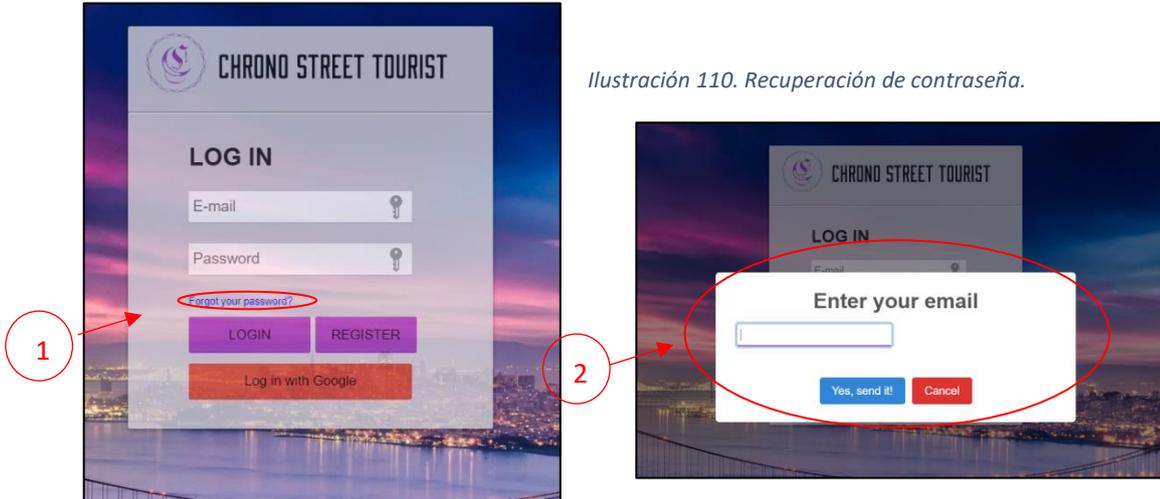


Ilustración 109. Home administrador.

6.7.1.2. Prueba recuperar contraseña.

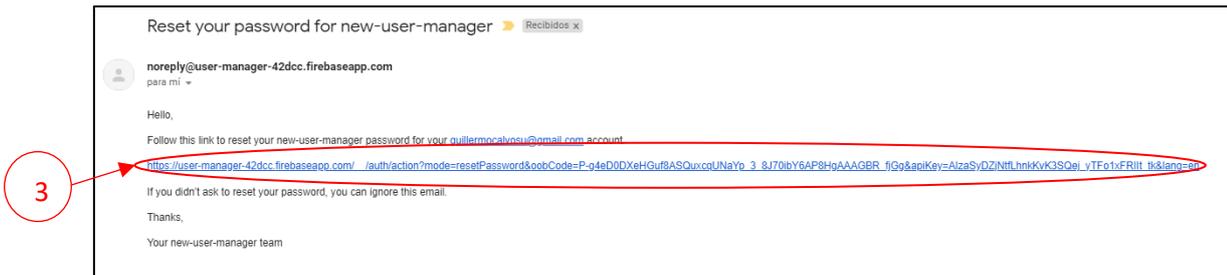
Se ha realizado esta prueba para comprobar el correcto funcionamiento del sistema al recuperar contraseña.

El usuario presiona el botón **Forgot your password?**, una vez pulsado, se le muestra una modal de solicitud de correo, la cual debe rellenar y pulsar el botón **accept**.



Una vez realizados los pasos anteriores, el usuario recibe en el correo indicado un link de restablecimiento de contraseña, al cual accede y rellena los campos necesarios.

Una vez hecho, presiona el botón **save**, y la contraseña habrá sido modificada en el sistema, y por tanto podrá acceder a este con las nuevas credenciales.



Reset your password

for **guillermocalvosu@gmail.com**

New password

SAVE

4

5

Detailed description: This is a screenshot of a web form titled "Reset your password" for the email address "guillermocalvosu@gmail.com". The form contains a text input field labeled "New password" with a password icon on the right. A blue button labeled "SAVE" is positioned below the input field. Two red circles with arrows point to the "New password" field (labeled with the number 4) and the "SAVE" button (labeled with the number 5).

Ilustración 112. vista cambio de contraseña.

Password changed

You can now sign in with your new password

Detailed description: This is a confirmation message displayed in a white box with a black border. It features the heading "Password changed" in bold black text, followed by the message "You can now sign in with your new password" in a standard black font.

Ilustración 113. Confirmación restablecimiento de contraseña.

6.7.1.3. Prueba ver lista de Targets y búsqueda con filtro.

Se ha realizado esta prueba para comprobar el correcto funcionamiento del sistema al mostrar la lista de Targets al usuario y al hacer una búsqueda con filtro en dicha lista.

Cuando el usuario accede a la interfaz de *Target Manager*, el sistema le mostrará el listado de Targets asociados con el usuario, en caso de ser administrador, el sistema le mostrará el listado de Targets completo. La interfaz difiere un poco en caso de ser administrador:

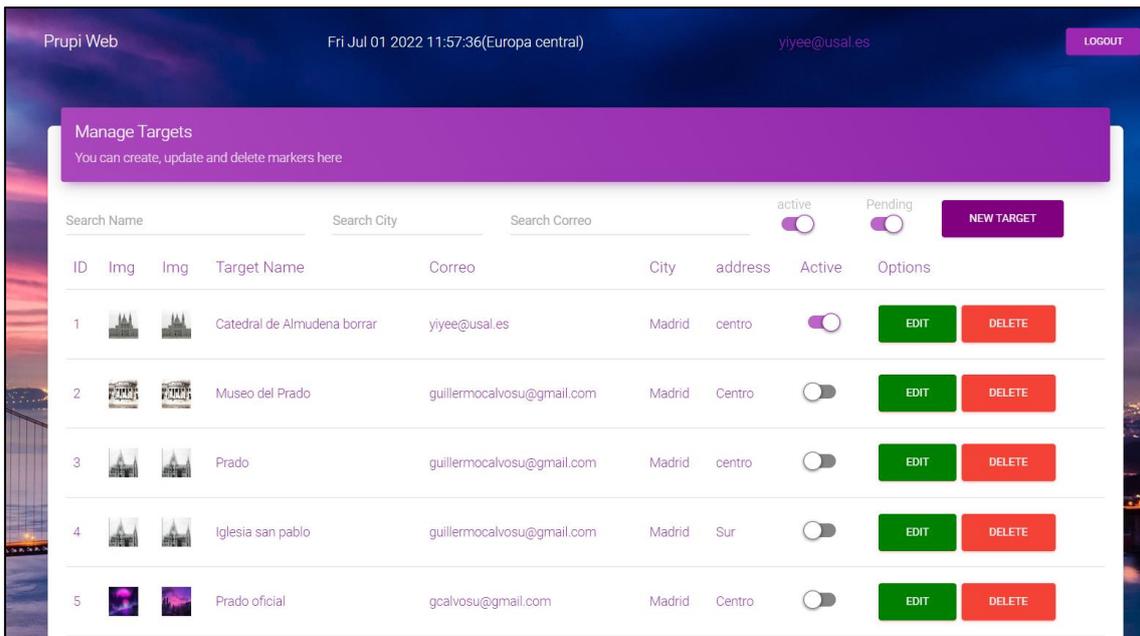


Ilustración 114. Target Manager User.

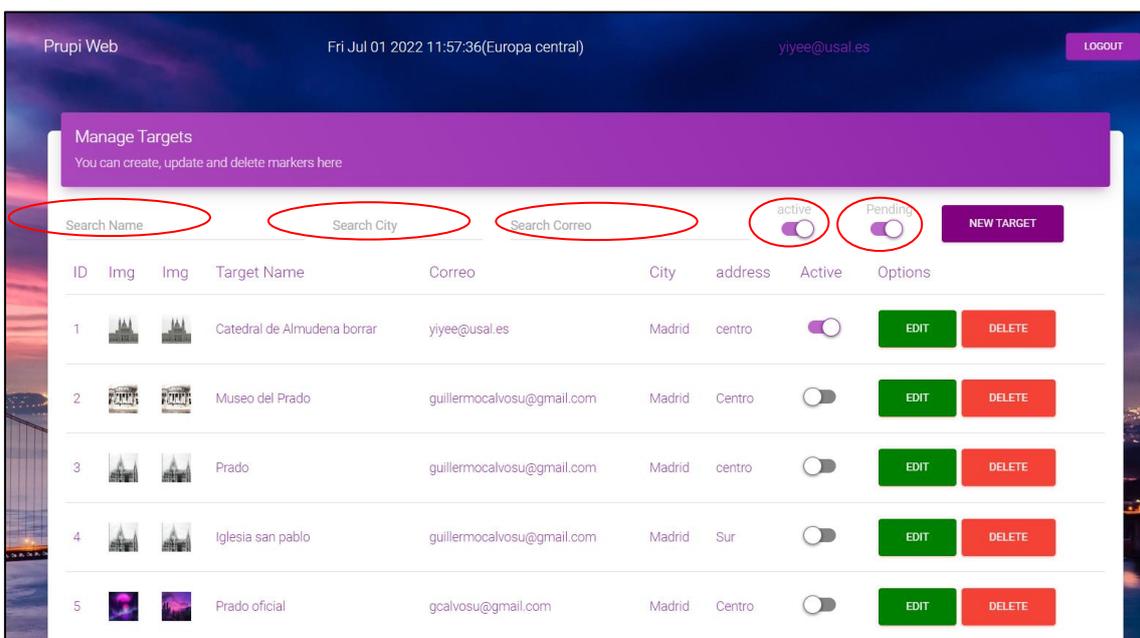


Ilustración 115. Target Manager Admin.

A continuación, el Usuario, añade el filtro, introduciendo los datos en los filtros por los cuales quiera filtrar. Los filtros son independientes, lo que permite al usuario poder filtrar por los que quiera a la vez:

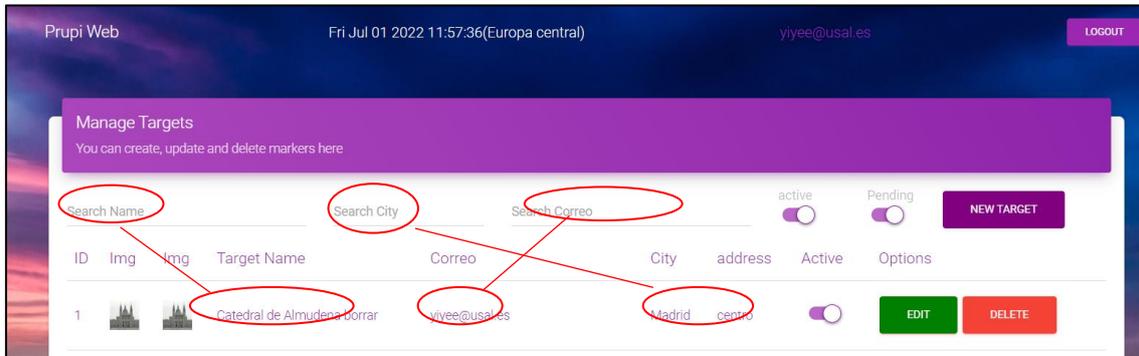


Ilustración 116. Ejemplo filtrado de Targets.

