

▪ Trabajo Fin de Grado ▪
Ingeniería de Software

Gestión de grabaciones de marcha de pacientes de
Parkinson usando Kinect

Asier Tamayo San Isidoro

Junio 2018

Resumen

Este documento corresponde a la memoria del Trabajo de Fin de Grado “Gestión de grabaciones de marcha de pacientes de Parkinson usando Kinect”, desarrollado por Asier Tamayo San Isidoro para la titulación del Grado en Ingeniería Informática en la especialidad de Ingeniería del Software, en la Facultad de Informática de la Universidad UPV/EHU de Donostia-San Sebastián.

En este proyecto se ha desarrollado una aplicación web para gestionar las grabaciones de marcha de pacientes de Parkinson. Esta aplicación nace como una propuesta del grupo BDI de la Facultad de Informática de San Sebastián y de la necesidad de una aplicación con la que gestionar las grabaciones realizadas como parte de la investigación que desarrollan en torno al Parkinson.

El proyecto se ha realizado bajo la dirección del profesor de la UPV/EHU y miembro del grupo de investigación BDI, Alfredo Goñi Sarriguren, y supervisión de Urko Lopez de Abetxuko, miembro también del grupo de investigación.

El presente documento, describe el proceso seguido para desarrollar la aplicación, así como los principales objetivos y requisitos del proyecto, las tecnologías utilizadas, el diseño y arquitectura de la aplicación, detalles de la implementación y pruebas realizadas. Además, se incluye el [manual de usuario](#) en forma de anexo.

La aplicación desarrollada está disponible a través del siguiente enlace: <https://gestion-pacientes-parkinson.firebaseio.com/>.

Índice

1. Antecedentes.....	15
1.1. Grabaciones.....	16
1.1.1. Kinect.....	16
1.1.2. Sesiones de grabación	18
1.1.3. Fichero de la grabación.....	20
1.2. Gestión de pacientes y grabaciones	21
1.3. Aplicación	21
2. Objetivos.....	23
2.1. Requisitos generales.....	24
2.2. Pila de Producto.....	24
3. Planificación.....	27
3.1. Alcance	28
3.2. Objetivos.....	28
3.3. Exclusiones	28
3.4. EDT.....	29
3.5. Diagrama de Gantt.....	30
3.6. Riesgos.....	31
3.7. Metodología	32
3.7.1. Equipo del proyecto.....	33
3.7.2. Iteraciones	33
4. Tecnologías.....	35
4.1. Elección de tecnologías	36
4.2. Tecnologías para el desarrollo de la aplicación.....	38
4.2.1. Lado Cliente.	38
4.2.2. Lado Servidor.....	40
4.3. Gestión del proyecto.	42
5. Requisitos funcionales	43
5.1. Diagrama de casos de uso	44
5.2. Flujo de eventos	45
5.2.1. Recuperar contraseña.....	45
5.2.2. Ver notas.....	46
6. Diseño.....	49

6.1. Arquitectura.....	50
6.1.1. Modelo	50
6.1.2. Vista	51
6.1.3. Controlador.....	51
6.2. Modelo de datos.....	51
6.3. Seguridad.....	54
6.3.1. Gestión segura de los datos en Firebase	54
6.3.2. Encriptación de los datos en local.....	57
6.4. Diagramas de secuencia	60
7. Implementación.....	65
7.1. Vistas	66
7.2. Controladores y Modelos	67
7.3. Comunicación con Firebase Database	68
7.3.1. Lectura de datos en Firebase.....	69
7.3.2. Escritura de datos en Firebase.....	69
7.3.3. Actualización de datos en Firebase.....	70
7.3.4. Borrado de datos en Firebase.....	70
7.4. Comunicación con Firebase Storage.....	70
7.4.1. Lectura de ficheros en Firebase	71
7.4.2. Almacenar ficheros en Firebase.....	71
7.5. Encriptado de los datos	72
7.5.1. Proceso de obtención de la clave de cifrado	72
7.5.2. Proceso de cifrado	73
7.5.2. Proceso de descifrado	74
7.6. Visualización de las grabaciones.....	75
7.6.1. Herramientas utilizadas	75
7.6.2. Dibujado del esqueleto	76
7.6.3. Visualización sincronizada de vídeo y esqueleto	78
7.6.4. Herramientas de reproducción.....	79
7.7. Dificultades durante el desarrollo.....	81
8. Pruebas.....	85
HU6. Visualización de esqueletos en sincronía con el vídeo.....	86
HU8. Encriptación de la información	89
9. Seguimiento del proyecto.....	93
9.1. Diagrama de Gantt final.....	94

9.2. Dedicaciones.....	95
10. Conclusiones.....	97
10.1. Metodología y tecnologías empleadas	98
10.2. Conocimientos adquiridos y experiencia personal	99
10.3. Puntos de extensión y mejora	99
11. Bibliografía.....	101
Anexo A. Historias de Usuario	105
Anexo B. Flujo de eventos	113
1. Iniciar sesión	114
2. Recuperar contraseña.....	115
3. Crear paciente	116
4. Editar paciente.....	117
5. Ver pacientes	118
6. Eliminar paciente/s	120
7. Crear grabación	121
8. Editar grabación.....	122
9. Ver grabaciones	123
10. Editar nota	125
11. Eliminar nota/s	126
12. Eliminar grabación/es	127
13. Visualizar grabación.....	128
14. Descargar grabación	130
Anexo C. Pruebas.....	131
HU1. Usuarios.....	132
HU2. Entorno pacientes.....	136
HU3. Entorno grabaciones.....	158
HU4. Visualización grafica de los esqueletos	183
HU5. Visualización de vídeos	186
HU7. Herramientas reproducción vídeo/esqueleto.....	189
HU9. Despliegue en Firebase.....	194
HU10. Sincronización GitHub.....	195
HU11. Estructura fichero de datos	197
Anexo D. Actas reuniones.....	198
Sprint 1.	199
Sprint 2.	199

Sprint 3.	200
Sprint 4.	200
Sprint 5.	201
Sprint 6.	201
Sprint 7.	202
Anexo E. Manual de usuario	204
1. Usuarios.....	206
1.1. Inicio de sesión	206
1.2. Cambio de contraseña	207
1.3. Clave de cifrado	208
2. Pacientes	208
2.1. Registrar paciente.....	208
2.2. Visualizar pacientes	210
2.3. Editar pacientes	210
2.4. Eliminar pacientes.....	211
2.5. Asignar grabación a paciente.....	211
3. Grabaciones	212
3.1. Registrar grabación	212
3.2. Visualizar grabaciones	213
3.3. Editar grabaciones	214
3.4. Eliminar grabaciones.....	214
3.5. Reproducir grabaciones	215
3.6. Descargar grabación	215
3.7. Previsualizar archivo CSV	216

Lista de Figuras y Tablas

Ilustración 1. Kinect	16
Ilustración 2. Representación esqueleto Kinect.....	17
Ilustración 3. Disposición cámaras sesión de grabación	18
Ilustración 4. Sesión grabación. Foto 1	18
Ilustración 5. Sesión grabación. Foto 2	19
Ilustración 6. Sesión grabación. Aplicaciones utilizadas	19
Ilustración 7. Aplicación anterior	21
Ilustración 8. EDT del proyecto	29
Ilustración 9. Diagrama de Gantt.....	30
Ilustración 10. Scrum, extraído de [6].....	32
Ilustración 11. Logo AngularJS	36
Ilustración 12. Logo de Firebase	40
Ilustración 13. Planes de Firebase	41
Ilustración 14. Diagrama de casos de uso.....	44
Ilustración 15. Recuperar contraseña.....	45
Ilustración 16. Ver notas.....	47
Ilustración 17. Arquitectura del proyecto.....	50
Ilustración 18. Modelo de datos	52
Ilustración 19. Firebase Database.....	53
Ilustración 20. Firebase Storage	53
Ilustración 21. Transporte cifrado de los datos	54
Ilustración 22. Capas de encriptación usadas en Firebase, extraída de [17]	55
Ilustración 23. Gestión de datos en Google, adaptada de [18].....	55
Ilustración 24. Gestión de claves para desencriptar un fragmento de datos, adaptado de [19] .	56
Ilustración 25. Visualización por defecto de los datos en Firebase.....	57
Ilustración 26. Visualización de los datos cifrados en Firebase.....	60
Ilustración 27. Diagrama Secuencia Iniciar sesión	61
Ilustración 28. Diagrama Secuencia Ver pacientes	62
Ilustración 29. Diagrama Secuencia Visualizar grabaciones.....	63
Ilustración 30. Vistas principales de la aplicación y hoja de estilos.....	66
Ilustración 31. Plantilla de una vista	66
Ilustración 32. Controladores de la aplicación.....	67
Ilustración 33. Cookie con la clave de encriptación	72
Ilustración 34. Dependencias columna vertebral	76
Ilustración 35. Visualización frontal y lateral del esqueleto.....	77
Ilustración 36. Visualización sincronizada de vídeo y esqueleto.....	78
Ilustración 37. Visualización grabación con herramientas.....	80
Ilustración 38. Visualización grabación. Añadir nota	80
Ilustración 39. Visualización grabación. Visualizar nota.....	80
Ilustración 40. Visualización esqueleto con ThreeJS.....	81
Ilustración 41. Visualización en ThreeJS (izquierda) con errores y esqueleto en GeoGebra (derecha)	82
Ilustración 42. Test HU6.1 Prueba 1	86

Ilustración 43. Test HU6.2 Prueba 1	86
Ilustración 44. Test HU6.3 Prueba 1	87
Ilustración 45. Test HU6.4 Prueba 1	87
Ilustración 46. Test HU6.5 Prueba 1	88
Ilustración 47. Test HU8.1 Prueba 1	89
Ilustración 48. Test HU8.2 Prueba 1	90
Ilustración 49. Test HU8.3 Prueba 1	91
Ilustración 50. Test HU8.3 Prueba 2	91
Ilustración 51. Test HU8.4 Prueba 1	92
Ilustración 52. Test HU8.4 Prueba 2	92
Ilustración 53. Versión final diagrama de Gantt	94
Ilustración 54. Inicio de sesión.....	114
Ilustración 55. Recuperar contraseña.....	115
Ilustración 56. Crear paciente.....	116
Ilustración 57. Editar paciente.....	117
Ilustración 58. Ver pacientes	119
Ilustración 59. Eliminar paciente/s	120
Ilustración 60. Crear grabación.....	121
Ilustración 61. Editar grabación	122
Ilustración 62. Ver grabaciones	124
Ilustración 63. Editar grabación	125
Ilustración 64. Eliminar nota/s.....	126
Ilustración 65. Eliminar grabación	127
Ilustración 66. Visualizar grabación	129
Ilustración 67. Descargar grabación	130
Ilustración 68. Test HU1.1 Prueba 1	132
Ilustración 69. Test HU1.1 Prueba 2	133
Ilustración 70. Test HU1.1 Prueba 3	133
Ilustración 71. Test HU1.1 Prueba 4	134
Ilustración 72. Test HU1.1 Prueba 5	134
Ilustración 73. Test HU1.1 Prueba 6	134
Ilustración 74. Test HU1.2 Prueba 1	135
Ilustración 75. Test HU1.3 Prueba 1	135
Ilustración 76. Test HU1.3 Prueba 2. Notificación	136
Ilustración 77. Test HU1.3 Prueba 2. Email recuperación.....	136
Ilustración 78. Test HU2.1 Prueba 1 y Prueba 2	137
Ilustración 79. Test HU2.1 Prueba 3	137
Ilustración 80. Test HU2.2 Prueba 4	138
Ilustración 81. Test HU2.2 Prueba 5	138
Ilustración 82. Test HU2.2 Prueba 6	138
Ilustración 83. Test HU2.2 Prueba 1	139
Ilustración 84. Test HU2.2 Prueba 2	139
Ilustración 85. Test HU2.2 Prueba 3	140
Ilustración 86. Test HU2.2 Prueba 4	140
Ilustración 87. Test HU2.3 Prueba 1	141
Ilustración 88. Test HU2.4 Prueba 1. Paciente creado por usuario1	142
Ilustración 89. Test HU2.4. Prueba 1. Sección pacientes del usuario2	142

Ilustración 90. Test HU2.4 Prueba 2. Sección pacientes del usuario2	142
Ilustración 91. Test HU2.4 Prueba 2. Paciente creado por usuario1	142
Ilustración 92. Test HU2.5 Prueba 1	143
Ilustración 93. Test HU2.5 Prueba 2	143
Ilustración 94. Test HU2.6 Prueba 1	144
Ilustración 95. Test HU2.7 Prueba 1	145
Ilustración 96. Test HU2.8 Prueba 1	145
Ilustración 97. Test HU2.8 Prueba 2	146
Ilustración 98. Test HU2.8 Prueba 3	146
Ilustración 99. Test HU2.8 Prueba 4	147
Ilustración 100. Test HU2.8 Prueba 5	147
Ilustración 101. Test HU2.8 Prueba 6	148
Ilustración 102. Test HU2.8 Prueba 7	148
Ilustración 103. Test HU2.9 Prueba 1	149
Ilustración 104. Test HU2.9 Prueba 2	149
Ilustración 105. Test HU2.9 Prueba 3	150
Ilustración 106. Test HU2.10 Prueba 1	150
Ilustración 107. Test HU2.10 Prueba 2	151
Ilustración 108. Test HU2.10 Prueba 3	151
Ilustración 109. Test HU2.10 Prueba 4	152
Ilustración 110. Test HU2.10 Prueba 5	152
Ilustración 111. Test HU2.10 Prueba 6	153
Ilustración 112. Test HU2.10 Prueba 7	153
Ilustración 113. Test HU2.11 Prueba 1	154
Ilustración 114. Test HU2.11 Prueba 2	154
Ilustración 115. Test HU2.11 Prueba 3	155
Ilustración 116. Test HU2.11 Prueba 4	155
Ilustración 117. Test HU2.11 Prueba 5	156
Ilustración 118. Test HU2.11 Prueba 6	156
Ilustración 119. Test HU2.11 Prueba 7	156
Ilustración 120. Test HU2.11 Prueba 8	157
Ilustración 121. Test HU2.11 Prueba 9	157
Ilustración 122. Test HU3.1 Prueba 1 y 2	158
Ilustración 123. Test HU3.1 Prueba 3	159
Ilustración 124. Test HU3.1 Prueba 4	159
Ilustración 125. Test HU3.1 Prueba 5	160
Ilustración 126. Test HU3.1 Prueba 6	160
Ilustración 127. Test HU3.2 Prueba 1	161
Ilustración 128. Test HU3.2 Prueba 2	161
Ilustración 129. Test HU3.2 Prueba 3	162
Ilustración 130. Test HU3.2 Prueba 4	162
Ilustración 131. Test HU3.3 Prueba 1	163
Ilustración 132. Test HU3.4 Prueba 1. Grabación creada por usuario1	164
Ilustración 133. Test HU3.4 Prueba 1. Sección grabaciones del usuario2	164
Ilustración 134. Test HU3.4 Prueba 2. Grabación creada por usuario1	165
Ilustración 135. Test HU3.4 Prueba 2. Sección grabaciones del usuario2	165
Ilustración 136. Test HU3.5 Prueba 1	165