6.7.1.4. Prueba subir nuevo Target.

Se ha realizado esta prueba para comprobar el correcto funcionamiento del sistema al crear un nuevo Target.

Para crear un nuevo Target el usuario presiona el botón *New Target*, en la interfaz *Target Manager* ([Ilustración 74.](#) Y [Ilustración 75.](#)), el sistema le responde mostrándole un formulario de solicitud de datos del Target a publicar.

El usuario rellenara los campos obligatorios, seleccionara las dos imágenes, imagen a superponer e imagen a reconocer, y presiona el botón *Confirm*, indicando así al sistema la confirmación de nuevo Target.

El sistema subirá los datos a Drive y Vuforia y le responderá con el resultado de la operación, o en caso de error, instrucciones al usuario del error y que debe hacer para la correcta publicación del Target. ([Ilustración 79. Target subido correctamente](#)).

En caso de ser usuario, el sistema sube el Target en estado pendiente esperando la revisión y activación de un administrador.

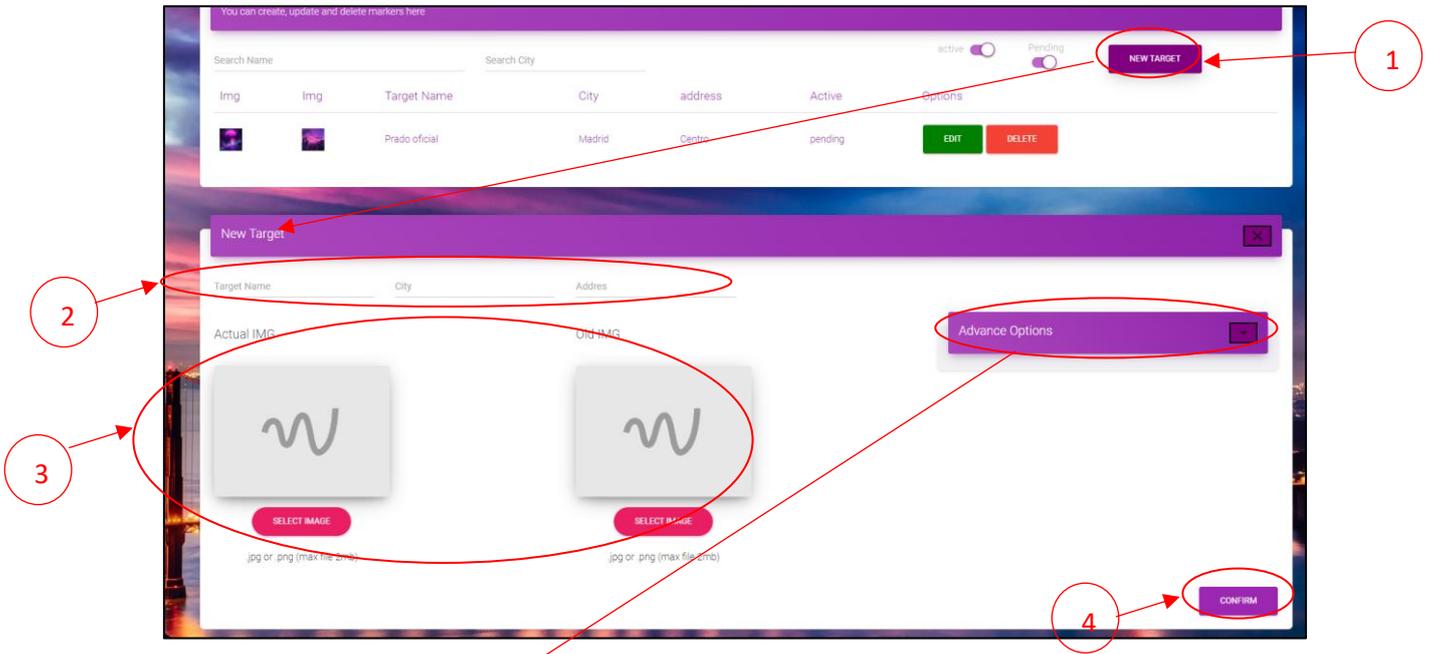


Ilustración 118. Formulario New Target.

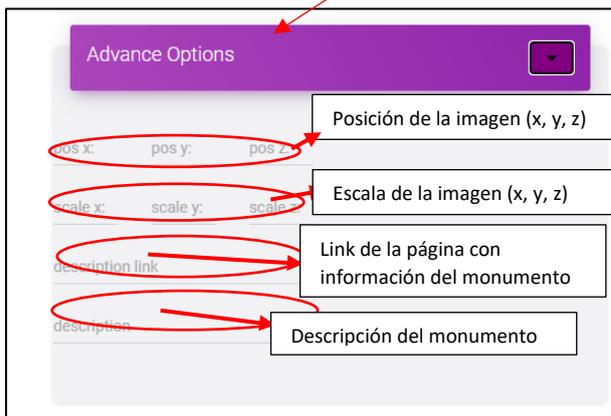


Ilustración 117. Opciones avanzadas.

En caso de ser un usuario más experimentado o un administrador, puede introducir parámetros adicionales de mayor nivel para mejorar la calidad del Target.

Si estos parámetros no se rellenan, el sistema lo hará con datos por defecto.

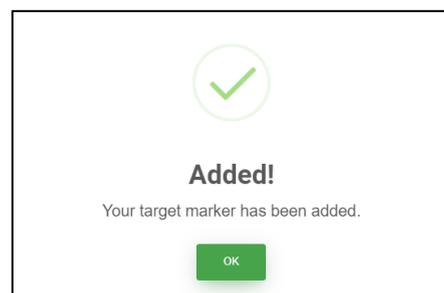


Ilustración 119. Target subido correctamente.



Ilustración 120. Comprobación Target subido.

6.7.1.5. Prueba ver Target y modificarlo.

Se ha realizado esta prueba para comprobar el correcto funcionamiento del sistema al ver la información de un Target, y consiguente, modificarlo.

Para modificar un Target el usuario presiona el botón *EDIT* que se muestra en la siguiente figura:

Search Name	Search Correo	Search City	Active	Options		
Img	Img	Target Name	City	address	Active	Options
		asdf	asdf	asdf	pending	<input type="button" value="EDIT"/> <input type="button" value="DELETE"/>
		Museo del Prado	Madrid	Centro	pending	<input type="button" value="EDIT"/> <input type="button" value="DELETE"/>
		Catedral Almudena	Madrid	centro	pending	<input type="button" value="EDIT"/> <input type="button" value="DELETE"/>

Ilustración 121. Editar Target.

El sistema le mostrara la información del Target, mostrándole las dos imágenes en grande, y los atributos que tiene esta:

Update Target

Target Name: Catedral Almudena City: Madrid Address: centro

Actual IMG: (jpg or png (max file 2mb))

Old IMG: (jpg or png (max file 2mb))

Advance Options

pos x:	pos y:	pos z:
0,1	0,1	0,1
scale x:	scale y:	scale z:
0,1	0,1	0,1

description link: _____

description: _____

Ilustración 122. Ver Target.

El usuario rellenara los campos o imágenes que quiera modificar y presionara el botón *confirm* para confirmar los cambios.

Update Target

Target Name: Prado City: Madrid Address: centro

Actual IMG:

Old IMG:

Advance Options

pos x:	pos y:	pos z:
0,1	0,1	0,1
scale x:	scale y:	scale z:
0,1	0,1	0,1

description link: hola.com

description: hola

Ilustración 123. Modificar Target.

El sistema modificará los datos y le responderá con el resultado de la operación, o en caso de error, instrucciones al usuario del error y que debe hacer para la correcta modificación del Target:

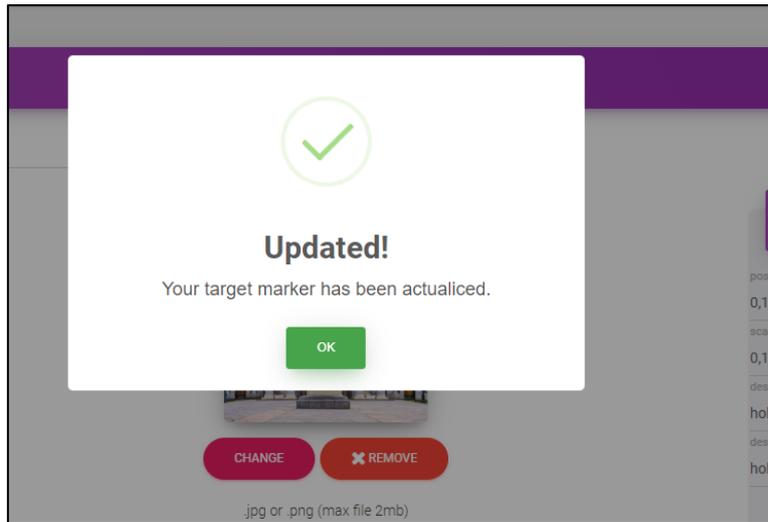


Ilustración 124. Resultado modificación Target.

El sistema mostrara en la lista el Target modificado:



Ilustración 125. Target modificado.

6.7.1.6. Prueba crear Usuario/Administrador.

Se ha realizado esta prueba para comprobar el correcto funcionamiento del sistema cuando un administrador crea un usuario/administrador.

Para crear un usuario o administrador, el administrador accede a la vista *Users*:

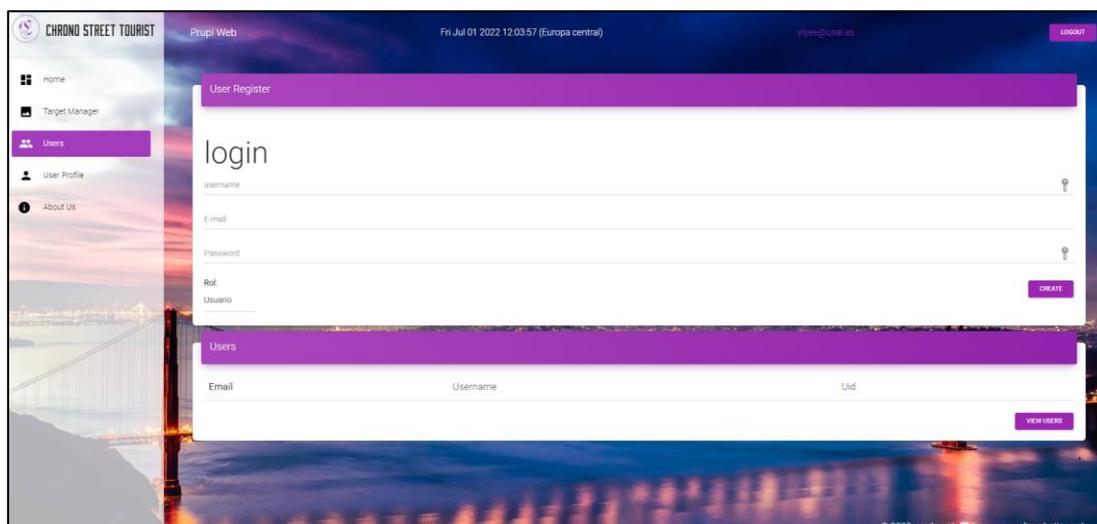


Ilustración 126. Vista Users.

El administrador rellena los campos del formulario *UserRegister*, selecciona el Rol y presiona el botón *Create*, el sistema responde con el resultado de la operación:

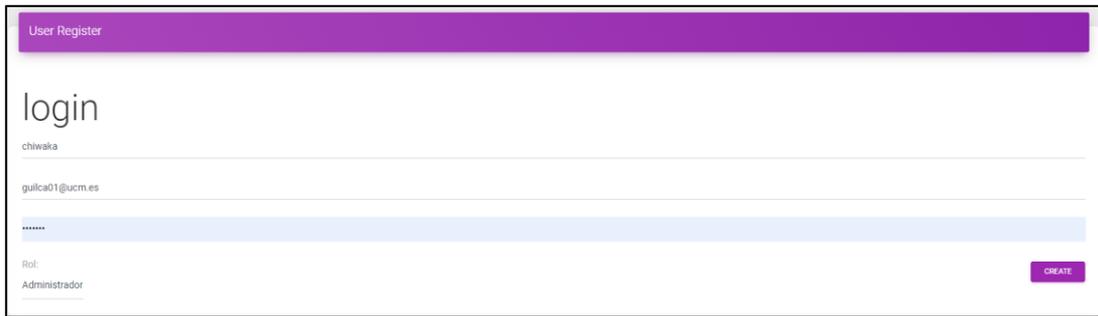


Ilustración 127. Create user

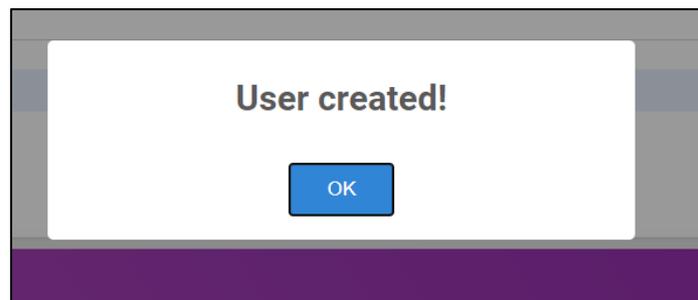
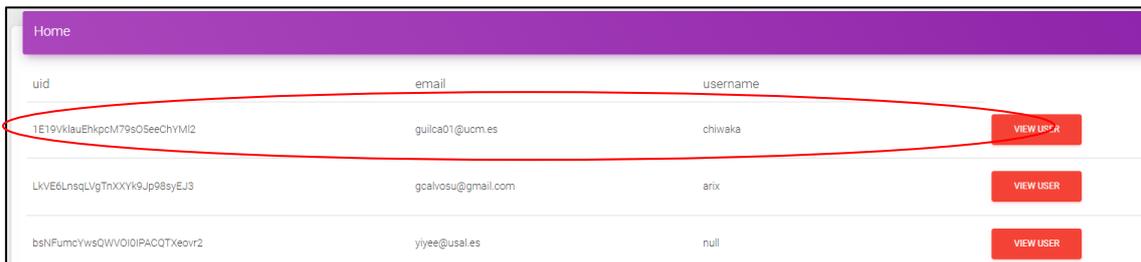


Ilustración 128. Respuesta creación usuario.



uid	email	username	
1E19VklauEhkpcM79s05eeChYMI2	guilca01@ucm.es	chiwaka	VIEW USER
LkV66LnsqLVgTnXXYk9Jp98syEJ3	gcalvosu@gmail.com	arix	VIEW USER
bstNFumcYwsQWVOi0IPACQTXeovr2	yiye@usal.es	null	VIEW USER

Ilustración 129. lista de usuarios con el nuevo usuario.

El servidor deberá estar corriendo la emulación en el puerto 5001 para este proceso ya que es una función cloud.

6.7.1.7. Prueba ver lista de usuarios e información de estos.

Se ha realizado esta prueba para comprobar el correcto funcionamiento del sistema cuando el administrador vista la lista de usuarios o la información de cualquiera de estos.

Para ver la lista de usuarios el administrador tiene que desplazarse a la vista *USERS* y clicar en el botón *view users*:

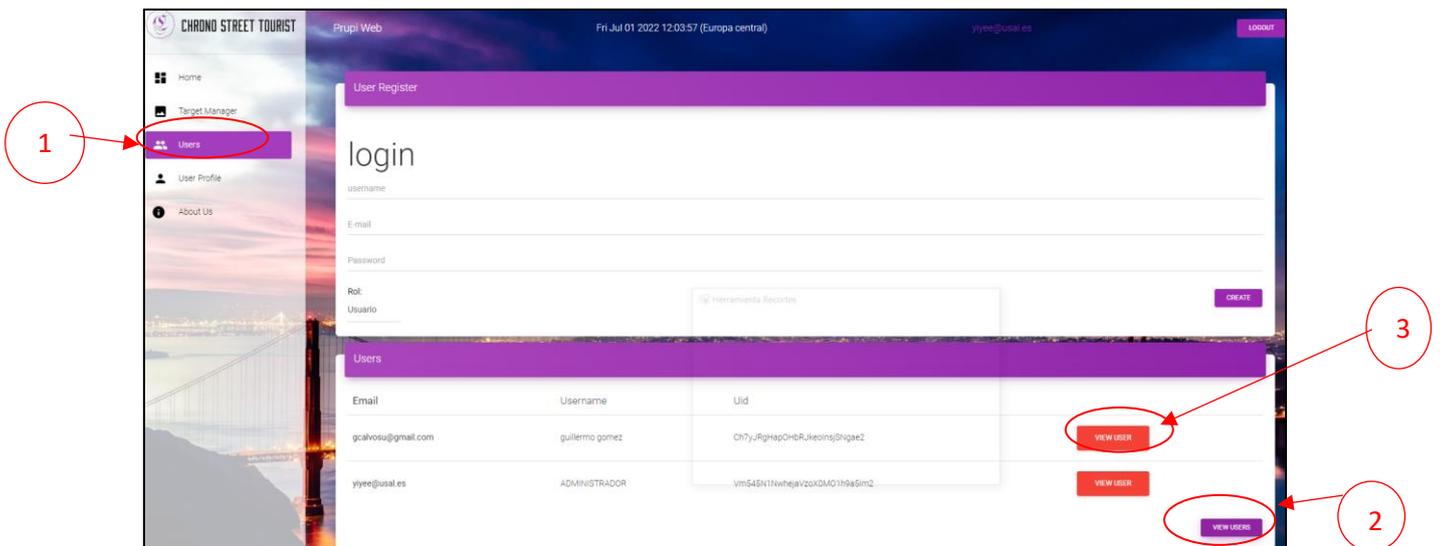


Ilustración 130. Lista usuarios

Una vez mostrada la lista, para ver la información del usuario, el administrador presiona el botón *View User*, correspondiente al usuario del cual quiere informarse (acción 3 en la imagen anterior):

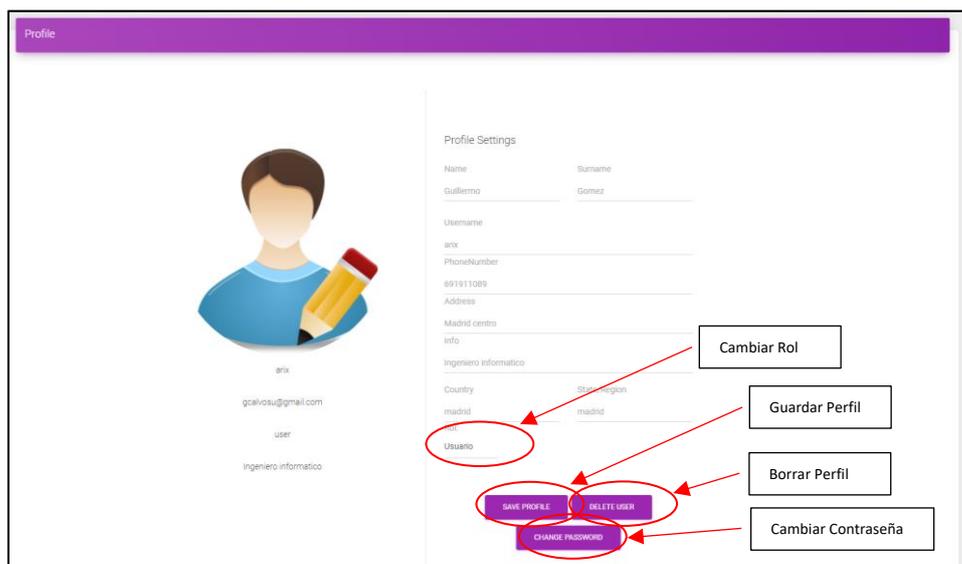


Ilustración 131. información de otro usuario (admin).

A partir de esta interfaz, el administrador estará en la información del usuario, desde la cual puede Cambiar el Rol, modificar datos del usuario, borrar el perfil del usuario o solicitar un cambio de contraseña para el usuario.

Las pruebas de dichos casos se realizarán a continuación y funcionan tanto para administradores sobre los datos de un usuario, como para los usuarios sobre su propio perfil, en este último caso, exceptuando Cambio de Rol.

6.7.1.8. Prueba modificar Usuario.

Se ha realizado esta prueba para comprobar el correcto funcionamiento del sistema cuando se modifica la información del perfil de usuario.

Primero se debe acceder a la interfaz del perfil del usuario, en caso de ser administrador, se detalla en

[Prueba ver lista de usuarios e información de](#) estos. [_como llegar a la información de un usuario.](#)

En caso de ser usuario, o administrador queriendo modificar su propia información, debe acceder a la vista *UserProfile*:

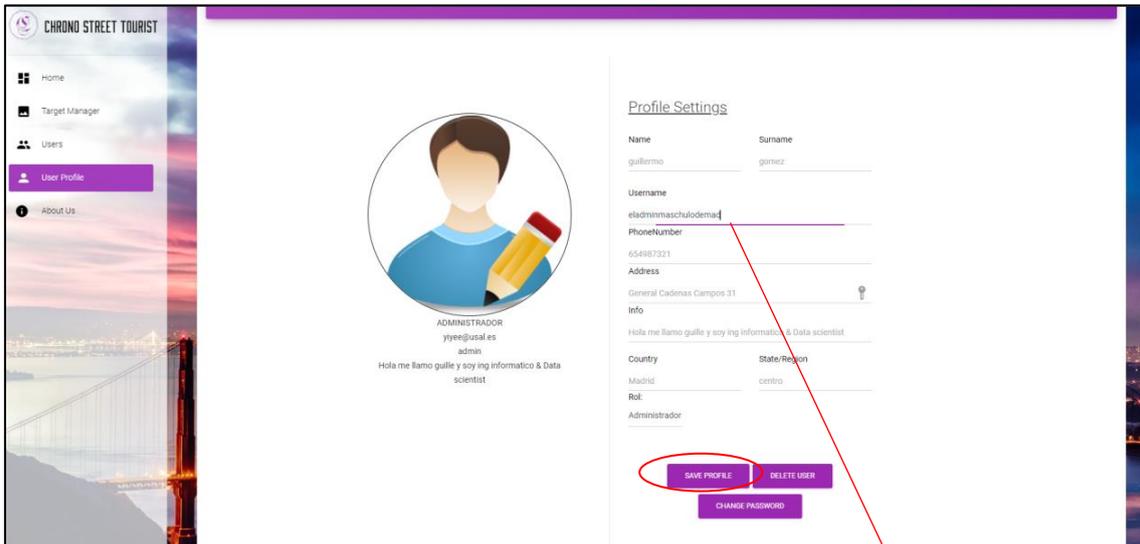


Ilustración 132. User profile

El usuario o administrador rellena únicamente los campos que quiere modificar, y presiona el botón *Save Profile*.