Ilustración 137. Test HU3.5 Prueba 2	166
Ilustración 138. Test HU3.6 Prueba 1	167
Ilustración 139. Test HU3.7 Prueba 1	167
Ilustración 140. Test HU3.7 Prueba 2	168
Ilustración 141. Test HU3.7 Prueba 3	168
Ilustración 142. Test HU3.7 Prueba 4	169
Ilustración 143. Test HU3.7 Prueba 5	169
Ilustración 144. Test HU3.7 Prueba 6	170
Ilustración 145. Test HU3.8 Prueba 1	170
Ilustración 146. Test HU3.8 Prueba 2	171
Ilustración 147. Test HU3.8 Prueba 3	171
Ilustración 148. Test HU3.9 Prueba 1	172
Ilustración 149. Test HU3.9 Prueba 2	172
Ilustración 150. Test HU3.9 Prueba 3	173
Ilustración 151. Test HU3.9 Prueba 4	173
Ilustración 152. Test HU3.9 Prueba 5	174
Ilustración 153. Test HU3.9 Prueba 6	174
Ilustración 154. Test HU3.9 Prueba 7	175
Ilustración 155. Test HU3.10 Prueba 1	175
Ilustración 156. Test HU3.10 Prueba 2	176
Ilustración 157. Test HU3.10 Prueba 3	176
Ilustración 158. Test HU3.10 Prueba 4	177
Ilustración 159. Test HU3.10 Prueba 5	177
Ilustración 160. Test HU3.10 Prueba 6	178
Ilustración 161. Test HU3.10 Prueba 7	178
Ilustración 162. Test HU3.10 Prueba 8	179
Ilustración 163. Test HU3.10 Prueba 9	179
Ilustración 164. Test HU3.10 Prueba 10	180
Ilustración 165. Test HU3.11 Prueba 1	180
Ilustración 166. Test HU3.12 Prueba 1	181
Ilustración 167. Test HU3.12 Prueba 1	181
Ilustración 168. Test HU3.13 Prueba 2 Notas	182
Ilustración 169. Test HU3.13 Prueba 2. JSON	182
Ilustración 170. Test HU4.1 Prueba 1	183
Ilustración 171. Test HU4.2 Prueba 1	184
Ilustración 172. Test HU4.3 Prueba 1	184
Ilustración 173. Test HU4.4 Prueba 1	185
Ilustración 174. Test HU5.1 Prueba 1	186
Ilustración 175. Test HU5.2 Prueba 1	186
Ilustración 176. Test HU5.2 Prueba 2	187
Ilustración 177. Test HU5.3 Prueba 1	187
Ilustración 178. Test HU5.3 Prueba 2	188
Ilustración 179. Test HU 7.1 Prueba 1	189
Ilustración 180. Test HU7.2 Prueba 1	190
Ilustración 181. Test HU7.3 Prueba 1	191
Ilustración 182. Test HU7.4 Prueba 1	192
Ilustración 183. Test HU7.5 Prueba 1	193

Ilustración 184. Test HU9.1 Prueba 1	194
Ilustración 185. Test HU10.1 Prueba 1	195
Ilustración 186. Test HU10.2 Prueba 1	196
Ilustración 187. Test HU11.1 Prueba 1	197
Ilustración 188. Inicio de sesión.....	206
Ilustración 189. Inicio de sesión. Cambio de contraseña.....	207
Ilustración 190. Cambio de contraseña	207
Ilustración 191. Menú Pacientes	208
Ilustración 192. Campo extra Paciente	209
Ilustración 193. Registro paciente	209
Ilustración 194. Visualizar pacientes.....	210
Ilustración 195. Editar paciente.....	211
Ilustración 196. Eliminar pacientes.....	211
Ilustración 197 Menú grabaciones	212
Ilustración 198. Campo extra Grabación	212
Ilustración 199. Registro grabación	213
Ilustración 200. Visualizar grabaciones.....	213
Ilustración 201. Editar grabación	214
Ilustración 202. Eliminar grabaciones.....	214
Ilustración 203. Reproducción grabación	215
Ilustración 204. Previsualizar CSV	216
Tabla 1. Formato fichero grabación.....	20
Tabla 2. Riesgos del proyecto	31
Tabla 3. Sprints del proyecto	33
Tabla 4. Comparativa ReactJS AngularJS.....	36
Tabla 5. Comparativa Firebase, Pusher y PubNub	37
Tabla 6. Comparación algoritmos de encriptación	58
Tabla 7. Dedicación total	95
Tabla 8. Dedicaciones por iteración.....	95

Lista de Fragmentos de Código

Código 1. Modelo paciente.....	68
Código 2. Definir instancia referencia Firebase	68
Código 3. Lectura en Firebase	69
Código 4. Escritura en Firebase	69
Código 5. Actualización Firebase	70
Código 6. Borrado en Firebase	70
Código 7. Definir instancia referencia Firebase Storage	70
Código 8. Almacenar fichero de vídeo	71
Código 9. Almacenar fichero de texto	71
Código 10. Encriptación Strings	73
Código 11. Encriptación CSV/JSON	73
Código 12. Encriptación vídeo	73
Código 13. Desencriptado Strings.....	74
Código 14. Desencriptado CSV/JSON.....	74
Código 15. Desencriptado vídeo.....	75
Código 16. Vector jerarquía de articulaciones.....	77
Código 17. Pseudocódigo pintado esqueletos.....	77

1. Antecedentes

En este capítulo se exponen los antecedentes del proyecto, por un lado, se detalla cómo es la gestión actual de los pacientes y las grabaciones y cuál es la aplicación que se utiliza para visualizar las grabaciones. Por otro lado, se explica cómo se llevan a cabo las grabaciones, qué herramientas se emplean para ello y cuál es la información que se obtiene tras una sesión de grabación.

El proyecto que se va a desarrollar es una idea propuesta por el grupo de investigación BDI de la UPV/EHU [1]. Este grupo centra parte de su investigación en torno a la eHealth [2], conjunto de Tecnologías de la Información y la Comunicación (TICs) que, a modo de herramientas, se emplean en el entorno sanitario en materia de prevención, diagnóstico, tratamiento, seguimiento, así como en la gestión de la salud, ahorrando costes al sistema sanitario y mejorando la eficacia de este. La investigación se centra en el Parkinson y en cómo por la forma en la que una persona camina se puede detectar si padece la enfermedad y en qué fase de la misma está. En el Parkinson se pueden distinguir 5 fases o estadios, en las que los síntomas se van agravando poco a poco a lo largo del tiempo. Para comprobar el estado de un paciente realizan grabaciones de marcha mediante la cámara de detección de movimientos Kinect, explicada en más detalle en el capítulo [1.1. Grabaciones](#), y posteriormente, mediante una serie de funciones desarrollada en R, analizan la grabación. También cuentan con una aplicación con la que visualizar las grabaciones. A continuación, se explica cómo se realizan las grabaciones y qué se obtiene de una grabación, cómo se gestionan los pacientes y las grabaciones y las características de la aplicación actual.

1.1. Grabaciones

En esta sección se aborda el tema de las grabaciones. Por un lado, se introduce Kinect [3], la herramienta utilizada para llevar a cabo las grabaciones de marcha de los pacientes. Por otro lado, se explica cómo se desarrolla una sesión de grabación. También se detalla cuál es el formato del fichero que se obtiene como resultado de la grabación. Este fichero contiene toda la información de la grabación.

1.1.1. Kinect

Kinect es una línea de dispositivos de detección de movimiento que fue producida por Microsoft para las consolas de videojuegos Xbox 360 y Xbox One, además de para PC. Está basado en un periférico complementario estilo cámara web. Este dispositivo permite a los usuarios controlar e interactuar con la consola sin necesidad de tener contacto físico con un controlador de videojuegos tradicional. La interacción se realiza a través de una interfaz de usuario natural que usa gestos y comandos de voz. Su éxito no se limita a su uso como control en videojuegos, sino también a que en poco tiempo la comunidad investigadora encontró aplicaciones de Kinect para las que en un principio no había sido diseñado. La tecnología de detección de Kinect competía directamente con cámaras 3D que resultaban muchos más caras de conseguir.



Ilustración 1. Kinect

Kinect permite obtener imágenes de profundidad de una escena e identificar al usuario y sus movimientos. Está formado por una cámara de vídeo, una cámara de profundidad basada en infrarrojos y una serie de cuatro micrófonos. Los datos obtenidos permiten visualizar la escena en tres dimensiones y proporcionan información sobre la posición del usuario, más concretamente sobre la ubicación de sus articulaciones. Para que la detección del usuario se realice correctamente, este debe situarse a una distancia de la cámara de entre 0.8 y 4 metros.

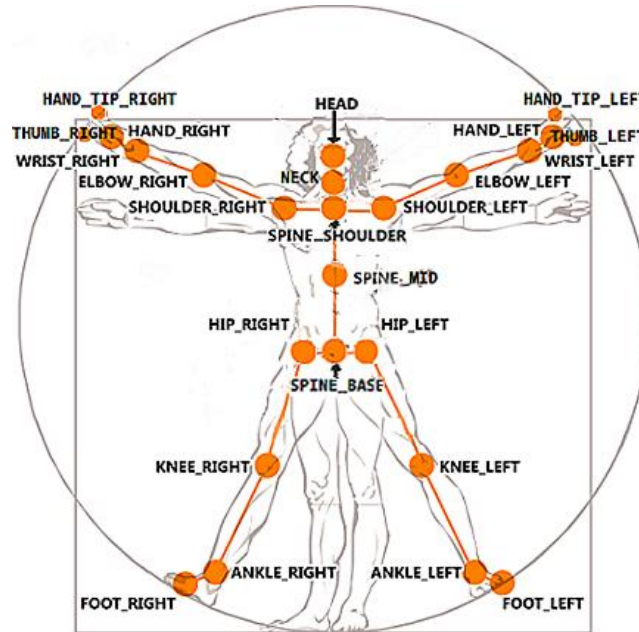


Ilustración 2. Representación esqueleto Kinect

El software de Kinect recupera un esqueleto que representa al paciente. Este esqueleto está compuesto por 25 articulaciones y obtiene la posición de cada una de ellas a una frecuencia aproximada de 30 lecturas (frames) por segundo. Los puntos que representan las articulaciones están definidos por coordenadas tridimensionales (X, Y, Z). Para los ejes X e Y el origen de coordenadas se sitúa en el centro de la imagen que obtiene la cámara, independientemente de la posición en la que se encuentre el usuario. Estos valores se dan en metros. Para el eje Z el origen de coordenadas se encuentra en la cámara, los valores que toma vienen dados en metros desde la cámara, con un rango de valores entre 0,8 y 4. En caso de no poder obtener la posición Z de un punto, ya sea porque se encuentra demasiado cerca o demasiado lejos, el valor que toma es 0.

1.1.2. Sesiones de grabación

Para realizar una sesión de grabación son necesarias dos Kinect, conectadas cada una de ellas a un ordenador portátil. La habitación en la que se desarrolle la grabación deberá tener un espacio aproximado de 6 metros de largo por 2 de ancho. La disposición de las cámaras será la de la siguiente imagen:

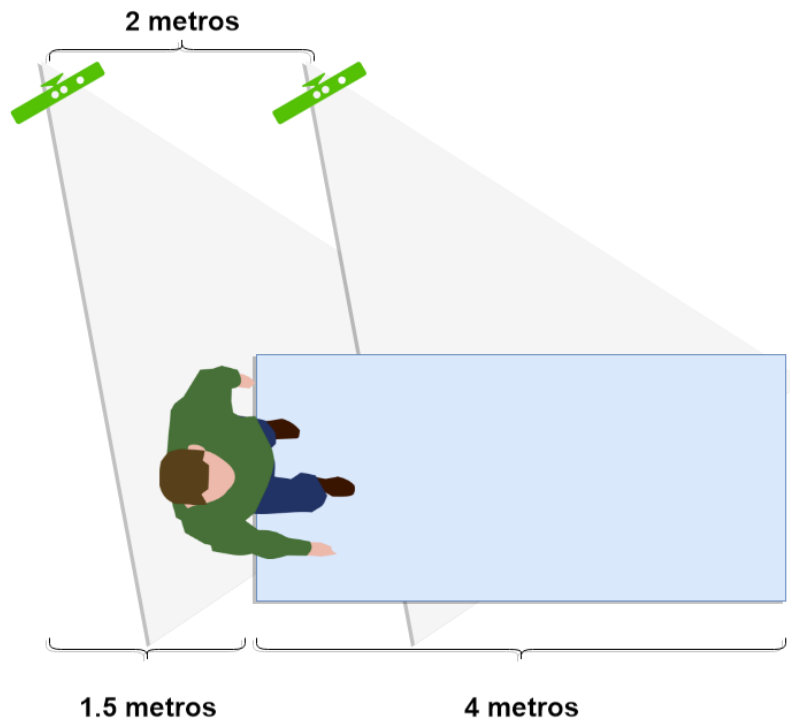


Ilustración 3. Disposición cámaras sesión de grabación

Las cámaras se colocan sobre un trípode a un metro de altura, inclinadas 60° horizontalmente, y con una separación de dos metros entre ellas. El recorrido que realiza el paciente empieza a 1.5 metros de la primera cámara, distancia mínima para que Kinect recoja correctamente la información, y avanza aproximadamente 4 metros, a continuación, el paciente gira 180° y vuelve al punto de partida. Este paseo lo repite 4 veces.



Ilustración 4. Sesión grabación. Foto 1



Ilustración 5. Sesión grabación. Foto 2

Durante la sesión de grabación se ejecutan dos aplicaciones de forma paralela. Por un lado, Microsoft Kinect Studio y por el otro la aplicación desarrollada por el grupo de investigación, con la que se obtiene el fichero CSV con las coordenadas de cada una de las articulaciones. En ambas aplicaciones se muestra el esqueleto del paciente durante la grabación, tal y como se puede observar a continuación:

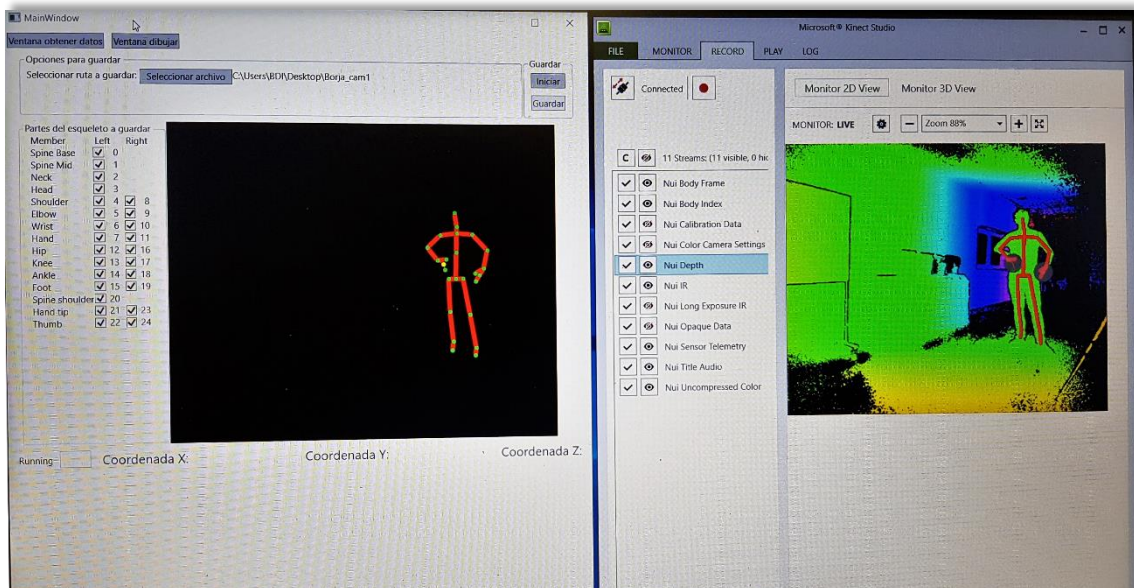


Ilustración 6. Sesión grabación. Aplicaciones utilizadas

1.1.3. Fichero de la grabación

Toda la información relativa al paciente generada por Kinect durante la grabación es tratada y posteriormente exportada a un archivo CSV. Los archivos CSV (del inglés comma-separated values) son un tipo de documento para representar datos en forma de tabla, en las que las columnas se separan por comas o punto y coma y las filas por saltos de línea.

Es necesario que la información obtenida sea tratada antes de poder ser visualizada en la aplicación, puesto que a lo largo de una grabación puede haber fotogramas para los que Kinect no haya sido capaz de obtener información sobre la posición del paciente o sobre alguna articulación en concreto. Esto se puede deber a que el paciente no estaba situado dentro del rango de captura de la cámara o porque el sensor fallase en ese instante. Para que la posición del paciente sea conocida durante toda la grabación (siempre que el paciente se encuentre delante de la cámara y en rango) se utilizan unas funciones implementadas en el lenguaje *R* [4] por el grupo de investigación BDI. Estas funciones son capaces de inferir la posición de una articulación si Kinect ha fallado a la hora de detectarla.

En el fichero que se obtiene como resultado de aplicar las funciones, se indica para cada instante de tiempo (por cada segundo se registra la posición del paciente alrededor de 30 veces) a lo largo de la grabación, la posición X, Y y Z (esqX, esqY, esqZ) de cada una de las 25 articulaciones. Para cada una de las articulaciones también se indica si su posición ha sido o no inferida (esqInf).

Las columnas que siguen a las articulaciones, no tienen un formato predefinido. Por lo que se permite que se introduzca cualquier información que pueda resultar oportuna para el análisis de los pacientes. Ya sea información sobre el paciente o datos sobre la grabación.

La estructura del fichero es la siguiente:

		Para cada articulación				
tiempo	hora	esqInf	esqX	esqY	esqZ	Columnas no predefinidas

Tabla 1. Formato fichero grabación

Tal y como se puede observar en el esquema de la estructura, en la primera posición se encuentra el tiempo relativo a cada fila de la grabación con el formato HH:MM:SS, empezando por 0:00:00. La segunda posición contiene la hora de la grabación con el mismo formato. A continuación, se encuentra la información relativa a las veinticinco articulaciones, cada una de ellas se define por cuatro columnas; la primera para indicar si la articulación ha sido inferida, en caso de serlo tomará como valor 1, y 0 en caso contrario. Las siguientes tres columnas representan las coordenadas X, Y y Z de la articulación, vienen dadas en metros.

Dada una fila del fichero, es posible representar el esqueleto del paciente para ese instante concreto.

1.2. Gestión de pacientes y grabaciones

Actualmente, antes de realizar una sesión de grabación, los pacientes participantes rellenan un formulario con datos personales. Solamente se recoge información relevante para la investigación, datos como el nombre o el DNI no lo son. Posteriormente, a cada paciente se le asigna un identificador único. Esta información se almacena en un fichero sin encriptar y no se relaciona directamente con las grabaciones.

Por otra parte, los ficheros que contienen toda la información sobre las grabaciones se almacenan en un disco duro externo junto con sus correspondientes vídeos. Este disco duro se encuentra encriptado en cumplimiento de la LOPD. Los vídeos no están sincronizados con los ficheros de grabación, es decir, el segundo 3 del vídeo no coincide temporalmente con la línea del fichero CSV correspondiente a este segundo, el tiempo de cada fila del CSV lo representa la primera columna de la fila, tal y como se ha explicado en el apartado anterior.

Tanto la gestión de pacientes como de ficheros se realiza de forma local.

1.3. Aplicación

La aplicación actual es una aplicación de escritorio que permite visualizar las grabaciones y ofrece al usuario herramientas para controlar la reproducción. La visualización consiste en la reproducción sincronizada de la vista frontal y lateral del esqueleto extraído de Kinect con el vídeo de la grabación (en caso de incluirlo). Para visualizar una grabación el usuario debe seleccionar de forma manual el fichero CSV y el fichero de vídeo correspondientes a la grabación. De esta forma la aplicación reproducirá ambos archivos sincronizados. Es necesario que los ficheros de la grabación se encuentren almacenados localmente para poder llevar a cabo la reproducción.

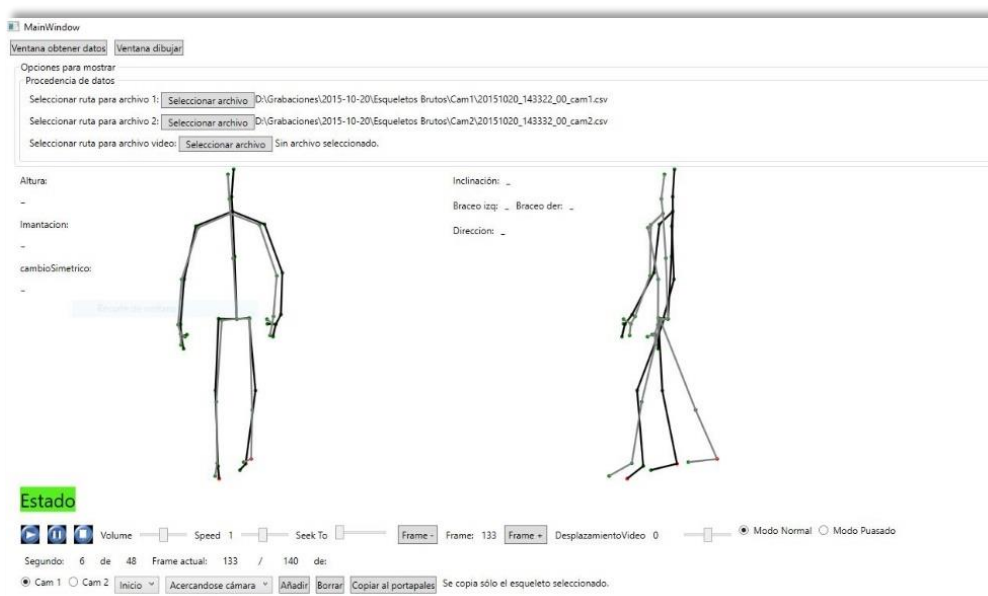


Ilustración 7. Aplicación anterior

2. Objetivos

En este capítulo se presentan los objetivos del proyecto. Por un lado, se exponen los objetivos definidos en la primera reunión. Por otro lado, se muestra el Product Backlog, formado por las Historias de Usuario finales, planteadas inicialmente en el análisis realizado junto al cliente durante las primeras reuniones.

2.1. Requisitos generales

En este apartado se exponen los requisitos generales y no funcionales del proyecto. Estos requisitos se han definido en la primera reunión del proyecto, celebrada el 19 de enero de 2018.

Este proyecto tiene como objetivo principal desarrollar una aplicación web para gestionar grabaciones de marcha de pacientes de Parkinson realizadas con Kinect. Para ello la aplicación permitirá dar de alta a pacientes y grabaciones. Los pacientes contarán con información personal, además de información relativa a la enfermedad. Por otro lado, las grabaciones incluirán el fichero CSV obtenido de Kinect y en caso de estar disponible, también el vídeo, así como información correspondiente a la sesión de grabación. Tanto los pacientes como las grabaciones se podrán modificar y borrar, asimismo las grabaciones se podrán asignar a los pacientes.

La aplicación web contará con una herramienta para visualizar las grabaciones; la visualización estará formada por el vídeo de la grabación y el esqueleto del paciente extraído de Kinect. El vídeo estará sincronizado con el esqueleto y el usuario podrá controlar su reproducción, pudiendo pausar, reproducir, avanzar y retroceder a su antojo. Además, podrá añadir notas al esqueleto que posteriormente se visualizarán junto a él.

Dentro de los objetivos secundarios del proyecto está la elección y aprendizaje de un framework con el que desarrollar la aplicación, además de una plataforma en la que alojar la aplicación web y almacenar toda la información.

Otro objetivo es la elección de un formato adecuado para la información generada por Kinect y por las funciones de preprocesamiento y extracción de características ya desarrolladas en R.

También será necesario definir mecanismos de confidencialidad de los datos, tanto en la comunicación entre cliente y servidor como en su posterior almacenamiento en la base de datos.

2.2. Pila de Producto

En base a los requisitos generales del proyecto y a posteriores reuniones, se define el Product Backlog del proyecto, compuesto por las Historias de Usuario en las que se detallan los requisitos. El Product Backlog forma parte de la metodología SCRUM, metodología con la que se llevará a cabo el proyecto y explicada en detalle en la sección [3.7. Metodología](#). El Product Backlog o Pila de Producto consiste en una lista donde se plasman los objetivos o requisitos del proyecto, representando la visión y expectativas del cliente respecto a los objetivos y entregas del producto. Además, en cada una de las Historias de Usuario se incluyen las pruebas necesarias para dar el visto bueno al requisito. Estas historias se van detallando a lo largo del proyecto.

A continuación, se presentan las versiones finales de dos de las historias de usuario. El resto de historias están accesibles en el [Anexo A. Historias de Usuario](#). Al principio del proyecto las historias no eran tan detalladas, han ido evolucionando a lo largo del proyecto hasta llegar al nivel de detalle actual.

Historia de usuario

Número: 2

Usuario: Cliente

Nombre historia: Entorno paciente

Descripción:

Como cliente quiero añadir pacientes para posteriormente ver información sobre ellos y gestionarlos.

Notas:

- Puedo añadir nuevos pacientes indicando un identificador, la fecha de nacimiento, el sexo, el año de inicio de la enfermedad, la clasificación (1, 2 o 3), el peso y la altura. Además, puedo añadir tantos atributos extra cómo quiera.
- Solo puedo ver los pacientes que formen parte de mi grupo.
- Los pacientes se muestran ordenados en una tabla.
- Puedo editar el paciente pulsando sobre cualquier dato en la tabla.
- Puedo ordenar los pacientes por cualquier campo.
- Puedo seleccionar uno o más pacientes para eliminarlos.
- Puedo filtrar los pacientes por su identificador.
- Puedo seleccionar el número de pacientes que se muestra por página.
- Puedo asignar al paciente una grabación cuando lo estoy registrando.
- Puedo ocultar columnas de la tabla.

Test:

- Añadir un nuevo paciente y comprobar que se ha añadido correctamente a la base de datos.
- Los pacientes pueden incluir tantos atributos extra como se crea conveniente.
- Asignar una grabación al paciente a la hora de registrarlo.
- Comprobar que los nuevos pacientes solamente son visibles para los usuarios que pertenecen al mismo grupo que el usuario que lo ha añadido.
- Eliminar un paciente y comprobar que se ha eliminado en la base de datos y no se muestra más en la tabla.
- Eliminar varios pacientes y comprobar que se han eliminado en la base de datos y no se muestran más en la tabla.
- Eliminar un paciente con grabación asociada y comprobar que la grabación queda sin asignar tras borrar el paciente.
- Editar todos los campos y comprobar que los cambios se ven reflejados en la base de datos.
- Filtrar los pacientes por el identificador.
- Reordenar la tabla por columnas.
- Ocultar columnas.

Historia de usuario

Número: 4

Usuario: Cliente

Nombre historia: Visualización gráfica de esqueletos

Descripción:

Como cliente quiero visualizar gráficamente los esqueletos a partir de un fichero para poder observar los movimientos del paciente junto con información relativa a los mismos.

Notas:

- Las articulaciones se mostrarán de distinto color si estas son inferidas o no.
- Se podrán añadir anotaciones al esqueleto durante la reproducción.
- Las anotaciones añadidas se mostrarán en futuras visualizaciones del esqueleto, estas se mostrarán en el punto sobre el que se ha pulsado para añadirla.

Test:

- Comprobar que el cliente puede visualizar el esqueleto para un fichero concreto.
- Comprobar que si la articulación es inferida se muestra de un color distinto.
- Se permite tomar anotaciones durante la reproducción pulsando en cualquier punto de la zona de visualización del esqueleto.
- Las anotaciones tomadas se visualizan durante la reproducción del esqueleto.

3. Planificación

En este capítulo se presenta la planificación del proyecto. En primer lugar, se expone el alcance del proyecto junto con los principales objetivos y exclusiones. En segundo lugar, se explica el diagrama EDT del proyecto, en él se descompone jerárquicamente el trabajo a desarrollar. En tercer lugar, se presenta el diagrama de Gantt, para exponer el tiempo de dedicación previsto a las diferentes tareas a lo largo de las iteraciones. En cuarto lugar, se analizan los riesgos que pueden aparecer durante el desarrollo del proyecto y sus correspondientes planes de contención. En último lugar, se presenta la metodología con la que se va a llevar a cabo el desarrollo y se define el equipo del proyecto y sus roles.

3.1. Alcance

El objeto del proyecto es obtener como producto una aplicación web accesible desde Internet donde gestionar grabaciones de marcha de pacientes de Parkinson realizadas con Kinect.

3.2. Objetivos

Los principales objetivos del proyecto están definidos en el capítulo [2. Objetivos](#), en él se definen los requisitos generales de la aplicación y la pila de producto obtenida como resultado del análisis de los requisitos del cliente.

3.3. Exclusiones

Por el coste adicional que supone su desarrollo, se excluyen del proyecto las siguientes funcionalidades o características de la aplicación:

- Gestión de las claves de cifrado usando un gestor de claves en la nube. El estudio de las distintas opciones para gestionar las claves está disponible en el capítulo [6.3.2.3 Gestión de las claves de cifrado](#).
- Recuperación de la clave de cifrado. En caso de que se olvide la clave de cifrado de un grupo, ofrecer la posibilidad de recuperarla. De esta forma se evitaría la pérdida de toda la información de un grupo.
- Traducción de la aplicación a varios idiomas.
- Asegurar la correcta visualización de la aplicación web en dispositivos móviles y tabletas.
- Permitir escoger la velocidad de reproducción del visualizador. Pudiendo elegir entre acelerar o ralentizar la reproducción.

3.4. EDT

La estructura de descomposición del trabajo (EDT) es una herramienta que consiste en la descomposición jerárquica del proyecto para cumplir con los objetivos del mismo. El EDT del proyecto a desarrollar se divide en cinco secciones en las que se agrupan los paquetes de trabajo:

1. **Gestión:** Esta sección agrupa los paquetes de trabajo relacionados con la gestión del proyecto, como son, la elección de las tecnologías, la planificación del proyecto, el seguimiento y control y las reuniones de seguimiento.
2. **Gestión del conocimiento:** Esta sección contiene el paquete de trabajo correspondiente a la adquisición de las competencias necesarias para elaborar la aplicación web.
3. **Desarrollo:** Esta sección contiene los paquetes de trabajo correspondientes a cada uno de los Sprints del proyecto y las pruebas correspondientes.
4. **Documentación:** Esta sección contiene los paquetes de trabajo relacionados con la documentación del proyecto, por un lado, la memoria del proyecto y por el otro, el manual de usuario.
5. **Defensa del proyecto:** Engloba los paquetes de trabajo tanto de la preparación de la exposición como de la propia defensa.