El sistema responde con el resultado de la operación.

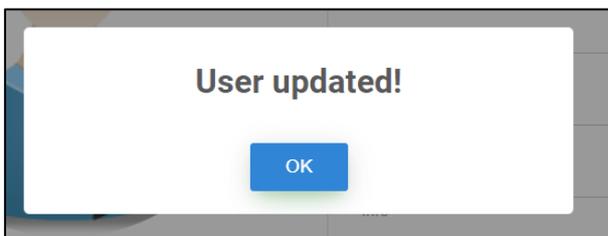


Ilustración 134. Userupdated.

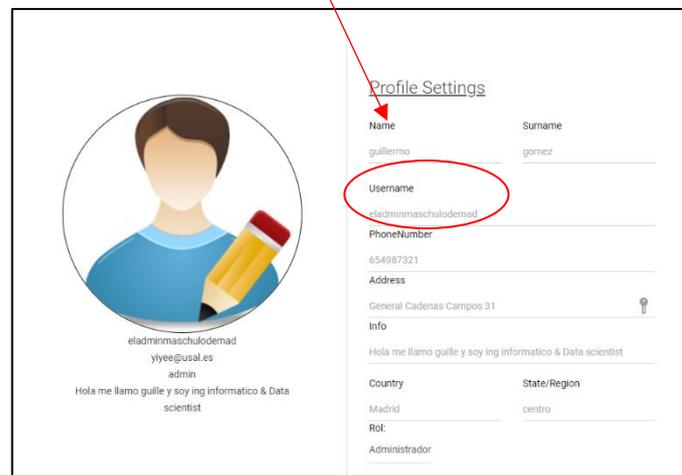


Ilustración 133. Usuario modificado.

6.7.1.9. Prueba cambiar Rol.

Se ha realizado esta prueba para comprobar el correcto funcionamiento del sistema cuando el administrador cambia el Rol de un usuario.

Partimos de la vista UserProfile que vemos en [Ilustración 91. información de otro usuario \(admin\)](#). El administrador selecciona el Rol para el usuario y presiona el boton *Save Profile*:

The screenshot shows a user profile form for a user named 'chiwaka'. The form includes fields for Name, Surname, Username, PhoneNumber, Address, Info, Country, and State/Region. The 'Role' dropdown menu is set to 'admin' and is circled with a red '1'. The 'SAVE PROFILE' button is also circled with a red '2'. Other buttons include 'DELETE USER' and 'CHANGE PASSWORD'.

Ilustración 135. Cambio Rol

The screenshot shows the same user profile form, but the 'Role' dropdown menu is now set to 'user' and is circled with a red oval. A red arrow points from the 'admin' role in the previous screenshot to this 'user' role. The 'SAVE PROFILE' button is no longer highlighted.

Ilustración 136. Rol cambiado

6.7.1.10. Prueba borrar usuario.

Se ha realizado esta prueba para comprobar el correcto funcionamiento del sistema cuando se borra un usuario.

Como hemos comentado anteriormente, puede realizarlo el administrador desde la información de un usuario, o el propio usuario desde su perfil.

Partimos de la vista UserProfile que vemos en [Ilustración 91. información de otro usuario \(admin\)](#). El administrador presiona el botón *Delete User*, y confirma la acción:

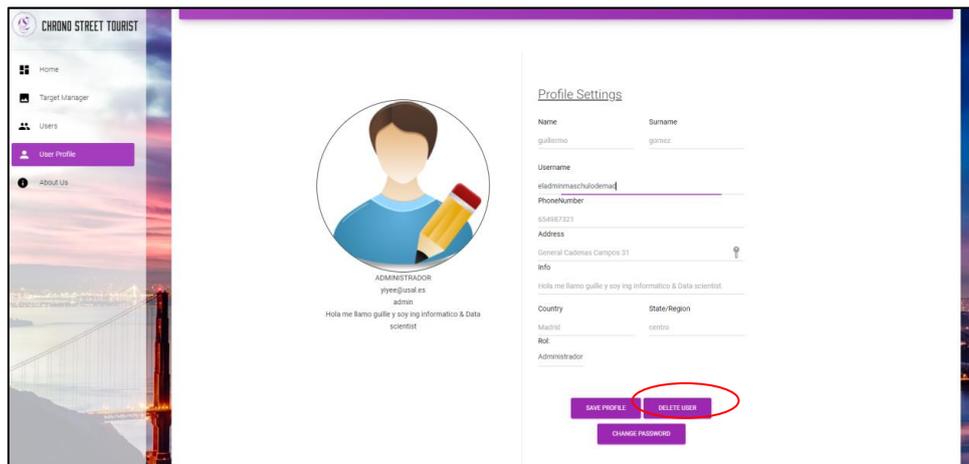


Ilustración 137. Info usuario a borrar

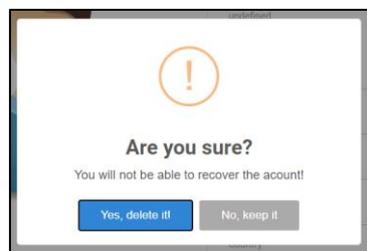


Ilustración 138. Confirmación borrar usuario

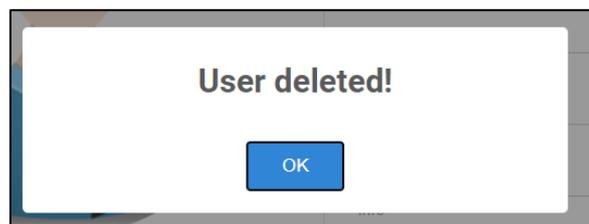


Ilustración 139. Respuesta borrar usuario

El sistema por tanto borra el usuario de la lista de usuarios:

Home			
uid	email	username	
LkVE6LnqLvgTxxkXk9lp98sjEJ3	gcalvosu@gmail.com	anx	VIEW USER
bskFumcvvs2WVVOi0PAC0T7xevr2	yiye@gmail.es	eladmmeschulodemad	VIEW USER
m2SRmDT94DvFn8D73PvMhP26hm2	gcalvosu@gmail.com	yiye@gmail.es	VIEW USER
mYwVFL8Jq1PuMrd6qdG56H483FS2	guillemocalvosu@gmail.com	Guillermo.gomez.gomez	VIEW USER
pV0Z2c8OUJf9zbuRyD8t5sRLs2	guillemocalvosu@gmail.com	guillemocalvosu@gmail.com	VIEW USER
			VIEW USERS

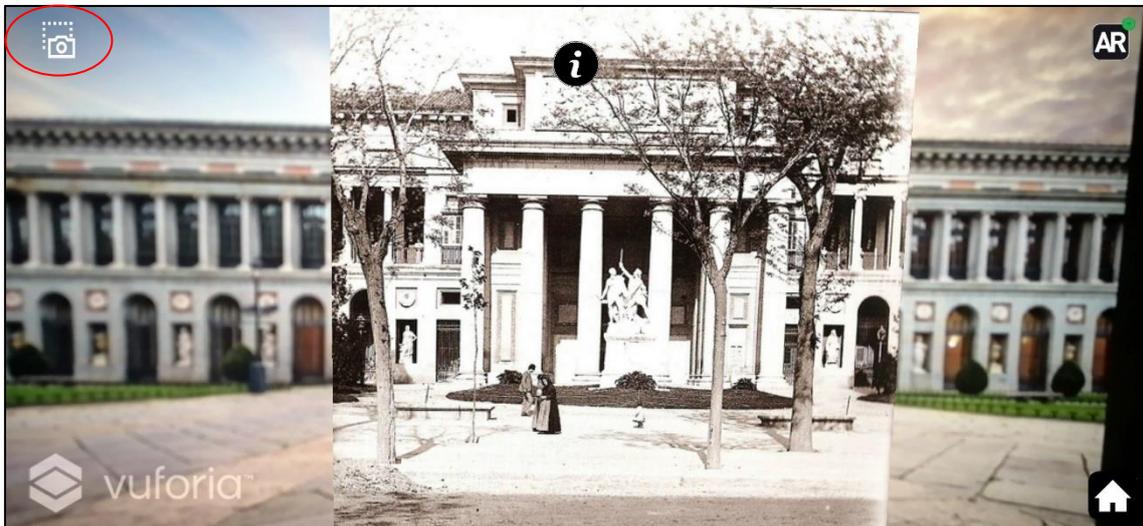
Ilustración 140. Lista sin el usuario borrado.

El servidor deberá estar corriendo la emulación en el puerto 5001 para este proceso ya que es una función cloud.

6.7.1.11. Prueba Sacar Captura de Pantalla.

Se ha realizado esta prueba para comprobar el correcto funcionamiento del sistema cuando se quiere sacar una captura de pantalla.

Partiremos del estado que aparece en la [Ilustración 108. Realidad Aumentada on](#) . El usuario presiona el icono de captura de pantalla, marcado en la siguiente imagen:



El sistema realiza la captura de pantalla y la guarda en el almacenamiento del móvil:

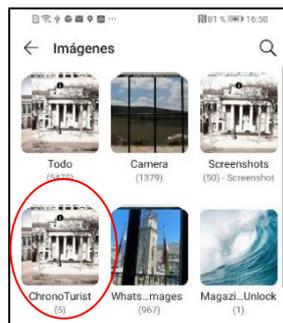


Ilustración 141.almacenamiento del móvil

6.7.1.12. Prueba completa subir Target y visualizarlo en la APP.

Se ha realizado esta prueba para comprobar el correcto funcionamiento del sistema cuando se quiere ver en la aplicación un nuevo monumento. Incluye subir nuevo monumento y visualizar monumento.