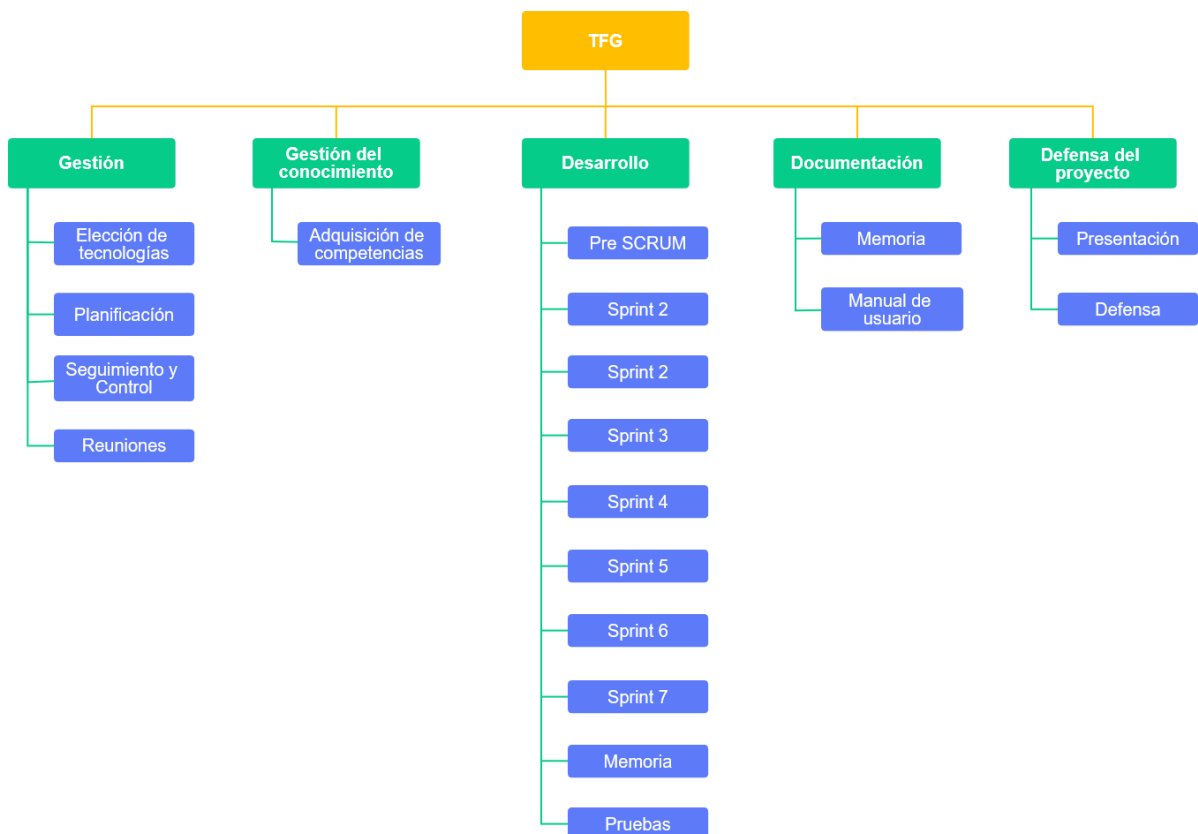


Ilustración 8. EDT del proyecto

3.5. Diagrama de Gantt

En este apartado se presenta el diagrama de Gantt inicial. En él se puede observar la división del tiempo del proyecto en iteraciones. Para cada iteración se han establecido 3 tareas comunes; la planificación de la misma, el desarrollo y la reunión de cierre, en la que se revisará lo desarrollado durante el sprint junto con el cliente.

A continuación, se muestra el diagrama de Gantt, con él se puede obtener una visión global de la organización del proyecto. El diagrama se ha dividido en dos imágenes para facilitar su visionado.

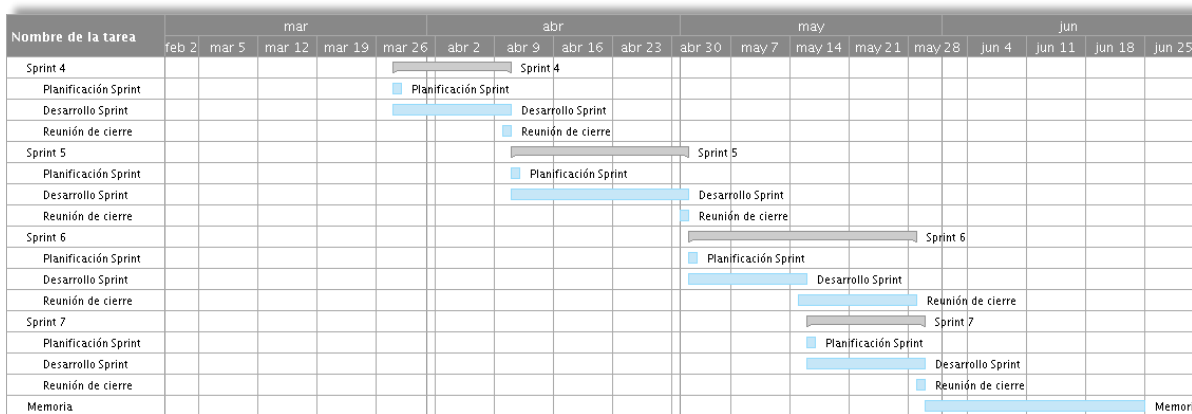
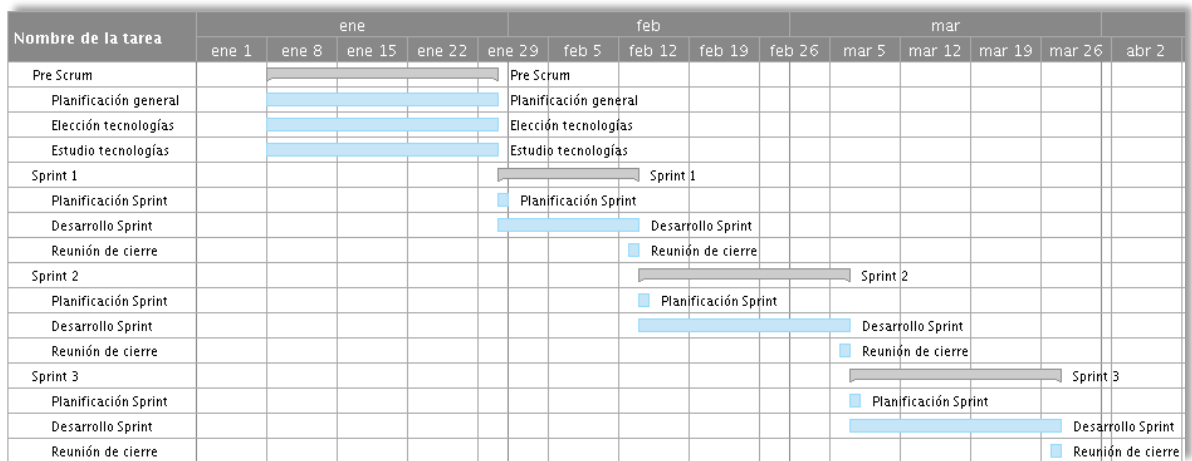


Ilustración 9. Diagrama de Gantt

3.6. Riesgos

A lo largo del desarrollo del proyecto existen factores de riesgo a tener en cuenta que pueden resultar en aplazamientos del desarrollo. A continuación, se enumeran los principales riesgos detectados y se recogen en una tabla, mostrando el impacto que ocasionarían sobre el proyecto y la probabilidad de que ocurran. Por cada uno de ellos se describe de forma breve su correspondiente plan de mitigación.

1. Problemas personales o de salud pueden resultar en una pausa temporal del proyecto.
2. Disponibilidad del director y del cliente para la celebración de reuniones.
3. Pérdidas de la información almacenada localmente pueden hacer que haya que desarrollar nuevamente funcionalidades de la aplicación o reescribir apartados de la memoria.
4. Fallo en la máquina de trabajo.
5. Fallo en los servidores de Firebase.
6. Asignaturas de cuarto curso. Suspender alguna de ellas disminuiría el tiempo disponible para dedicar al proyecto.
7. Problemas con las tecnologías utilizadas.

Probabilidad	Impacto				
	Muy bajo	Bajo	Medio	Alto	Muy alto
Muy alta					
Alta					
Media				(7)	
Baja			(4, 6)	(2)	(3)
Muy baja				(1)	(5)

Tabla 2. Riesgos del proyecto

Plan de mitigación para cada uno de los riesgos:

1. Se planificará el trabajo con márgenes más extensos de los que se estima que harían falta en caso de no haber ningún retraso.
2. En cada reunión que se celebre se acordará la fecha de la próxima reunión.
3. El proyecto contará con control de versiones en GitHub, además tanto los archivos relativos a la aplicación web como los documentos de gestión y documentación estarán sincronizados en la nube con Google Drive. De esta forma es posible recuperar la información en caso de que se produzca una pérdida en local, pudiendo acceder a ella desde cualquier dispositivo con conexión a Internet.
4. Se contará con una segunda máquina de trabajo que reemplazaría a la principal en caso de que esta sufriese algún fallo. La información estará a salvo, ya que se contará con una copia de seguridad en la nube, siendo esta parte del plan de mitigación del riesgo (3).
5. Se haría uso de las copias de seguridad existentes, alojando la aplicación web en otro Hosting (como A2Hosting) y migrando la base de datos a otra plataforma (como Pusher) de forma temporal.
6. Se planificará el trabajo con márgenes más extensos de los que se estima que harían falta en caso de no haber ningún retraso, además una vez terminados los exámenes la dedicación al proyecto será total.
7. Se estudiarán siempre tecnologías alternativas. En el caso de que alguna tecnología diese problemas durante el desarrollo se recurriría a alguna de las alternativas.

3.7. Metodología

Se ha decidido seguir la metodología de desarrollo Scrum [5] para el desarrollo del proyecto. Scrum es una metodología ágil para el desarrollo de software o la gestión de proyectos. Antes de dar la definición formal de la metodología será necesario definir en qué consiste una metodología ágil. El desarrollo de software ágil se define como:

El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así, el trabajo es realizado mediante la colaboración de equipos autoorganizados y multidisciplinarios, inmersos en un proceso compartido de toma de decisiones a corto plazo.

El desarrollo de un proyecto se divide en iteraciones, donde cada iteración del ciclo de vida incluye: planificación, análisis de requisitos, diseño, codificación, pruebas y documentación. Teniendo gran importancia el concepto de "Finalizado" (Done), ya que el objetivo de cada iteración no es agregar toda la funcionalidad para justificar el lanzamiento del producto al mercado, sino incrementar el valor por medio de "software que funciona" (sin errores).

Por otro lado, Scrum es un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible, productiva y creativamente. Al inicio del proyecto se define el Product Backlog (pila de producto) del mismo, este consiste en una lista ordenada donde se detallan los requisitos del producto, y es la única fuente de requisitos para cualquier cambio a realizarse en el producto. La pila irá evolucionando a lo largo del proyecto. El Product Owner es su responsable. Scrum utiliza el progreso real de un proyecto para planificar. Los proyectos se dividen en periodos de trabajo breves, conocidos como sprints (iteraciones). Cada uno de ellos tiene una duración aproximada de alrededor de tres semanas. Al final de cada sprint, el cliente y los miembros del equipo se reúnen para evaluar el trabajo realizado a lo largo del sprint y planificar los siguientes pasos a seguir. Esto permite que la dirección del proyecto se ajuste o se reoriente una vez finalizado el trabajo, sin especulaciones ni predicciones.

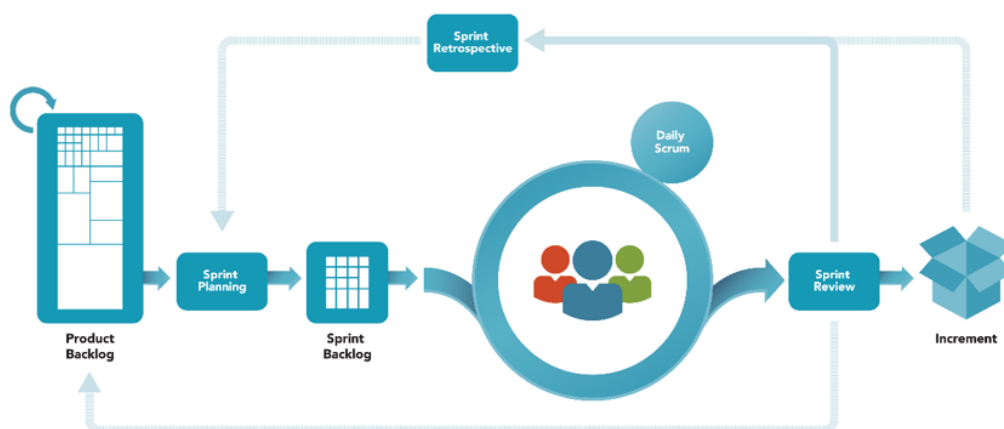


Ilustración 10. Scrum, extraído de [6]

3.7.1. Equipo del proyecto

Los miembros del equipo de un proyecto Scrum tienen tres papeles fundamentales: Product Owner (propietario del producto), Scrum Master y Team Member (equipo de desarrollo).

- **Product Owner:** Se encarga de comunicar la visión del producto al equipo de desarrollo y determinar los requisitos y prioridades del desarrollo, reflejándolo en el Product Backlog. Es el papel con más autoridad y en consecuencia el de más responsabilidad.
- **Scrum Master:** Actúa como enlace entre el propietario del proyecto y el equipo. El Scrum Master no dirige al equipo. Se encarga de gestionar y facilitar la ejecución de las reglas Scrum, también aconseja al Product Owner sobre cómo maximizar el ROI (Return Of Investment) para el equipo.
- **Team Member:** Es el responsable de construir el producto. En cada sprint, el equipo es responsable de determinar cómo va a lograr acabar el trabajo. Esto garantiza al equipo un grado de autonomía, pero, al igual que con el Product Owner, esta libertad viene acompañada por la responsabilidad de cumplir los objetivos del Sprint.

El equipo está compuesto por un total de 3 personas, y teniendo en cuenta los roles que establece la metodología Scrum, el reparto de responsabilidades se ha llevado a cabo de la siguiente manera:

- Desarrollador: Asier Tamayo.
- Scrum Master: Urko Lopez de Abetxuko.
- Propietario del producto: Alfredo Goñi.

La pila de producto del proyecto se puede consultar en el capítulo [2.2 Pila de Producto](#).

3.7.2. Iteraciones

La duración del proyecto rondará las 300 horas y abarcará seis meses aproximadamente, entre enero y junio. Lo que resulta en una planificación que contempla un total de 7 Sprints (alrededor de dos/tres semanas por sprint). La duración de cada una de las iteraciones se ha establecido en función de la carga de trabajo prevista para las asignaturas del cuatrimestre. En la tabla también se refleja el periodo del proyecto previo a adoptar la metodología Scrum, denominado en la tabla como *pre Scrum*. La organización temporal se puede ver con más detalle en el [diagrama de Gantt](#).

Número Iteración	Fecha inicio	Fecha finalización
Pre Scrum	08/01/2018	30/01/2018
Sprint 1	31/01/2018	14/02/2018
Sprint 2	15/02/2018	28/03/2018
Sprint 3	29/03/2018	10/04/2018
Sprint 4	11/04/2018	17/04/2018
Sprint 5	18/04/2018	01/05/2018
Sprint 6	02/05/2018	15/05/2018
Sprint 7	16/05/2018	29/05/2018
Memoria	30/05/2018	22/06/2018

Tabla 3. Sprints del proyecto

4. Tecnologías

Este capítulo se divide en dos partes. Por un lado, se describe la elección de las principales tecnologías utilizadas en el proyecto, y por el otro, se presentan todas las tecnologías usadas en el proyecto, tanto en el desarrollo de la aplicación, como en la gestión del proyecto.

4.1. Elección de tecnologías

La primera elección es la tecnología con la que se desarrollará la aplicación web y que será la principal tecnología del lado cliente. Como posibles herramientas de desarrollo pareció interesante considerar tanto el framework AngularJS [7] como la biblioteca ReactJS [8], ya que las dos permiten desarrollar aplicaciones web de una sola página. Ambas tecnologías son usadas por los investigadores del grupo, por lo que, en caso de dudas o dificultades durante el desarrollo, sus conocimientos serán de gran ayuda. Para elegir entre las dos alternativas se realiza la siguiente tabla comparativa:

Tecnología	ReactJS	AngularJS
Autor	Comunidad de Facebook	Google
Tipo	Librería JavaScript	Framework MVC
Lenguaje	JSX	JavaScript, HTML
Curva de aprendizaje	Baja	Alta
Renderización	Servidor	Cliente
Arquitectura de la App	Ninguna	MVC
Data Binding	Unidireccional	Bidireccional
Routing	No tiene soporte directo	Si
Última versión	16.4.0 (23/05/2018)	1.6.8 (18/12/2017)

Tabla 4. Comparativa ReactJS AngularJS

Tras analizar las características de ambas tecnologías se escoge AngularJS. El principal motivo es su arquitectura, mientras que React únicamente se encarga de la vista de la aplicación, AngularJS ofrece una arquitectura MVC completa.

El Data Binding, tecnología que mapea las propiedades de la instancia de un objeto de datos con su localización en una fuente de datos, también ha sido decisivo a la hora de escoger la tecnología. Mientras que en React es unidireccional, es decir, desde el HTML no es posible cambiar el valor de una variable del controlador, pues no están directamente asociadas, en AngularJS es bidireccional, lo que permite que las actualizaciones sobre el modelo se reflejan automáticamente en la vista y viceversa. Especialmente útil en la aplicación a desarrollar, pues la interacción del usuario con la información de la aplicación web será constante.

Otro motivo para su elección ha sido la cantidad de documentación que hay disponible en la red y la enorme comunidad que hay detrás.

También ha tenido especial relevancia en la elección, la existencia de AngularJS Material [9], un framework de componentes de la interfaz de usuario, que contiene cantidad de componentes que facilitan enormemente el desarrollo de la aplicación siguiendo la estética Material Design de Google. React también cuenta con la librería Materialize [10] para el mismo propósito, pero esta librería es menos completa y ofrece menos facilidades que AngularJS Material.



Ilustración 11. Logo AngularJS

La segunda elección es la tecnología que se encargará de gestionar el lado servidor. El objetivo es encontrar una plataforma capaz de alojar la aplicación web, de albergar los datos de los pacientes y grabaciones y sus ficheros CSV y vídeos, además, toda la información debe estar almacenada de forma segura. También es necesario que la plataforma permita gestionar usuarios. Todo ello sin tener que escribir una línea de código en el lado servidor. Además, esta tecnología debe ser totalmente compatible con AngularJS, la tecnología escogida para el lado cliente. A continuación, se presenta una tabla comparativa entre distintas plataformas en la nube que ofrecen productos similares:

Plataforma	Firebase	Pusher	PubNub
Autor	Google	Pusher	PubNub
Precio	Versión gratuita con limitaciones	Versión gratuita con limitaciones	De pago, prueba gratuita
Seguridad	SSL en comunicación cliente/servidor y encriptación en servidores	SSL en comunicación cliente/servidor. No se encuentra información sobre encriptado en servidores	SSL en comunicación cliente/servidor y encriptación en servidores
Base de datos	NoSQL en tiempo real, almacenamiento de archivos	NoSQL en tiempo real	NoSQL en tiempo real
Gestión de usuarios	Si	No	Parcial
Hosting	Si	No	No
Compatibilidad con AngularJS	Total	Total	Total

Tabla 5. Comparativa Firebase, Pusher y PubNub

Tras estudiar las características de las alternativas presentadas en la tabla, se escoge Firebase como plataforma del lado servidor. Los principales motivos para la elección han sido que cuenta con una versión gratuita muy completa detallada en el capítulo [4.2.2. Lado Servidor](#). Por su compatibilidad total con AngularJS (ambas tecnologías han sido desarrolladas por Google), por el nivel de seguridad que proporciona tanto en las comunicaciones entre cliente y servidor como en sus servidores, por ofrecer la posibilidad de alojar la aplicación web de forma gratuita y contar con almacenamiento de datos en tiempo real y almacenamiento específico para ficheros. Permite también gestionar las cuentas de los usuarios de la aplicación, ofreciendo funciones ya implementadas para crear nuevos usuarios y recuperar las contraseñas de los mismos. Además, cuenta con una gran base de usuarios y una documentación disponible en español donde se detallan todas las características.

En resumen, se escoge AngularJS como tecnología con la que desarrollar el lado cliente de la aplicación web y Firebase como plataforma para alojarla y gestionar los usuarios y toda la información de la aplicación.

4.2. Tecnologías para el desarrollo de la aplicación

A continuación, se presentan las tecnologías usadas en el desarrollo de la aplicación. Estas se dividen en tecnologías del lado cliente y tecnologías del lado servidor.

4.2.1. Lado Cliente.

AngularJS

AngularJS, es una tecnología del lado cliente, un framework Modelo Vista Controlador JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página (SPA). El paradigma de programación de AngularJS también es conocido como Modelo Vista Cualquier cosa (Model-View-Whatever) [11]. Donde cualquier cosa representa “lo que sea que funcione para ti”. Este modelo permite una separación de responsabilidades y un mantenimiento de la aplicación muy sencillo.

Angular permite construir aplicaciones web modernas e interactivas mediante el aumento del nivel de abstracción entre el desarrollador y las tareas más comunes del desarrollo web.

Entre el contexto de Angular y el controlador se sitúa algo llamado “two way data Binding”, que permite que las actualizaciones sobre el modelo se reflejen automáticamente en la vista (típicamente documentos HTML) y viceversa, lo que facilita la interacción entre el modelo y la vista. El desarrollador debe insertar en el HTML la anotación correspondiente de Angular en la posición de la vista en la que quiera mostrar el dato del modelo y de la que quiera obtener información para almacenar en el controlador. De este modo, tanto si es el usuario el que modifica ese dato (por ejemplo, el dato a introducir en la caja de texto de un formulario) o si se modifica desde el controlador, automáticamente dicho cambio se reflejará en el otro lado.

Esto se realiza gracias al \$scope de Angular. El scope es un objeto que hace referencia al modelo de la aplicación y es capaz de almacenar tanto variables como funciones en el contexto de un controlador asociado a una vista.

AngularJS también extiende el tradicional HTML con etiquetas propias (directivas). Para interpretar las etiquetas propias la biblioteca lee el HTML, que contiene atributos adicionales en las etiquetas, entonces Angular interpreta esos atributos como directivas que vinculan las partes de entrada o salida de la página a un modelo que está representado por variables de JavaScript, ya sea configuradas manualmente, o recuperadas de recursos JSON estáticos o dinámicos. De esta forma interactúa el modelo con la vista y viceversa. Por ejemplo, para asociar una variable del controlador con un input de la vista, bastará con incluir el atributo ng-model e indicar el nombre de la variable del scope en la que se quiere almacenar su contenido. También existen otras muchas características, como filtros, servicios, routing entre vistas, testing, ...

Además, AngularJS es compatible con todos los navegadores de última generación (Chrome, Firefox, Safari, Opera, Webkits, Edge).

AngularJS Material

AngularJS Material [9] es un framework de componentes de UI (Interfaz de Usuario). Estos componentes funcionan como etiquetas y atributos de HTML. El framework proporciona un conjunto de componentes UI reutilizables basados en Material Design. Implementa entre otras muchas cosas, formularios con control de errores, notificaciones, ventanas emergentes, buscadores, menús, estructuración del contenido de la página... Por lo que esta librería facilita enormemente el trabajo de diseño de la aplicación web.

Papa Parse

Papa Parse [12] es un parser de CSV (o texto delimitado) para JavaScript. Es fiable y correcto de acuerdo con RFC 4180, y permite convertir ficheros CSV (almacenados localmente o en la nube) en vectores de datos, detectando automáticamente el delimitador de datos. La conversión de los ficheros CSV a vectores facilita enormemente la manipulación y el manejo de los datos.

CryptoJS

CryptoJS [13] es una librería desarrollada por Google que consta de una colección de algoritmos criptográficos estandarizados y seguros implementados en JavaScript. Son rápidos y tienen una interfaz consistente y simple. Se ha empleado para encriptar los datos y los ficheros.

Moment.js

Moment.js [14] es una librería para gestionar las fechas en JavaScript, para ello implementa un sistema de manejo de fechas mucho más cómodo y simple. Permite, entre otras cosas, formatear las fechas en función de la localización (idioma/cultura) o realizar operaciones con fechas, como calcular el tiempo transcurrido entre dos fechas.

Md-data-table

Md-data-table [15] implementa tablas de datos de Material Design en AngularJS. Esta librería es requerida para solventar la falta de tablas en el framework AngularJS Material. Las tablas de datos se utilizan para presentar conjuntos de datos en bruto, siendo particularmente útiles para visualizar y manipular grandes conjuntos de datos, siendo este el caso del proyecto. Además, la librería facilita la interacción y manipulación de los datos por parte del usuario permitiendo ordenar las columnas, seleccionar filas, realizar búsquedas...

Default-passive-events

Esta librería habilita los detectores pasivos de eventos de forma predeterminada para algunos eventos. Básicamente establece la propiedad `{passive: true}` a los nuevos detectores de eventos automáticamente. De esta forma se evita que los navegadores muestren el warning “[Violation] Added non-passive event listener to...”.

4.2.2. Lado Servidor.

Firebase

Firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles desarrollada por James Tamplin y Andrew Lee en 2011 y adquirida por Google en 2014.



Ilustración 12. Logo de Firebase

Esta plataforma ofrece una gran cantidad de servicios para el desarrollo de aplicaciones. Los principales servicios se describen a continuación:

- **Realtime Database:** proporciona una base de datos NoSQL alojada en la nube que permite a través de una API almacenar y sincronizar datos entre los usuarios en tiempo real. Los datos se almacenan en formato JSON.
- **Firestore:** servicio para almacenar y procesar con rapidez y facilidad el contenido que generan los usuarios, como fotos o vídeos.
- **Hosting:** permite implementar una aplicación web de una sola página, una página de destino de aplicación web o una app web progresiva. Además, provee a los dominios de certificado SSL, asegurando una total seguridad en la comunicación entre cliente y servidor.
- **Authentication:** autenticación de usuarios utilizando únicamente código del lado del cliente. Permite la autenticación de usuarios mediante email y contraseña, que se almacenará en Firebase, y la autenticación con las plataformas Facebook, Twitter, Google y Github.
- **Google Analytics:** proporciona una visión detallada sobre el uso de la aplicación por parte de los usuarios.
- **Cloud Messaging:** plataforma para enviar mensajes y notificaciones a los usuarios. Está disponible para las plataformas Android, IOS y aplicaciones web. Permite enviar correos electrónicos, notificaciones de escritorio y SMS.

Firebase cuenta tanto con versión gratuita como de pago. Todos los servicios anteriormente mencionados están disponibles en la versión gratuita, pero con limitaciones. Las limitaciones establecen un máximo de conexiones simultáneas, un ancho de banda limitado, capacidad de almacenamiento reducida, una base de datos... A continuación, se muestra una tabla con las características de cada uno de los planes disponibles y sus limitaciones:

Productos	Plan Spark Límites generosos para aficionados Sin cargo	Plan Flame Precios predecibles para apps en expansión USD 25/mes	Plan Blaze Calcula los precios de las apps a gran escala Pago por uso ✓ Free usage from Spark plan included*
Incluido sin cargo Authentication (except Phone Auth), Analytics, Predictions, App Indexing, Dynamic Links, Invites, Remote Config, Cloud Messaging (FCM), Performance Monitoring, and Crashlytics.	✓ Incluidos	✓ Incluidos	✓ Incluidos
Realtime Database Conexiones simultáneas [?] GB almacenados GB descargados Multiple databases per project	100 1GB 10 GB/mes ✗	100k 2.5 GB 20 GB/mes ✗	100k/database \$5/GB USD 1/GB ✓
Hosting GB almacenados GB transferred Dominio personalizado y SSL	1 GB 10 GB/mes ✓	10 GB 50GB/month ✓	USD 0.026/GB USD 0.15/GB ✓
Storage [?] GB almacenados GB descargados Operaciones de carga Operaciones de descarga	5 GB 1 GB/día 20,000/día 50,000/día	50 GB 50 GB/día 100K/day 250,000/día	USD 0.026/GB USD 0.12/GB \$0.05/10k \$0.004/10k

Ilustración 13. Planes de Firebase

Puesto que los servicios ofrecidos en el plan gratuito cumplen con los requisitos iniciales del proyecto, se opta por su uso. Las limitaciones presentes en este plan para los servicios no afectan a la aplicación web del proyecto, al menos no durante su desarrollo ni una vez puesta en producción, a menos que una vez publicada el número de usuarios sea elevado y por lo tanto la cantidad de información a almacenar y a transferir crezca significativamente. En esa situación, sería necesario optar por un plan de pago. Entre los dos planes de pago disponibles, el plan Blaze parece el más conveniente, ya que únicamente se paga por lo que se usa. En conclusión, durante el desarrollo y en producción se usará el plan básico gratuito, y en caso de necesitar ampliar los servicios se optaría por el plan Blaze.

En este proyecto se usan los servicios de Hosting, para publicar la página web, Realtime Database, para almacenar información relativa a los pacientes y las grabaciones, Firebase Storage, para almacenar los ficheros CSV de las grabaciones y los ficheros de vídeo, y Firebase Auth para la gestión y autenticación de los usuarios. También se usa Firebase Cloud Messaging, cuando un usuario ha olvidado la contraseña se le envía un correo electrónico para recuperar la contraseña de inicio de sesión. Gracias a Firebase se evita tener que desarrollar y gestionar la parte del servidor, delegando en los servicios ofrecidos por Google.

4.3. Gestión del proyecto.

A continuación, se presentan las herramientas empleadas en la gestión del proyecto:

Microsoft Word

Programa de ofimática orientado al procesamiento de textos utilizado para el desarrollo de la memoria y el manual de usuario.

Microsoft Excel

Programa de ofimática de hojas de cálculo utilizado para la gestión del tiempo de trabajo.

Google Drive

Google Drive es un servicio de alojamiento de archivos multiplataforma en la nube. Este servicio permite almacenar y sincronizar archivos en línea y entre dispositivos (ordenadores, móviles, tabletas). Además, permite compartir contenido entre usuarios. Este servicio se ha empleado en el proyecto para respaldar todos los archivos del proyecto, así como la documentación relativa a la gestión del proyecto, la memoria y el manual de usuario.