Para ver como subir un monumento más detallado dirijase a [Prueba subir nuevo Target.](#)

Primero el usuario sube el Target que quiere visualizar:

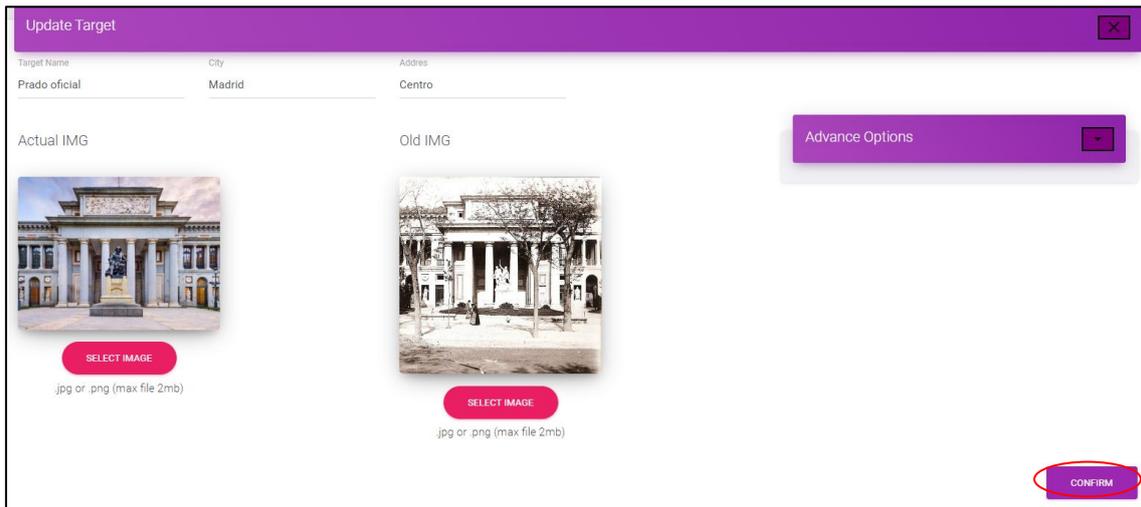


Ilustración 142. Prueba visualizar monumento nuevo 1.

Una vez subido el usuario espera a que el administrador le aprueba el Target y pase al estado active:



Ilustración 143 Target Pendiente.

El administrador aprueba el Target y lo activa:



Ilustración 144. Activar Target.

El usuario ya tiene el Target activo. Se dirige a la aplicación móvil, inicia sesión, accede a la vista de Realidad Aumentada, y apunta al monumento:



Ilustración 145. Target Activo

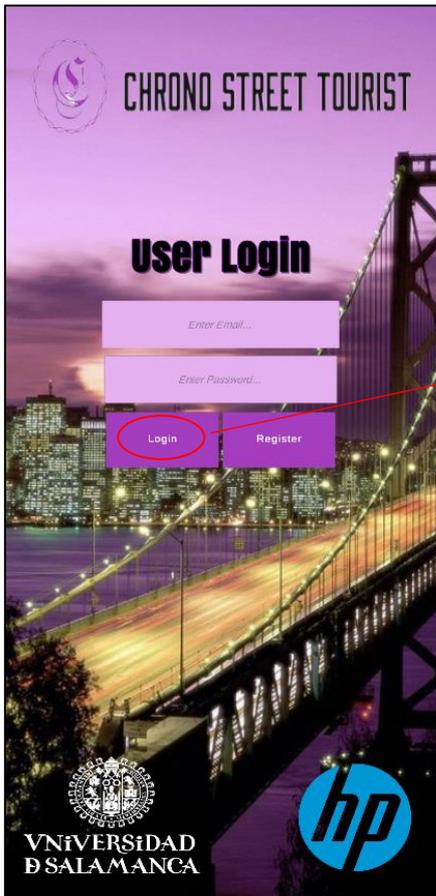


Ilustración 147. Vista opciones unity



Ilustración 146. inicio sesión

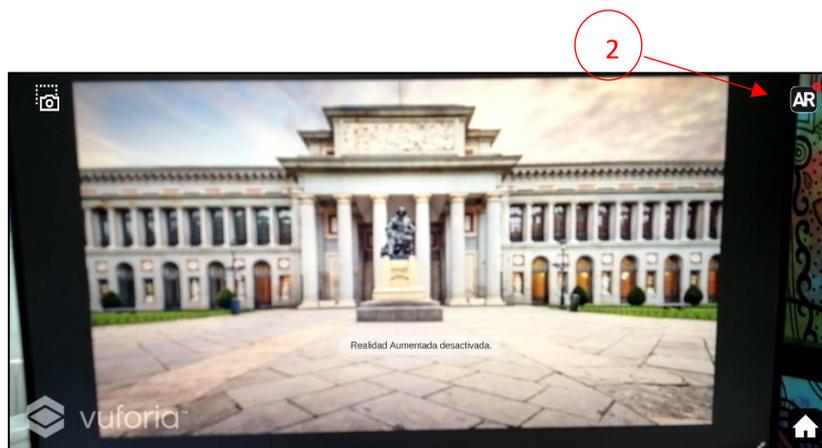


Ilustración 148. Realidad aumentada off



Ilustración 149. Realidad Aumentada on.

Para ver la información del monumento, el usuario presiona el botón de información que aparece en la parte superior del dispositivo móvil:

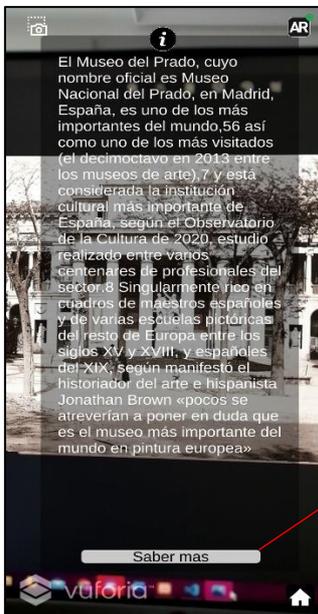


Ilustración 151. información monumento



Ilustración 150. Wiki del monumento

6.7.1.13. Pruebas de seguridad.

Se ha realizado esta prueba para comprobar el correcto funcionamiento del sistema cuando se quiere realizar una acción ilícita dentro de este.

Se han realizado varias pruebas sobre seguridad dentro del sistema, a continuación, se mostrarán las 2 más importantes.

Pérdida de sesión activa.

Si se pierde la sesión activa, el sistema redirigirá a la vista de autenticación, regulando así el acceso y la seguridad de los usuarios:

Borramos los datos de navegación (para poder comprobarlo) y el sistema redirige de la vista de *Target Manager* en la que estábamos a la vista de autenticación:

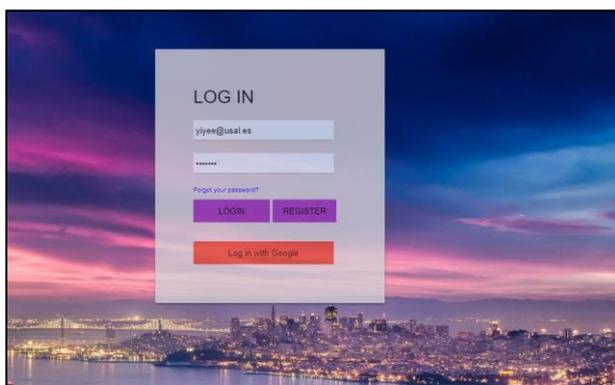


Ilustración 152. seguridad sesión activa

Si no hay sesión activa no se puede acceder a ninguna parte del sistema.

Usuarios accediendo a partes del administrador.

Cuando un usuario intenta acceder a un apartado que es exclusivo del administrador, el sistema le redirige al home de usuario.

Vamos a explicar un apartado que tiene más riesgo.

Cuando el administrador accede al perfil de un usuario está accediendo a una página a la que el propio usuario sí puede acceder. Si el usuario normal quiere jaquear el sistema e intenta acceder al `UserProfile` de otro usuario, con un link parecido al siguiente:

<http://localhost/firebase/mio/UserProfile.html?m2SRmQT94DQvFnaD73PYMnP26Nm2>

El propio sistema lo detectará y le dará un único aviso:

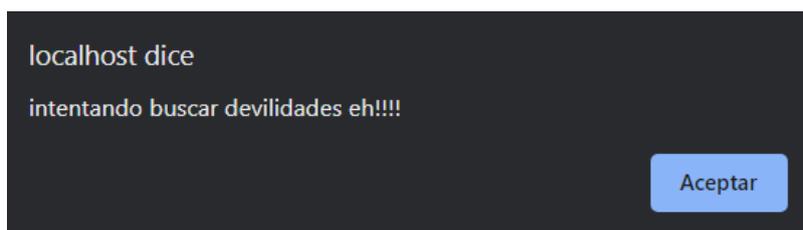


Ilustración 153. Aviso actividad ilícita.

7. Sistema desarrollado.

En este apartado veremos la funcionalidad del sistema como sistema final, complementando el anterior apartado donde hemos abarcado gran parte de esta funcionalidad, por tanto, se contará de manera resumida.

Para ver la información más detallada se puede consultar el anexo V - Manual de Usuario.

7.1. Aplicación Android.

En este apartado se recogerá todo el funcionamiento de la aplicación móvil.

La aplicación se podrá descargar desde el repositorio donde se encuentra alojado el proyecto completo:

[GITLAB](#)

La aplicación está diseñada para el uso en diferentes tipos de dispositivos, se ajustará automáticamente al dispositivo.

La aplicación Android es simplemente para visualizar la Realidad Aumentada, por tanto, no diferenciará entre usuarios y administradores, ni tampoco contará con una interfaz de manejo de Targets, ni de manejo de usuarios.

Queremos la mayor simplicidad y minimalismo posible para que sea fácil el uso de la aplicación por parte del usuario, y, por tanto, atraer miradas.

1.1. Escena inicio de sesión / registro.

En este subapartado se recoge la parte en la que el usuario puede registrarse o iniciar sesión, cada uno con su respectiva interfaz, donde podremos iniciar sesión, si tenemos una cuenta ya registrada en el sistema, o, por ende, podremos registrarnos.

Para registrarnos, debemos pulsar el botón *Register*, el cual nos lleva a [la Ilustración 152. Vista Registro](#), introducimos el username, email y contraseña, le damos al botón de *register*, y se creará automáticamente un usuario en la base de datos de la aplicación, en caso de que el correo no esté en uso.

Si ya estamos registrados, tan solo tendríamos que introducir las credenciales, y el sistema nos dirigirá a la siguiente escena: *Escena de Realidad Aumentada*, [Vista Panel Principal](#).



Ilustración 155. Vista inicio Sesión.

Ilustración 154. Vista Registro.