WebStorm

WebStorm es un IDE desarrollado por JetBrains que ofrece asistencia de codificación para JavaScript, HTML y CSS y una amplia gama de tecnologías web modernas, entre ellas AngularJS. WebStorm está perfectamente equipado para el desarrollo complejo del lado del cliente y el desarrollo del lado del servidor con Node.js. Además, permite, simplemente pulsando un botón, ejecutar el proyecto de forma local, agilizando enormemente el testeo de la aplicación, ya que no es necesario ejecutar manualmente un servidor local ni subir la aplicación al hosting cada vez que se realiza un cambio y se quiere comprobar su correcto funcionamiento.

GitHub

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Aunque habitualmente el código de los proyectos está disponible públicamente, es posible almacenar el repositorio en privado con una cuenta de pago. En el proyecto se ha usado GitHub tanto como copia de seguridad del proyecto, como herramienta para gestionar las versiones del proyecto.

Draw.io

Draw.io es una aplicación para navegador de código abierto para crear diagramas. Se ha empleado para hacer el diagrama de casos de uso, el modelo de datos, y algún diagrama extra, por ejemplo, el diagrama para ilustrar la arquitectura de la aplicación.

5. Requisitos funcionales

En este capítulo se presentan los requisitos funcionales de la aplicación. Estos requisitos se ilustran mediante un diagrama de casos de uso, y el flujo de eventos correspondiente a dos de los casos de uso. El resto de flujo de eventos están disponibles en el [Anexo B. Flujos de eventos.](#)

5.1. Diagrama de casos de uso

A continuación, se muestra el diagrama de casos de uso de la aplicación. El diagrama se ha diseñado tras analizar los requisitos y objetivos de la aplicación a desarrollar. En él se distinguen dos actores, el usuario que accede a la aplicación sin haber iniciado sesión, que únicamente puede iniciar sesión o restaurar su contraseña y el usuario con sesión iniciada, que sería el rol que se consigue una vez ejecutado con éxito el caso de uso *Iniciar sesión*.

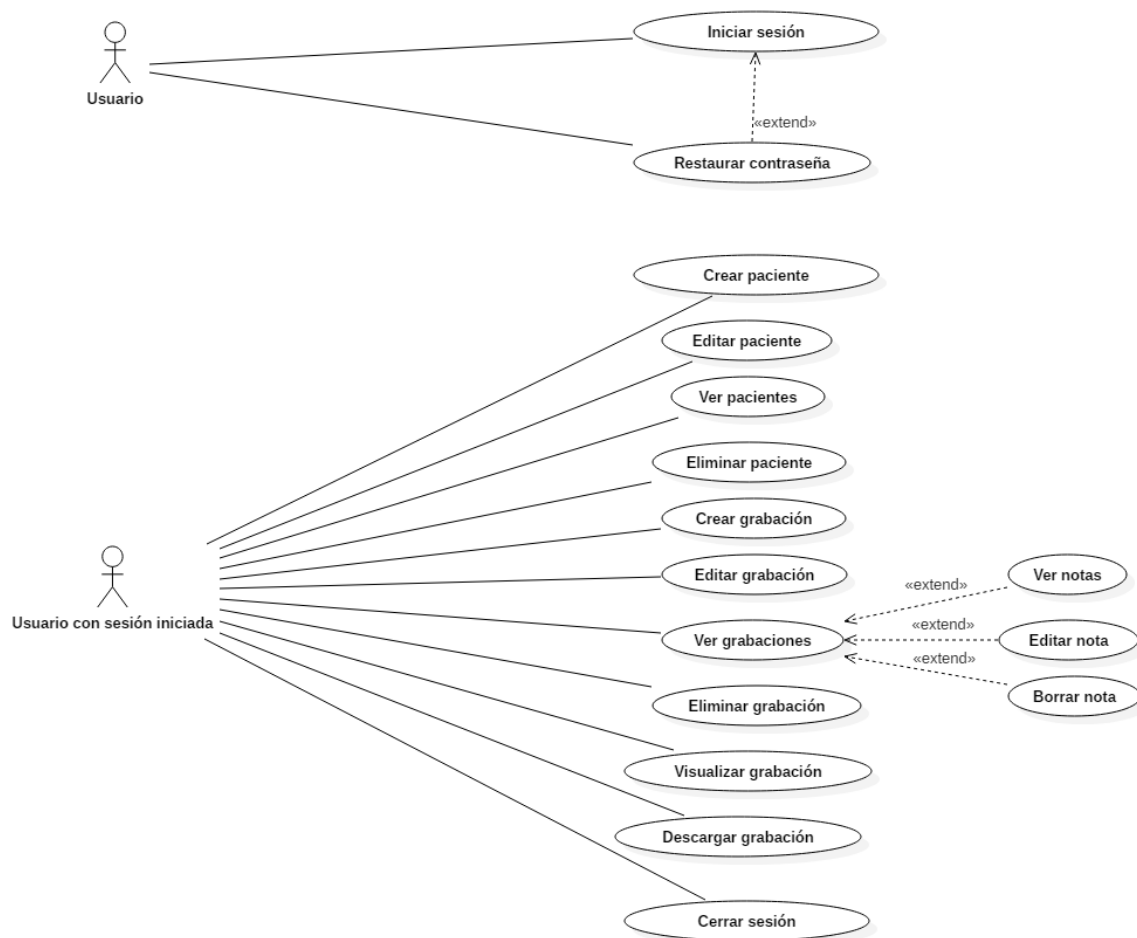


Ilustración 14. Diagrama de casos de uso

5.2. Flujo de eventos

En esta sección se presentan dos de los diagramas de flujo de eventos correspondientes a los casos de uso mostrados en el diagrama del apartado anterior. El resto de flujos de eventos están en el [Anexo B. Flujos de eventos](#).

5.2.1. Recuperar contraseña

Descripción: Permite al usuario recuperar la contraseña.

Actores: Usuario.

Precondiciones: Disponer de un usuario y contraseña para acceder a la aplicación.

Flujo de eventos:

1. El usuario pulsa sobre “¿Has olvidado la contraseña?” en el formulario de inicio de sesión.
2. El usuario introduce su nombre de usuario.
 - 2.1 Si el nombre de usuario no tiene un formato de correo válido el sistema muestra un mensaje de error.
3. El usuario pulsa sobre el botón “Restaurar”.
4. El sistema envía un correo electrónico con un enlace para crear una contraseña nueva.
 - 4.1 Si el nombre de usuario no está registrado, el sistema informa al usuario de que no se ha podido enviar el correo de recuperación. Continúa en el paso 1.
5. El usuario accede al correo electrónico y pulsa en el enlace.
6. El usuario introduce la nueva contraseña.
7. El sistema actualiza la contraseña del usuario.
 - 7.1 Si ha habido algún error el sistema informa al usuario. Continúa en el paso 3.

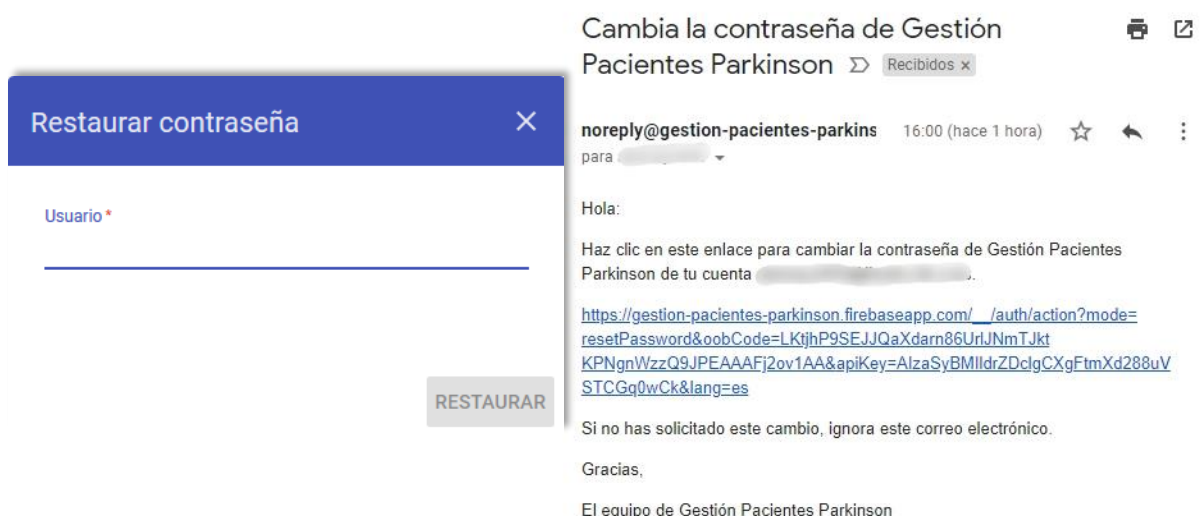


Ilustración 15. Recuperar contraseña

- 2.1.2.1.6.1. Si hay más de una página de notas:
 - 2.1.2.1.6.1.1. El sistema mostrará la página de notas siguiente a la actual.
- 2.1.2.1.7. Si el usuario pulsa en la flecha de navegación derecha de la parte inferior derecha de la tabla:
 - 2.1.2.1.7.1. Si hay más de una página de notas:
 - 2.1.2.1.7.1.1. El sistema mostrará la última página de notas.



Gestión notas		
<input type="checkbox"/>	Minuto ↑	Texto
<input type="checkbox"/>	00:00:03.9330869	Desviación derecha
<input type="checkbox"/>	00:00:30.8669765	Imantado

Página: 1 Filas por página: 10 1 - 2 de 2

Ilustración 16. Ver notas

6. Diseño

Este capítulo está dedicado al diseño de la aplicación. Se analizará la arquitectura, basada en el patrón MVC, el diseño del modelo de datos y el diseño de la seguridad de la aplicación. Además, se incluyen los diagramas de secuencia más representativos de la aplicación.

6.1. Arquitectura

La arquitectura de la aplicación web desarrollada sigue el patrón MVC. Este patrón consiste en separar los datos y la lógica de negocio de la interfaz de usuario. Para ello el patrón MVC propone la construcción de tres componentes distintos que son el Modelo, la Vista y el Controlador. A continuación, se explica de forma más detallada cada uno de los componentes.

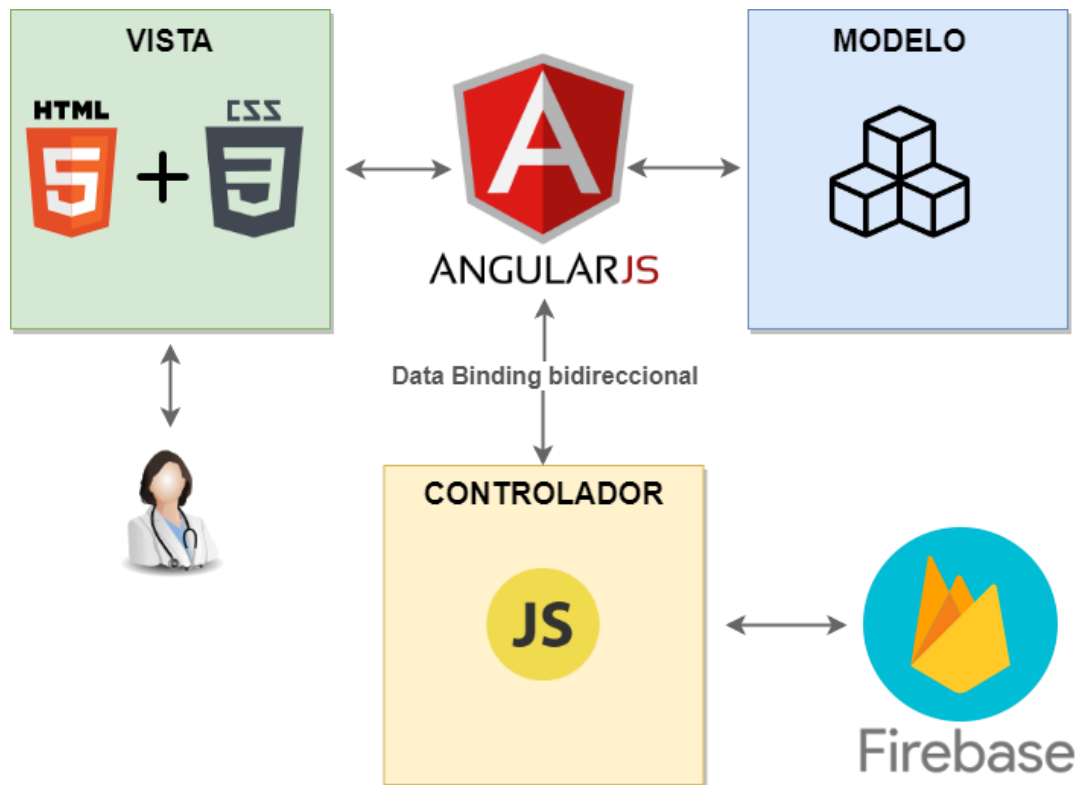


Ilustración 17. Arquitectura del proyecto

6.1.1. Modelo

El modelo es la representación de la información con la cual el sistema opera. El modelo de datos se obtiene directamente de Firebase y tal como se puede ver en el esquema de la arquitectura, es el controlador el encargado de la comunicación.

6.1.2. Vista

La vista está formada por las interfaces de usuario; en ellas se presenta el modelo en un formato adecuado para que el usuario interactúe con él. En la aplicación web la vista está desarrollada en HTML5 y CSS3, además de por las directivas de AngularJS (etiquetas HTML personalizadas). A través de las vistas, el usuario puede llevar a cabo las funcionalidades disponibles en la aplicación; crear pacientes, gestionarlos, crear grabaciones, gestionarlas, visualizar las grabaciones...

Otra característica esencial de AngularJS y más concretamente de las vistas, es el Data Binding; es decir, la forma en la que el modelo y la vista comparten información. Si el objetivo es mostrar información del modelo en la vista, bastará con incluir en la vista el nombre de la variable que se quiera visualizar entre llaves dobles. Por ejemplo, si se quiere visualizar el identificador del paciente será necesario añadir `{{ paciente.id }}` en el punto de la vista en el que se quiera incluir. Si el objetivo es que el modelo reciba información de la vista, por ejemplo, en un formulario, bastará con incluir en la etiqueta HTML el atributo `ng-model` con el nombre de la variable del modelo en la que se quiera guardar esa información. Por ejemplo, a la hora de crear una grabación, en el input en el que se introduce el lugar en el que se ha llevado a cabo, bastará con incluir el atributo `ng-model` indicando que el texto que se introduzca en ese input se corresponde con la variable `grabacion.lugar` del modelo.

```
<input name="lugar" ng-model="grabacion.lugar">
```

6.1.3. Controlador

El controlador es el componente más complejo de la aplicación y el que más tareas realiza. Este se encarga de responder a eventos, acciones del usuario, e invoca peticiones al modelo y a la vista. En AngularJS se corresponden con ficheros JavaScript. En la aplicación del proyecto los controladores también se encargan de llevar a cabo la encriptación de la información y toda la comunicación con Firebase, enviando y recibiendo los datos de usuarios, pacientes y grabaciones. Los controladores hacen de puente entre las vistas y los modelos.

6.2. Modelo de datos

Toda la información que utiliza la aplicación web se encuentra almacenada en Firebase. Por un lado, se encuentran los datos de inicio de sesión de los usuarios, estos se manejan con la herramienta Firebase Authentication que proporciona servicios de backend, SDK fáciles de usar y bibliotecas de IU ya elaboradas para autenticar a los usuarios. Por otro lado, están los datos relativos a los pacientes y las grabaciones que se almacenan en la base de datos en tiempo real de Firebase, una base de datos NoSQL alojada en la nube en la que los datos, almacenados en formato JSON, se sincronizan con todos los clientes en tiempo real. Además, en Firebase Storage se almacenan los ficheros CSV de cada grabación, y en caso de existir, el vídeo de la grabación.

Para visualizar el modelo de datos de la aplicación se ha realizado el siguiente diagrama para poder entender mejor la estructura de los datos, puesto que la base de datos en tiempo real es NoSQL, cada una de las clases del diagrama no representa una tabla de la base de datos:

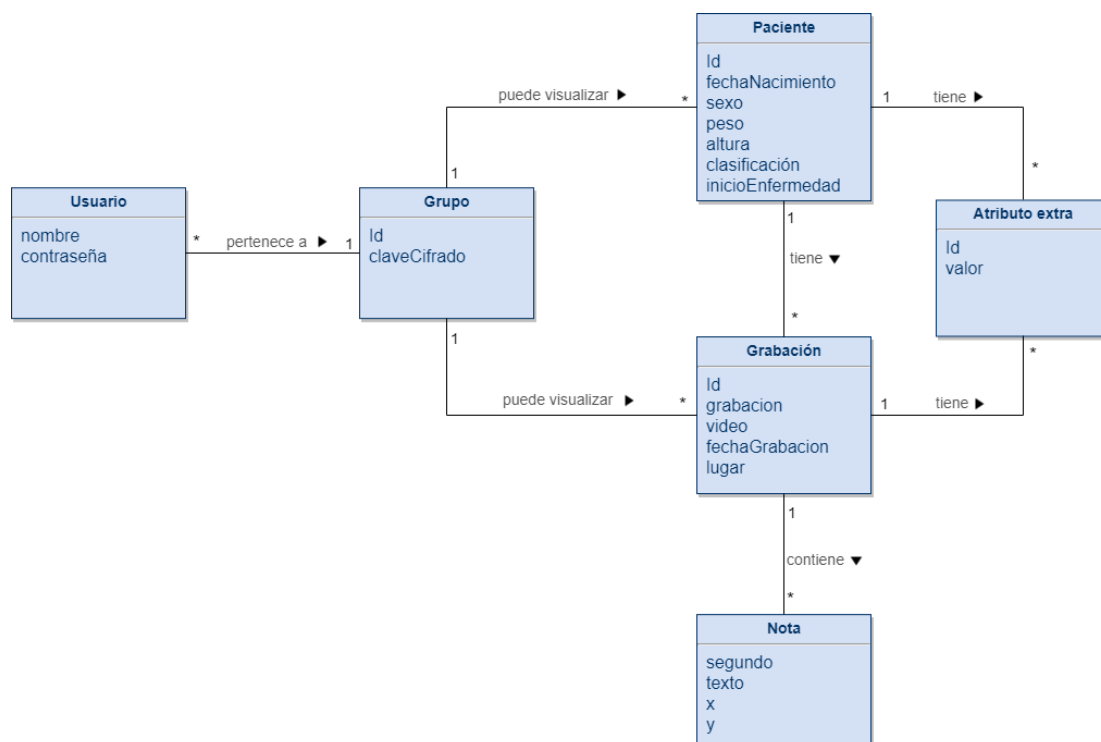


Ilustración 18. Modelo de datos

El modelo de la base de datos se ha diseñado en base a las siguientes consideraciones:

- La aplicación va a ser usada por distintos grupos de usuarios, por ejemplo, los doctores del Hospital Donostia representarían un grupo.
- Cada grupo de usuarios tiene una clave de cifrado con la que se cifrará toda la información generada por los usuarios del grupo.
- Cuando un usuario añade un paciente al sistema, todos los usuarios de su grupo pueden visualizar/gestionar ese paciente.
- Cada usuario de la aplicación pertenecerá a un grupo de usuarios.
- Para cada paciente se registrará un identificador, la fecha de nacimiento, sexo, clasificación y año de inicio de la enfermedad. De forma opcional el peso y la altura.
- Cada grabación pertenecerá al grupo de usuarios del usuario que haya realizado el registro, y solamente será visible para ese grupo.
- Para cada grabación se registrará un identificador, el lugar, la fecha en la que se llevó a cabo y el fichero CSV de la grabación. Opcionalmente se podrá incluir su vídeo.
- Tanto los pacientes como las grabaciones pueden tener tantos atributos extra como se crea conveniente.
- Los atributos extra estarán compuestos de un identificador y un valor.
- Los pacientes tendrán sus grabaciones asociadas.
- Cada grabación corresponde a un único paciente.
- A las grabaciones se les puede añadir notas, que se mostrarán durante su reproducción.
- Las notas guardan el segundo de reproducción, el texto y las coordenadas.

A continuación, se muestra cuál es la estructura real con la que se almacenan los datos en la base de datos de Firebase. La estructura es una jerarquía que recuerda a un documento JSON, en la cual se pueden observar los elementos citados en el modelo de datos:



Ilustración 19. Firebase Database

Por otro lado, los ficheros se almacenan en la Firebase Storage, una herramienta muy similar al sistema de ficheros de Windows:

Nombre	Tamaño	Tipo	Última modificación
grabaciones/	–	Carpeta	–
grabacionesJSON/	–	Carpeta	–
videos/	–	Carpeta	–

Ilustración 20. Firebase Storage

6.3. Seguridad

En este apartado se abordará el diseño de la seguridad. Por un lado, se hará un análisis exhaustivo del nivel de seguridad ofrecido por Firebase, por otro lado, se analizarán las alternativas para gestionar la seguridad en el lado cliente. La información sobre el cifrado de datos en Firebase se ha extraído de [16].

6.3.1. Gestión segura de los datos en Firebase

Puesto que los datos con los que trabaja la aplicación web son datos de pacientes reales, es necesario asegurar que se cumple con la normativa establecida en la LOPD, que entre otras cosas establece que los datos deben ser cifrados y guardados bajo estrictas medidas de seguridad.

El cifrado o encriptación es un proceso que toma datos legibles como entrada, llamada texto plano, y lo transforma en una salida, llamada texto cifrado, que revela poca o ninguna información sobre el texto plano. Mediante el cifrado se añade una capa extra para la protección de los datos, asegurando que, si los datos caen en manos de un atacante, este será incapaz de entender o descifrar la información sin tener acceso a las claves de cifrado.

En pos del cumplimiento de la LOPD, se ha realizado un análisis exhaustivo de la seguridad que ofrece Firebase, una plataforma para el desarrollo de aplicaciones web ofrecida por Google donde se aloja tanto la aplicación como los datos.

El análisis se divide en dos apartados; por un lado, se analiza la seguridad en el transporte de los datos entre el navegador del cliente y los servidores de Firebase, y por otro, la seguridad en el almacenamiento de los datos.

6.3.1.1. Transporte cifrado de los datos

Todo el tráfico entre los servidores de la base de datos de Firebase y el navegador del usuario pasa por conexiones cifradas. Esto se puede ver fácilmente observando la URL que se usa para conectarse a la aplicación, que es de la forma `https://<nombreApp>.firebaseapp.com`.

Que la conexión se realice mediante el protocolo HTTPS implica que se utiliza un cifrado basado en SSL/TLS para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente), siendo este más apropiado para el tráfico de información sensible que el protocolo HTTP. De esta forma se evitan ataques man-in-the-middle y eavesdropping, y se asegura que la comunicación entre cliente y servidor sea confidencial.

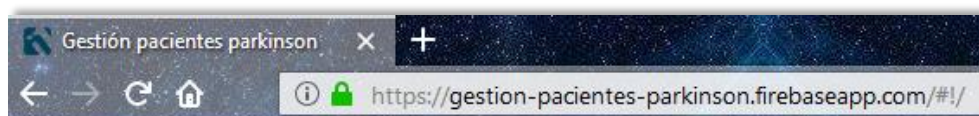


Ilustración 21. Transporte cifrado de los datos

6.3.1.2. Cifrado en reposo en Firebase

Por cifrado en reposo, se hace referencia al cifrado utilizado para proteger los datos almacenados en un disco o medios de respaldo. Google usa varias capas de encriptación para proteger los datos, de esta forma añade protección redundante a los datos.

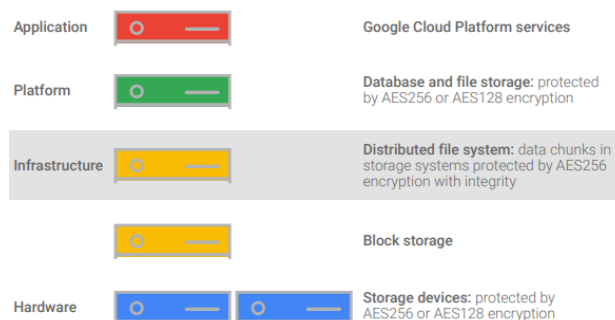


Ilustración 22. Capas de encriptación usadas en Firebase, extraída de [17]

Para entender cómo funciona la encriptación en los servicios de almacenamiento de Firebase es importante definir la forma en la que Google almacena los datos. Los datos se dividen en archivos más pequeños denominados fragmentos o *chunks* de cara al almacenamiento; siendo el tamaño de cada uno de estos fragmentos de hasta varios GB. Cada fragmento se cifra a nivel de almacenamiento con una clave de cifrado individual: dos fragmentos no tendrán la misma clave de cifrado, incluso si son parte del mismo Objeto de Firebase, propiedad del mismo cliente o se almacenan en la misma máquina. Cuando se actualiza un fragmento, se cifra con una nueva clave, en lugar de reutilizar la clave anterior. Esta partición de datos, en la que cada uno se usa una clave diferente, garantiza que el "radio de explosión" de una clave se limita únicamente al fragmento que hace uso de esa clave.

La encriptación de los datos se lleva a cabo antes de ser escritos en el disco.

Cada fragmento de datos tiene un identificador único. Las listas de control de acceso (ACL) garantizan que cada fragmento puede ser descifrado sólo por los servicios de Google que operan bajo roles autorizados, a los que se les concede acceso en ese momento. Esto previene el acceso a los datos sin autorización, reforzando tanto la seguridad como la privacidad de los datos.

Cada fragmento se distribuye y se replica encriptado para poder recuperar la información en caso de desastres. Una persona maliciosa que quiera acceder a los datos de los clientes necesitaría acceder (1) a todos los fragmentos de almacenamiento correspondientes a los datos que desean, y (2) conocer las claves de cifrado correspondientes a cada uno de los fragmentos.



Ilustración 23. Gestión de datos en Google, adaptada de [18]

Google usa el algoritmo Advanced Encryption Standard (AES) para encriptar los datos en reposo. AES es ampliamente utilizado pues tanto AES256 como AES128 son recomendados por el Instituto Nacional de Estándares y Tecnología (NIST) para su uso en almacenamiento a largo plazo (desde noviembre de 2015).

6.3.1.3. Gestión de claves

La clave utilizada para encriptar los datos de un fragmento es llamada *data encryption key* (DEK). Las DEK se encriptan con una *key encryption key* (KEK). Junto a cada fragmento se almacena la DEK encriptada. Las KEKs se almacenan en el Servicio de Administración de Claves de Google (KMS), un repositorio construido específicamente para este propósito.

Los datos almacenados en Firebase están encriptados con DEK usando AES256 o AES128. Las DEK están envueltas con KEK usando AES256 o AES128.

Cuando un servicio de Firebase accede a un fragmento cifrado de datos, esto es lo que sucede:

1. El servicio realiza una llamada al sistema de almacenamiento para obtener los datos que necesita.
2. El sistema de almacenamiento identifica los fragmentos en los que se almacenan esos datos y dónde están almacenados.
3. Para cada fragmento, el sistema de almacenamiento extrae la DEK envuelta almacenada junto al fragmento y la envía al KMS para desenvolverla de su KEK.
4. El sistema de almacenamiento verifica que se tiene permiso para acceder a ese fragmento de datos y el KMS verifica que el sistema de almacenamiento esté autorizado para usar el KEK asociado con el servicio, y para desenvolver ese DEK específico.
5. El KMS hace una de las siguientes cosas:
 - Pasa el DEK desenvuelto de vuelta al sistema de almacenamiento, que descifra el fragmento de datos y lo pasa al servicio.
 - O, en algunos casos raros, pasa el DEK desenvuelto al servicio; el sistema de almacenamiento pasa el fragmento de datos encriptados al servicio, que descifra el fragmento de datos y lo usa.



Ilustración 24. Gestión de claves para desencriptar un fragmento de datos, adaptado de [19]

6.3.2. Encriptación de los datos en local

En este apartado se define el proceso seguido para elegir un algoritmo de encriptación y posteriormente una librería criptográfica de JavaScript con la que se encriptarán los datos usando el algoritmo previamente escogido. Firebase provee tanto de comunicaciones seguras entre cliente y servidor, mediante el protocolo SSL/TLS, como de almacenamiento seguro mediante el cifrado de los datos usando el algoritmo AES, tal y como se ha definido anteriormente. El objetivo de este apartado es definir una estrategia para aumentar la seguridad de la aplicación. Concretamente, el propósito es evitar que el administrador del sistema, con acceso al panel de control del proyecto en Firebase, tenga acceso a los datos de los pacientes. Firebase está configurado de tal forma que el usuario con acceso al panel de control es capaz de visualizar cualquier dato almacenado en la base de datos y cualquier fichero almacenado en Firebase Storage. Por lo tanto, la meta es encriptar esta información en el cliente, antes de ser enviada al servidor de Firebase, para que ni siquiera el administrador del sistema tenga acceso a la misma. Por defecto el administrador de Firebase visualizará los datos tal y como se muestra en la siguiente imagen:

```
-LCZPCKiGCNhFaQ8ckJ-
├── altura: 165
├── apellido: "Salas"
├── extra
│   ├── Jubilado: "No"
│   ├── Localidad: "Madrid"
│   └── Telefono: "698532147"
├── fechaNacimiento: "14/10/1961"
├── nombre: "Paquita"
└── sexo: "Mujer"
```

Ilustración 25. Visualización por defecto de los datos en Firebase

6.3.2.1. Fuente de información.

Para realizar este análisis, se han analizado varias comparativas de algoritmos criptográficos en las que se analizan las principales ventajas e inconvenientes de su uso, además de su rendimiento. En cuanto a la librería de JavaScript, se ha recogido información de [20] sobre el rendimiento de varios algoritmos criptográficos y del rendimiento de distintas librerías criptográficas [21].

6.3.2.2. Algoritmo de encriptación

A continuación, se comparan los principales algoritmos de encriptación (DES, AES y 3DES) para finalmente escoger el más apropiado.

DES fue el estándar de cifrado de flujo de 56 bits usado en 1976. Esto hace que las cantidades de espacio para la clave sean aproximadamente de 72 mil billones de posibilidades. A pesar de que parece grande, actualmente no es suficiente y es vulnerable a los ataques de fuerza bruta.

3DES consiste en aplicar el cifrado DES tres veces en secuencia. 3DES se aplica con tres claves de cifrado diferentes (K1, K2 y K3), por lo que la longitud efectiva de la clave es de 168 bits, haciendo este método de cifrado muchísimo más seguro que el DES.