1.2. Escena Realidad Aumentada

En este subapartado vamos a ver la parte del sistema de Realidad aumentada, correspondiente con las pruebas [Prueba Sacar Captura de Pantalla.](#) Y [Prueba completa subir Target y visualizarlo en la APP. .](#)

Para empezar, nos encontramos con la vista principal:

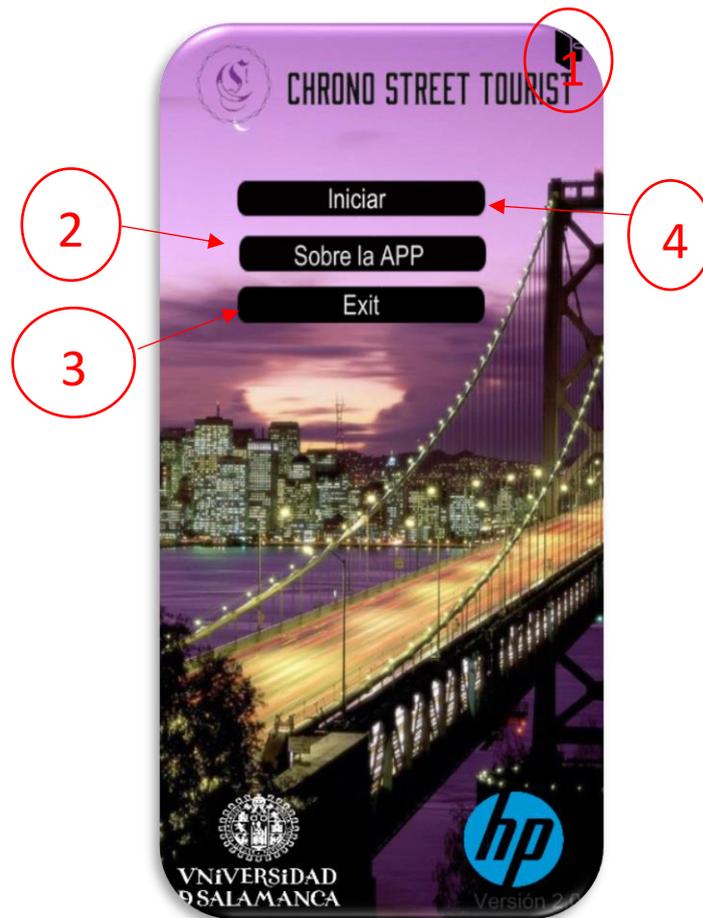


Ilustración 156. Vista Panel Principal

Desde aquí podemos hacer varias acciones, comenzaremos por las más simples:

1. Botón de Cerrar Sesión, pulsándolo nos redirigirá a la [Vista inicio Sesión](#).
2. Botón de Información, el cual nos mostrará información acerca de la aplicación, quien es su creador, instituciones colaboradoras, objetivo de la aplicación ...: [Vista Inf sobre la app](#).

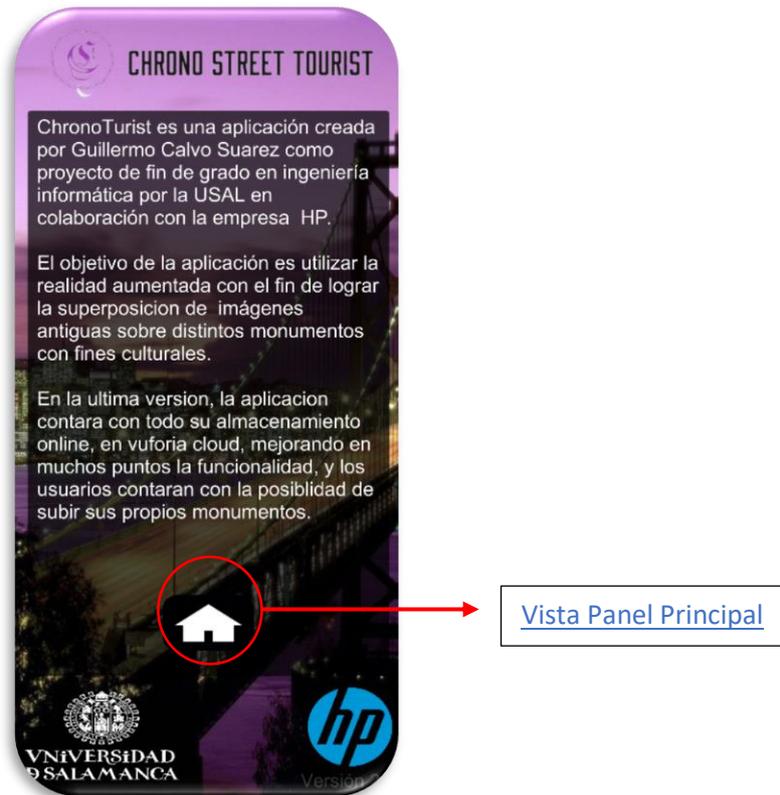


Ilustración 157. Vista Inf sobre la app

3. **Botón *exit*:** Nos cierra la aplicación y volvemos a la pantalla de inicio de nuestro teléfono.
4. **Botón de Realidad Aumentada:** Nos lleva a la vista de Realidad Aumentada, donde podremos ver la verdadera funcionalidad de la aplicación, viendo monumentos de hace siglos sobre el monumento actual, en la vida real, gracias a la cámara. También se da la posibilidad de funcionalidades adicionales que explicaremos más adelante:



Ilustración 158. Vista Realidad Aumentada Desactivada

Activa/Desactiva la Realidad Aumentada, el punto rojo nos indica que esta desactivada.

Si activamos la Realidad Aumentada, y apuntamos al mismo monumento, podremos visualizar el monumento de hace siglos superpuesto:

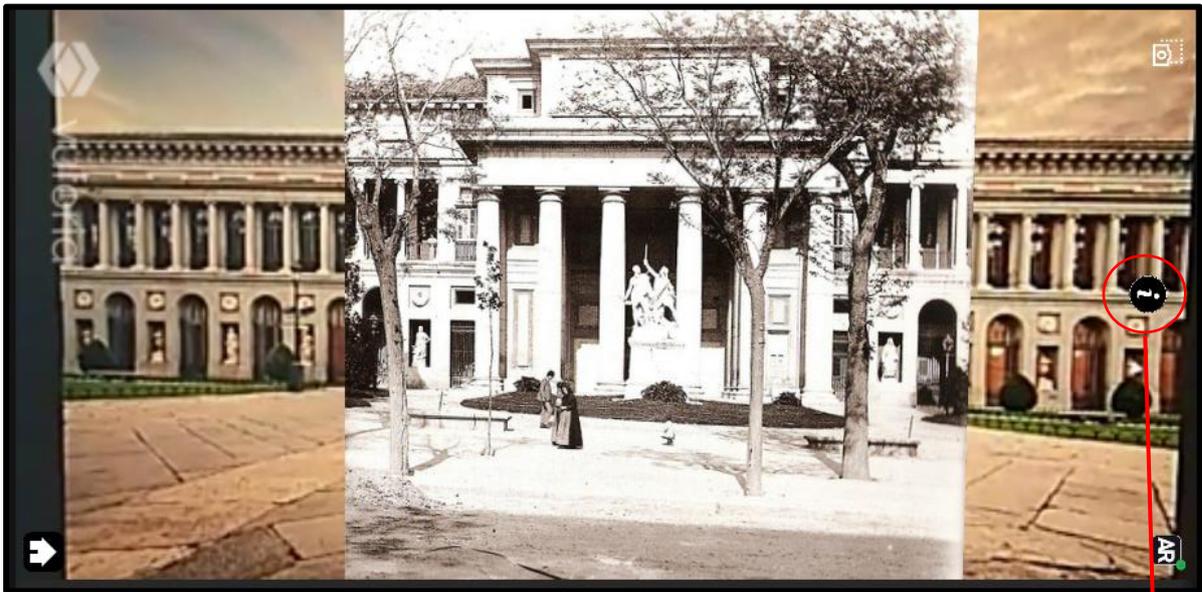


Ilustración 159. Vista Realidad Aumentada Activada

Botón de información, que nos mostrara en pantalla información acerca del monumento. [Vista información del monumento](#)

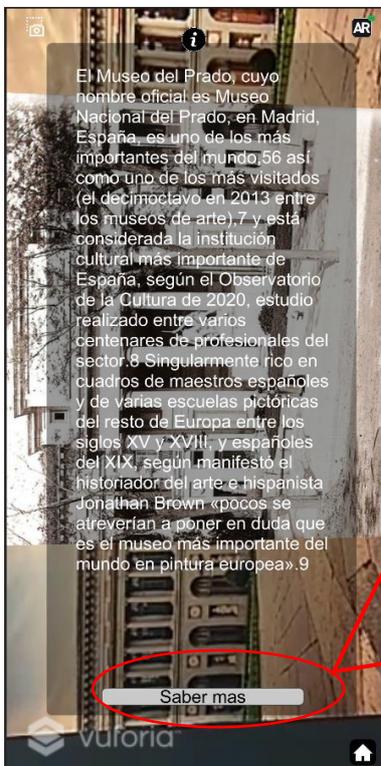


Ilustración 160. Vista información del monumento

En caso de querer saber más del monumento, tenemos la posibilidad de presionar el botón de saber más, el cual nos redirigirá a una página web, donde podemos encontrar información del monumento mucho más detallada.



8. Aplicación Web.

8.1. Inicio de sesión/Registro.

Como hemos dicho, el objetivo del proyecto es que sea lo más simple y minimalista posible, para dar resultado a una aplicación intuitiva y fácil de usar para los usuarios. Para ello todo el manejo de Targets y de usuarios se hace desde una interfaz web, lo cual va a ser mucho mejor:

1. Para el usuario que solo quiera usar la aplicación Android como una guía turística sin contribuir en la progresión de esta, simplemente visualizando la Realidad Aumentada creada por otros usuarios y administradores.
2. Para un usuario que aparte de usar la aplicación Android, quiere contribuir en la progresión del proyecto, publicando Targets y añadiéndole calidad a la aplicación.
3. Para el administrador, al estar la parte de manejo de Targets y usuarios separada, le será mucho más simple llevar un correcto control, añadiendo que interactúa desde una interfaz web y por tanto tendrá mejor resultado y le resultará mucho más cómodo manejándolo desde un ordenador.

Una vez accedemos a la interfaz web, lo primero que nos encontramos es el formulario de inicio de sesión, donde podremos acceder al sistema, introduciendo nuestras credenciales o a través de una cuenta de Google, o acceder a la interfaz de registro, como se ve en [Prueba registrar usuario nuevo e iniciar sesión.](#)

También podemos recuperar la contraseña en caso de haberla perdido ([Prueba recuperar contraseña.](#)):

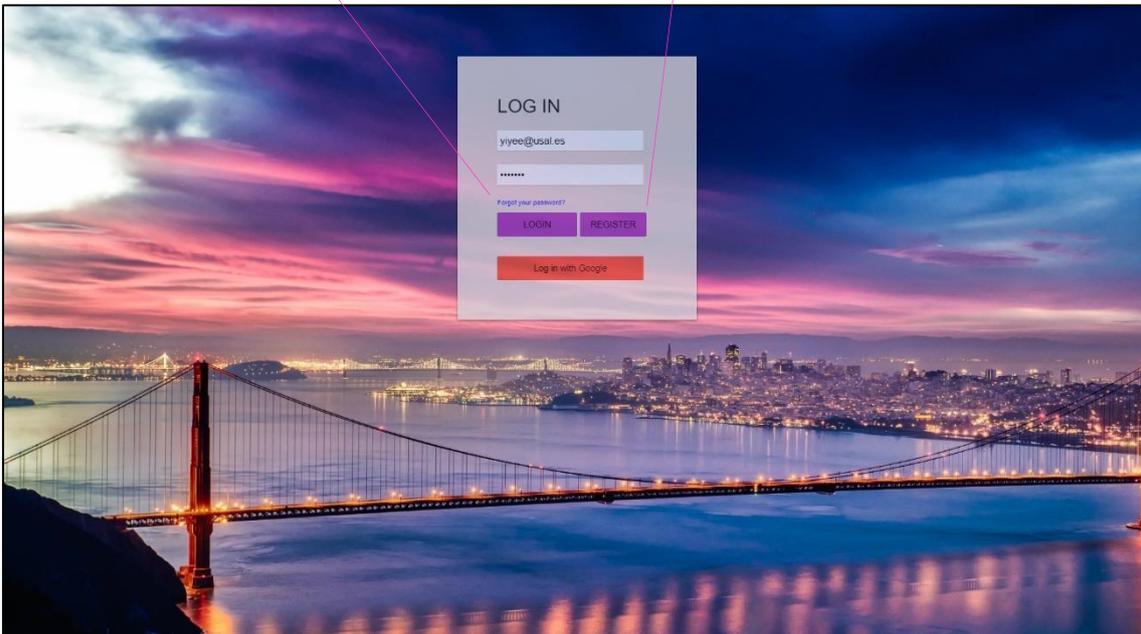


Ilustración 161. Inicio Sesión Web

3.2. Home

Una vez autenticados, pasaremos a la interfaz de home, dependiendo del rol que tengamos, en la home visualizaremos unas cosas u otras.

ADMINISTRADOR:

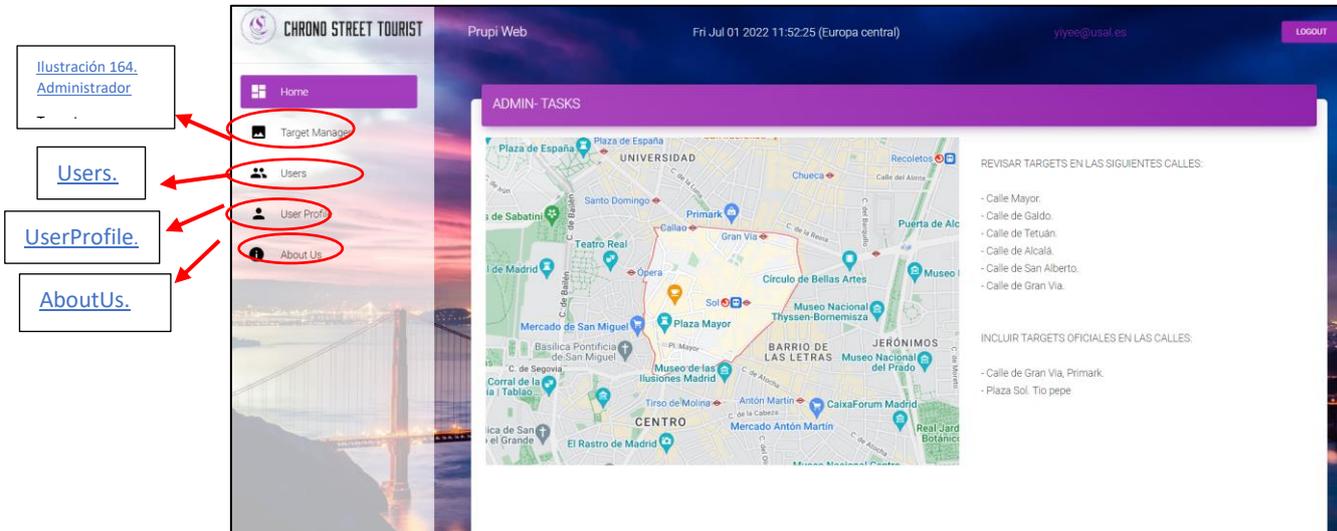


Ilustración 162. Home Administrador

En el home encontraremos las tareas que tiene que hacer el usuario administrador, en que zonas, las fechas de entrega de dichas tareas, y por último, un panel de usuarios sospechosos que hayan subido Targets sin respetar las políticas de uso.

USUARIO:

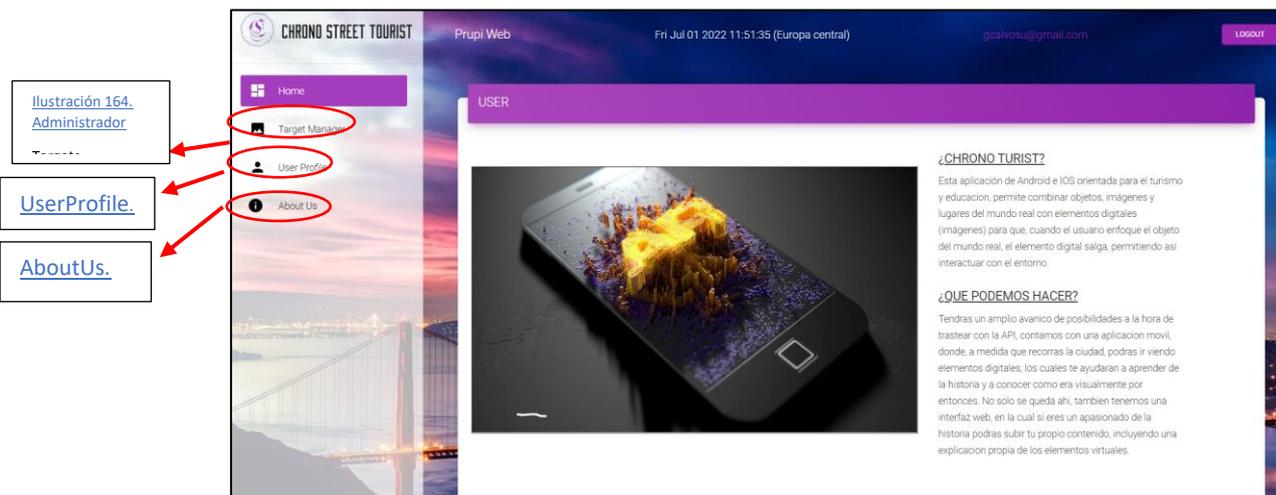


Ilustración 163. Home Usuario

En el home del usuario encontraremos información acerca del proyecto y de los fines que tiene. También encontraremos las posibilidades de uso que tiene la aplicación, tanto web como Android. Por último, para poner a la gente en situación, se cuenta brevemente la tecnología que usa la aplicación.

3.3. Manejo de Targets.

ID	Img	Img	Target Name	Correo	City	address	Active	Options
1			Catedral de Almudena borrar	yiyee@usal.es	Madrid	centro	<input checked="" type="checkbox"/>	EDIT DELETE
2			Museo del Prado	guillermocalvosu@gmail.com	Madrid	Centro	<input checked="" type="checkbox"/>	EDIT DELETE
3			Prado	guillermocalvosu@gmail.com	Madrid	centro	<input type="checkbox"/>	EDIT DELETE
4			Iglesia san pablo	guillermocalvosu@gmail.com	Madrid	Sur	<input type="checkbox"/>	EDIT DELETE
5			Prado oficial	gcalvosu@gmail.com	Madrid	Centro	<input type="checkbox"/>	EDIT DELETE

Ilustración 164. Administrador Targets Administrador.

Hay pequeñas diferencias en la tabla del usuario, filtrado por correo, ID, Correo, y la posibilidad de activar los Targets esta sustituida por el estado del Target:

Img	Target Name	City	address	Active	Options
	prueba	Madrid	micasa	pending	EDIT DELETE

Ilustración 165. Administrador Targets Usuario

En este apartado tendremos todo el control de los Targets, es decir, la configuración de cada monumento para que la podamos visualizar con Realidad Aumentada.

Dentro del manejo de Targets, podemos encontrarnos:

- Visualizar todos los Targets que hemos subido, en caso del administrador, visualizara todos los Targets. [Prueba ver lista de Targets y búsqueda con filtro.](#)
- Filtrar la lista de Targets, se complementan un filtrado a otro, es decir podemos filtrar por cualquiera de los filtros a la vez. [Prueba ver lista de Targets y búsqueda con filtro.](#)
- Subir un nuevo Target. [Prueba subir nuevo Target.](#)
- Editar Target existente. [Prueba ver Target y modificarlo.](#)
- Activar Target (Administrador).
- Borrar Target.

1.4. Usuarios. (Administrador)

Este apartado, al que solo puede acceder el administrador, contiene la creación de nuevos usuarios y la lista de usuarios:

Email	Username	Uid
gcalvosu@gmail.com	guillermo gomez	CH7yJRgh4pCHbRjkeinsSHgae2
yyee@usal.es	ADMINISTRADOR	VmS4SN1HwhejavzoxDMO1H9a5m2

Ilustración 166. Users.

Desde esta vista podremos registrar usuarios, rellenando el formulario de *User Register* y presionando el botón *create*. Tiene la posibilidad de crear usuarios y administradores. [Prueba crear Usuario/Administrador.](#)

En la parte inferior de la vista nos encontramos con la lista de los usuarios, mostrándonos atributos principales, si presionamos el botón *ver user*, nos redirige a la vista de *UserProfile*, que veremos en el siguiente apartado, con los datos del usuario y posibilidad de editarlos. [Prueba ver lista de usuarios e información de estos.](#)

1.5. Perfil del Usuario.

En este apartado veremos la información del usuario, con posibilidad de editarla, cambiar la contraseña, cambiar el rol de usuario(administrador) y borrar la cuenta: [Prueba modificar Usuario.](#) [Prueba cambiar Rol.](#) [Prueba borrar usuario.](#)

The screenshot displays the 'User Profile' page for an administrator. The profile information is as follows:

Field	Value
Name	guillermo
Surname	gomez
Username	eladminmaschuloedemas
PhoneNumber	654987321
Address	General Cadomas Campos 31
Info	Hola me llamo guille y soy ing informatico & Data scientist
Country	Madrid
State/Region	centro
City	Administrador
Role	Administrador

Legend:

- Nombre
- Apellido
- Nickname
- Teléfono
- Dirección
- Info del usuario
- País
- Ciudad
- Rol

Buttons: SAVE PROFILE, DELETE USER, CHANGE PASSWORD

Ilustración 167. UserProfile.

[Prueba modificar Usuario.](#)

Cambiar Contraseña

[Prueba borrar usuario.](#)

1.6. AboutUs.

En este apartado simplemente se recoge información sobre el creador del proyecto y sus colaboradores:

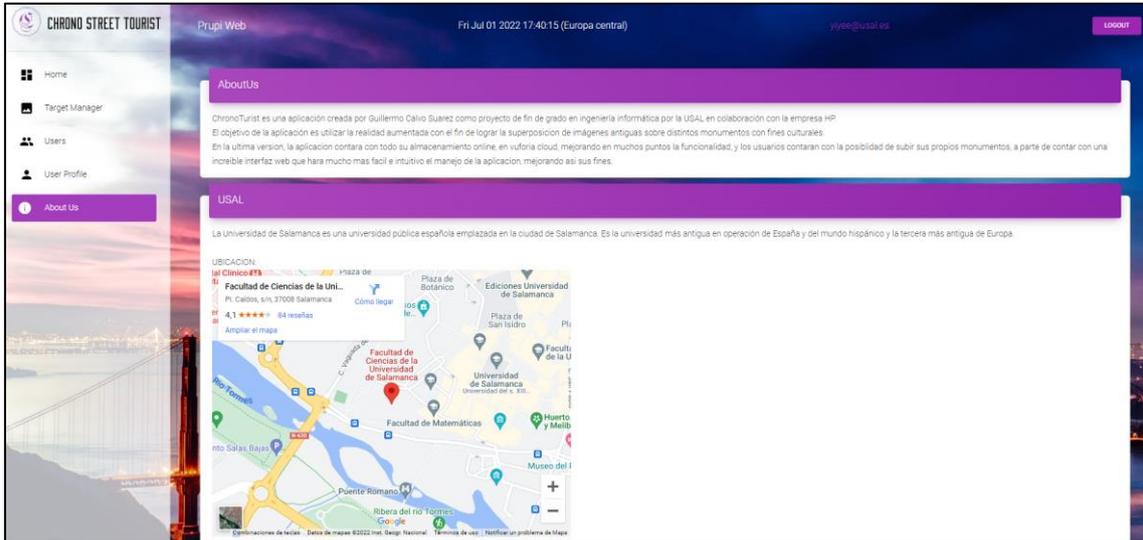


Ilustración 168. AboutUs.

9. Conclusiones

En este apartado se narran las conclusiones que se han obtenido tras finalizar el desarrollo del proyecto.

En primer lugar, quiero destacar que todos los objetivos del proyecto, tanto funcionales como no funcionales, han sido alcanzados.

Se ha desarrollado un sistema completo que permite visualizar monumentos históricos sobre la propia realidad, mostrando información sobre el mismo. También se ha desarrollado una interfaz web para el control de los Targets (monumentos), donde los usuarios y administradores pueden llevar un correcto manejo de Targets, ver la lista de Targets propios, ver información de un Target, publicar nuevo Target, modificarlo, borrarlos....

También se ha construido un robusto control de usuarios que permite diferenciar entre usuario y administrador y dotar a cada uno de su funcionalidad correspondiente.

Gracias al desarrollo del proyecto se han podido fijar las ideas adquiridas a lo largo del grado de ingeniería informática, llevando a la práctica estos conocimientos teóricos. Se ha adquirido conocimiento externo al grado sobre patrones arquitectónicos, herramientas, técnicas, bibliotecas, lenguajes... de gran utilidad y actualidad que hacen posicionar mi CV unos escalones más arriba. Se ha desarrollado un sistema software completo con la metodología ágil SCRUM.

El aprendizaje en Unity y Vuforia no ha sido de gran dificultad, ya que siempre me ha apasionado el desarrollo de videojuegos y he realizado algún curso (Máster Realidad Aumentada con Unity 3D, Vuforia y C#), por tanto, me he desenvuelto bien.

La mayor dificultad del proyecto ha sido la parte del manejo de Targets desde la propia web. La información de la documentación de Vuforia y de los foros es muy pobre y los SDK ofrecidos están desactualizados, por este motivo he diseñado mi propia API para el manejo de los Targets, la cual se puede usar como un paquete, para el manejo de Realidad Aumentada de otras plataformas.

El manejo de usuarios con Firebase ha sido muy buena elección, aunque no es fácil de usar, una vez aprendes lo básico, seguir aprendiendo es muy satisfactorio. También puedes usar Firebase como base de datos a tiempo real de otro tipo de colecciones.

Una de las dificultades con la que me he encontrado ha sido las limitaciones de las cuentas, tanto de Vuforia, como de Firebase. Si el uso del proyecto fuera reducido no habría problema, pero si el proyecto se escala, haría falta contratar el plan adecuado. Otra limitación dentro de Firebase ha sido la necesidad de introducir un método de pago para el uso de las funciones Cloud, necesarias para ejecutar funciones desde el servidor, el cual hemos solucionado emulándolas en el puerto 5001.

En resumen, estoy bastante satisfecho con el desarrollo del proyecto, he aprendido lenguajes, herramientas y técnicas de las que no sabía nada y son punta en el sector de la Informática, he aprendido a organizar proyectos a gran escala, a buscar información que de verdad sea valiosa, a ser muy organizado con el código.... Pero sobre todo me he dado cuenta, que sea el problema que sea, tengo la capacidad de buscarle una solución óptima.

Me siento muy orgulloso de haber sido seleccionado por una empresa tan importante en la historia de la informática, HP, para la realización de un proyecto propio de la empresa, con las tecnologías más punteras en el mercado actualmente, Realidad Aumentada, inteligencia artificial...

10. Líneas de trabajo futuras

En esta sección de la memoria se narran las posibles modificaciones o ampliaciones del proyecto en el futuro:

- Adaptar la aplicación a teléfonos con sistema operativo IOS, ampliando así el número de dispositivos que pueden hacer uso de la aplicación.
- Añadir la funcionalidad de que cuando se reconozca el monumento, se reproduzca un audio sobre la historia de este, permitiendo así que el usuario no tenga que acceder a páginas web externas de la aplicación y escuchar la información referente al monumento mientras lo contempla en la vida real.
- Añadir un mapa que te muestre los monumentos más importantes de la ciudad y los más cercanos.
- Añadir rutas, al punto anterior, para hacer mucho más agradable la experiencia que da la aplicación de guía turístico.
- Añadir funcionalidad para poner imágenes en GoogleStreetView.

9. Bibliografía

- [1] Casini, J. (2019). Realidad Virtual y video en 360°. Orígenes, desarrollos actuales y su uso en el periodismo. In *XXI° Congreso de la Red de Carreras de Comunicación Social y Periodismo*. Escuela de Ciencias de la Comunicación, Facultad de Humanidades (UNSa).
- [2] Claudio Servin(1968). Espada de damócles, 26, agosto 1968. <https://proyectoidis.org/espada-de-damocles/>. [Último acceso 4/2022]
- [3] Two Reality. Realidad aumentada en educación: ventajas para el usuario., 2010. <https://www.tworeality.com/realidad-aumentada-en-educacion-y-sus-grandes-ventajas-para-el-usuario/>. [Último acceso 4/2022]
- [4] Won Hyung. A study on interactive media art to apply emotion recognition, 2014. https://www.researchgate.net/figure/Myron-Krueger-Videoplance-1975_fig1_274621011. [Último acceso 4/2022]
- [5] Louis B. Rosenberg , 1992, THE USE OF VIRTUAL FIXTURES AS PERCEPTUAL OVERLAYS TO ENHANCE OPERATOR PERFORMANCE IN REMOTE ENVIRONMENTS.
- [6] Argo Project. A brief history of augmented reality (infographic), 2018. <https://iat.es/tecnologias/realidad-aumentada/>. [Último acceso 4/2022]
- [7] Won Hyung. A study on interactive media art to apply emotion recognition, 2014.
- [8] Raúl Domínguez Sánchez. ¿qué es y cómo ha evolucionado la realidad aumentada?, 20016.
- [9] ChronoTulist. Proyecto de final de realizado por José Antonio Cerrato Casillas, universidad de Burgos, 2021.
- [10] Communications. Los siete usos de la Realidad Aumentada que ya están aquí, 2018. <https://www.bbva.com/es/siete-usos-realidad-aumentada-ya-estan-aqui/>. [Último acceso 4/2022]
- [11] Claudia Rodríguez. ¿cuáles son los elementos que componen a una imagen digital y para qué sirven?, 2018. <https://alphauniverse-latin.com/notas/elementos-que-componen-la-imagen>. [Último acceso 4/2022]
- [12] Qué es un CMS, características y cuáles son los más populares. [https://www.iebschool.com/blog/que-es-un-cms-e-commerce/#:~:text=Un%20CMS%20\(Content%20Management%20System,cualquier%20tipo%20de%20p%C3%A1gina%20web](https://www.iebschool.com/blog/que-es-un-cms-e-commerce/#:~:text=Un%20CMS%20(Content%20Management%20System,cualquier%20tipo%20de%20p%C3%A1gina%20web). [Último acceso 4/2022]
- [13] ¿Qué es una API Rest? 8-mayo-2020: <https://www.redhat.com/es/topics/api/what-is-a-rest-api> [Último acceso 4/2022]

- [14] Qué es REST, RESTful, API RESTful y JSON: <https://bravedeveloper.com/2021/09/01/que-es-rest-restful-api-restful-y-json/> [Último acceso 4/2022]
- [15] Didier Stricker, John Karigiannis, Ioannis T Christou, Tim Gleue, and Nikos Ioannidis. Augmented reality for visitors of cultural heritage sites. In Proc. of Int. Conf. on Cultural and Scientific Aspects of Experimental Media Spaces, pages 89–93, 2001.
- [16] Kari Rainio, Petri Honkamaa, and Kaisa Spilling. Presenting historical photos using augmented reality. *ARea*, 15:11–12, 2015.
- [17] Julio Alonso Arévalo. Aurasma: app de realidad aumentada para insertar una historia en un libro en el espacio físico., 2016. <https://universoabierto.org/2016/09/02/aurasma-app-de-realidad-aumentada-para-insertar-una-historia-en-un-libro-en-el-espacio-fisico/comment-page-1/>. [Último acceso 4/2022]
- [18] Kari Rainio, Petri Honkamaa, and Kaisa Spilling. Presenting historical photos using augmented reality. *ARea*, 15:11–12, 2015.
- [19] Ana de la Fuente. Aurasma., 2015. <http://raenelaula.blogspot.com/p/aurasma.html>. [Último acceso 4/2022]
- [20] Tatiana Grapsas. Realidad aumentada: ¿qué es, cómo funciona y para qué sirve?, 2019. <https://rockcontent.com/es/blog/realidad-aumentada/>. [Último acceso 4/2022]
- [21] Andrés Schiaffarino, 2021, Modelo cliente-servidor <https://blog.infranetworking.com/modelo-cliente-servidor>. [Último acceso 4/2022]
- [22] Sandra Garrido Sotomayor, 9 diciembre, 2021. Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa. <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/> [último acceso 4/2022]
- [23] María Tena, 2021. ¿Qué es una metodología ‘agile’?. <https://www.bbva.com/es/metodologia-agile-la-revolucion-las-formas-trabajo/> [último acceso 4/2022]
- [24] Cristian A. Remón, Pablo Thomas. Análisis de Estimación de Esfuerzo aplicando Puntos de Caso de Uso.
- [25] Roy K. Clemmons, Febrero 2006. Project Estimation With Use Case Points.
- [26] *The only way to make the deadline—the only way to go fast—is to keep the code as clean as possible at all times.* Robert C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*.
- [27] *What is MVC?* Advantages and Disadvantages of MVC, Jithin, October 28th, 2016.