AES es el algoritmo más utilizado hoy en día y el cifrado simétrico estándar del gobierno de los Estados Unidos. La longitud de clave puede ser de 128 bits (10 rondas), 192 bits (12 rondas) o 256 bits (14 rondas). El algoritmo está basado en sustituciones, permutaciones y transformaciones lineales, ejecutadas varias veces en bloques de datos de 16 bytes. El cambio de un solo bit, ya sea en la clave, o en los bloques de texto, resulta en un bloque de texto cifrado completamente diferente. AES supera a 3DES tanto en software como en hardware.

	DES	3DES	AES
Desarrollado	1976	1998	2000
Longitud clave	56 bits	168 bits	128, 192 o 256 bits
Tipo de cifrado	Simétrico	Simétrico	Simétrico
Tamaño de bloque	64 bits	64 bits	128 bits
Seguridad	Vulnerable	3 veces más seguro que DES	Segura

Tabla 6. Comparación algoritmos de encriptación

De tal modo que se decide usar el algoritmo AES para encriptar los datos por el nivel de seguridad que proporciona y por el buen rendimiento que presenta.

6.3.2.3 Gestión de las claves de cifrado

Tan importante como la elección del algoritmo de encriptación es la gestión de las claves con las que se llevará a cabo el cifrado de los datos. Partiendo de la base de que cada grupo de usuarios de un mismo colectivo, por ejemplo, los usuarios del Hospital Donostia, usarán la misma clave de cifrado, se contemplan las siguientes estrategias para gestionar las claves:

Gestión local. La clave de cifrado/descifrado se solicita al usuario a la hora de iniciar sesión en la aplicación. Su hash MD5 que se usará como clave se almacena en una cookie del navegador y su hash SHA256 se almacena en Firebase para que al iniciar sesión se pueda comprobar que la clave introducida es correcta. En cada comunicación entre cliente y servidor se usa la clave almacenada en la cookie bien para cifrar los datos, en caso de que se envíe información al servidor, o bien para descifrar los datos recibidos del servidor antes de mostrarlos. Por lo tanto, las claves no salen del navegador del cliente. De esta forma se evita tanto que el administrador del sistema pueda visualizar los datos en plano desde Firebase como que tenga acceso a las

claves de cifrado usadas por los usuarios de la aplicación. Sin embargo, esta estrategia no contempla la recuperación de claves. Dada la situación de que a un grupo de usuarios se les olvida la clave de cifrado esta es irrecuperable pues no está almacenada en ningún lugar, el hash criptográfico solamente sirve para comprobar su validez, no para recuperarla. Lo mismo ocurre con los datos del grupo en cuestión, puesto que se almacenan cifrados con la clave, en el momento en el que se pierde el acceso a la clave se pierde el acceso a los datos.

Gestión en Firebase. Generar una clave de cifrado/descifrado por cada grupo de usuarios y almacenarla en Firebase, de esta forma cada vez que un usuario perteneciente a un grupo accede a la aplicación automáticamente se recupera la clave asociada al grupo. Durante el tiempo que esté la sesión activa, la clave se almacena en una cookie, haciendo uso de ella en las comunicaciones entre cliente y servidor. En este caso, las claves se almacenan tanto en el cliente como en el servidor. El envío de la clave entre servidor y cliente se realiza de forma segura mediante el protocolo SSL/TLS y en Firebase se almacena cifrada. Aunque de esta forma no evitamos que el administrador del sistema (usuario con permisos para acceder a la consola de Firebase) tenga acceso a las claves, ya que desde la consola de Firebase es posible visualizar los datos sin la capa de cifrado con la que Firebase los almacena.

Servicio de gestión de claves en la nube. Existen en Internet diversos servicios para gestionar claves. Entre ellos se encuentran AWS Key Management Service (Amazon) o Cloud Key Management Service (Google). Estos servicios permiten ejercer un control centralizado sobre las claves de cifrado usadas para proteger los datos. Permiten crear, rotar, deshabilitar y eliminar las claves de cifrado, así como definir las políticas de uso de dichas claves y auditar el uso que se hace de las mismas. Por lo tanto, para cada grupo de usuarios se crearía una clave de cifrado mediante el servicio de gestión, quedando la clave asociada a ese grupo. Cuando un usuario accede a la aplicación se recupera la clave de cifrado del grupo al que pertenece y se almacena en una cookie mientras dure la sesión, quedando disponible para su uso en las comunicaciones posteriores entre cliente y servidor. De esta forma se evita que el administrador del sistema tenga acceso a la información almacenada en Firebase y a las claves de los usuarios. La principal desventaja es que son servicios de pago.

Se decide realizar la gestión de claves de forma local. Se descarta la gestión en Firebase por no cumplir el objetivo principal, que el administrador del sistema no tenga acceso a la información almacenada en Firebase. También se descarta usar un servicio de gestión de claves, en este caso por ser un servicio de pago y porque implica un mayor coste de desarrollo, debido a que hay que configurar el sistema para que la comunicación con el servicio se realice de forma segura. De igual forma habría que realizar una administración adecuada de las claves, creando una nueva cuando se crea un nuevo grupo de usuarios, recuperando la clave correspondiente al grupo del usuario cuando este inicia sesión y permitiendo al grupo de usuarios recuperar su clave en caso de olvido. Gestionando las claves localmente se simplifica la administración de las claves, ya que solamente hay que asegurar que el usuario introduzca la clave de cifrado de su grupo al iniciar sesión y almacenarla temporalmente en una cookie, para posteriormente usarla en las comunicaciones entre cliente y servidor.



Ilustración 26. Visualización de los datos cifrados en Firebase

Es evidente que, si la aplicación se implementase fuera del ámbito académico, la estrategia elegida no sería la adecuada, ya que tal y como se ha comentado con anterioridad, en caso de que un grupo de usuarios pierda su clave, la información de ese grupo se perdería su opción a ser recuperada. De manera que en un entorno profesional la estrategia adecuada sería usar un servicio de gestión de claves en la nube.

6.3.2.4 Herramienta de encriptación

Para llevar a cabo la encriptación de los datos se ha usado la librería **CryptoJS**. Esta librería desarrollada por Google consta de una colección de algoritmos criptográficos estandarizados y seguros implementados en JavaScript utilizando las mejores prácticas y patrones. Son rápidos y tienen una interfaz consistente y simple.

6.4. Diagramas de secuencia

En este apartado se presentan varios diagramas de secuencia. Se han elegido los casos de uso iniciar sesión, ver pacientes y visualizar grabación por ser los más representativos de la aplicación. Con estos tres diagramas se obtiene una visión global del funcionamiento de la aplicación.

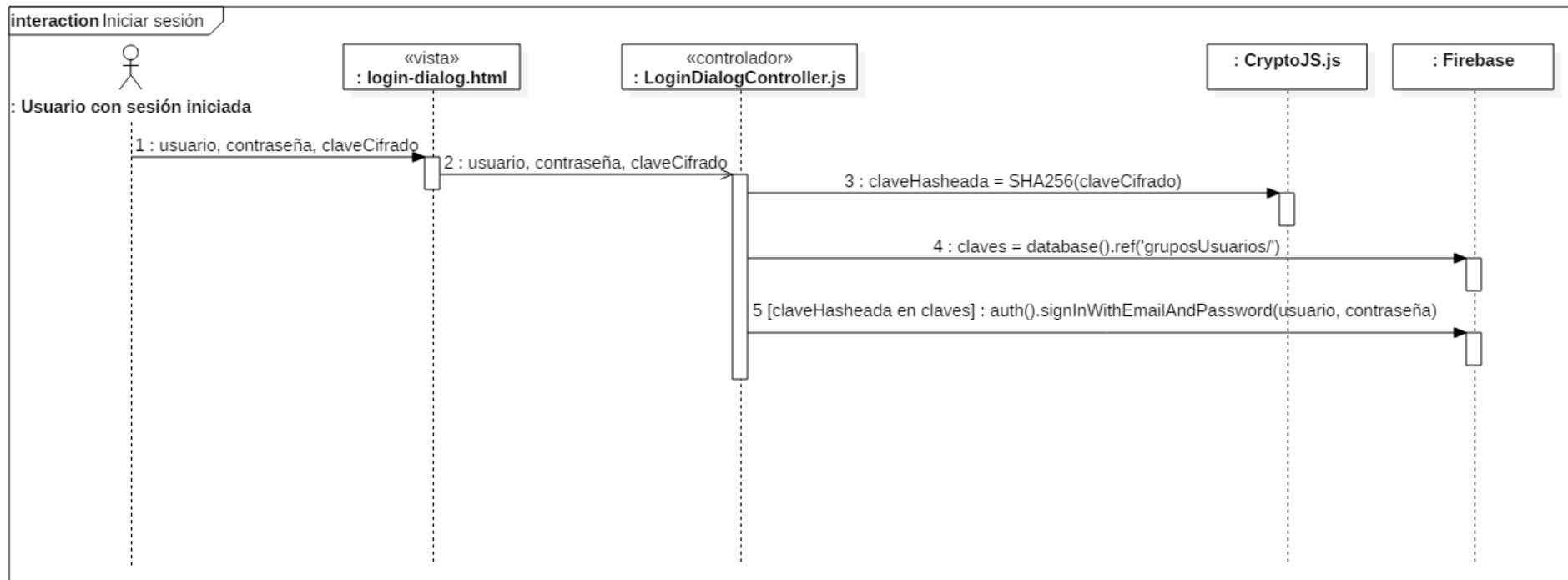


Ilustración 27. Diagrama Secuencia Iniciar sesión

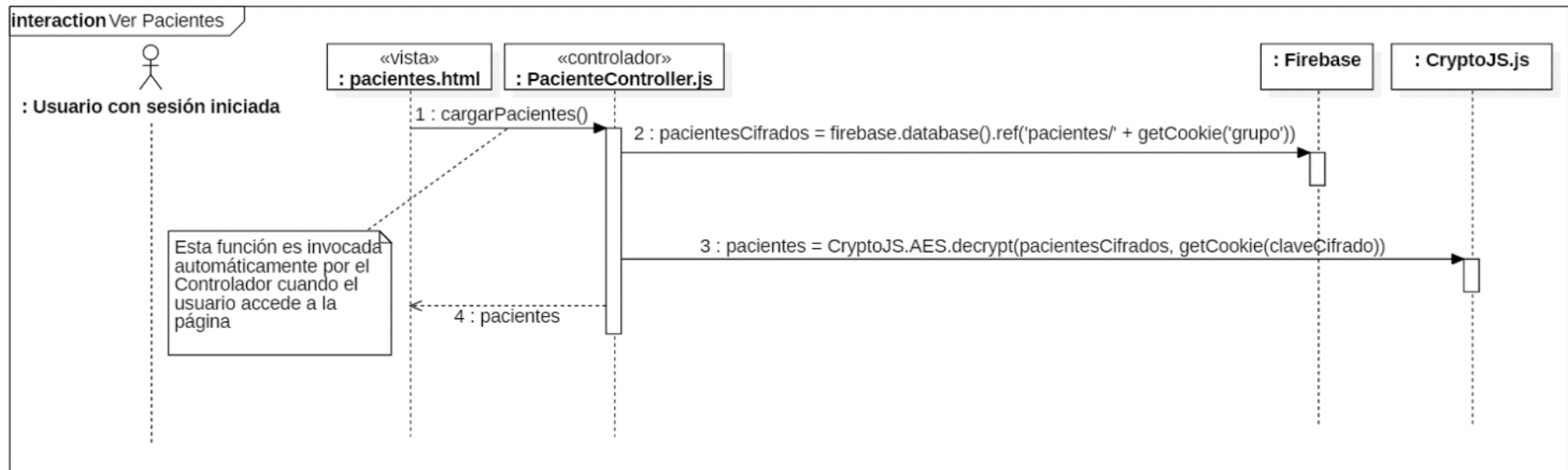


Ilustración 28. Diagrama Secuencia Ver pacientes

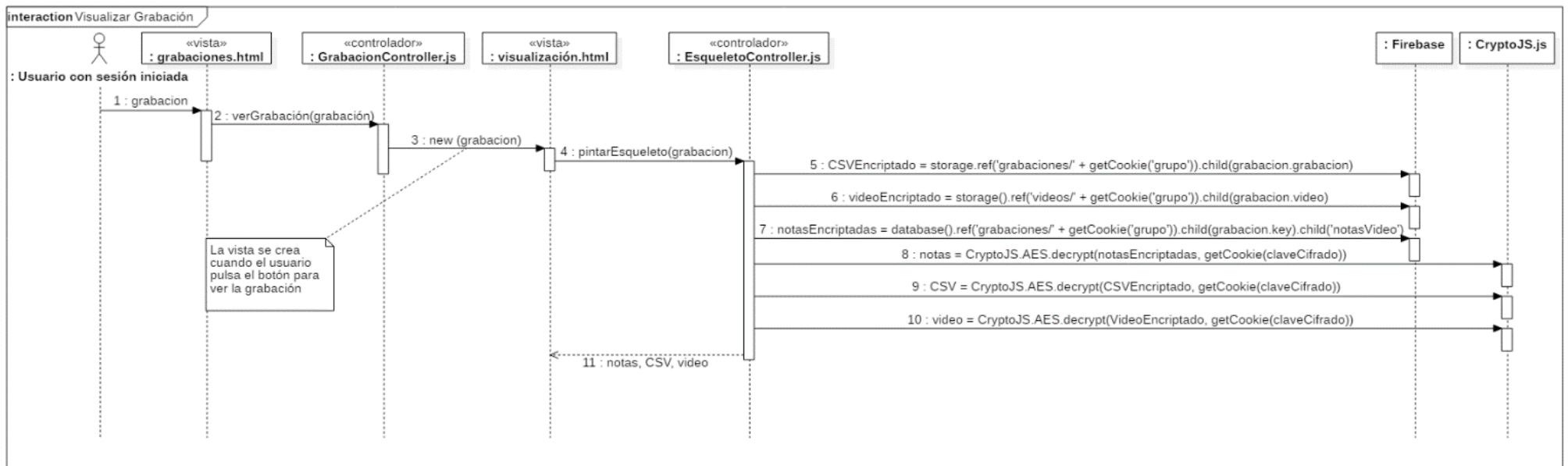


Ilustración 29. Diagrama Secuencia Visualizar grabaciones

7. Implementación

En este capítulo se aborda la implementación de la aplicación. Tal y como se ha explicado en el capítulo anterior, la aplicación sigue el patrón MVC.

Por un lado, se detalla cómo se ha implementado el patrón MVC en la aplicación y se explican las partes más relevantes del desarrollo. Por otro lado, se describen los problemas y dificultades encontrados durante el desarrollo.

7.1. Vistas

Las vistas están desarrolladas en código HTML5 y CSS3, además incluyen directivas propias de AngularJS. Con el lenguaje HTML se establece el esqueleto de la vista; cómo se distribuye la información sobre la vista y cuál es esa información. La información que se puede mostrar con HTML es información estática, será necesario el uso de AngularJS para dotar a esa información de dinamismo. Por otro lado, las hojas de estilo CSS se encargan del aspecto del esqueleto, especificando la fuente, el tamaño de la letra, los colores e incluso animaciones. Todo ello con el objetivo de que la aplicación sea atractiva e intuitiva. Tal y como se ha comentado, las directivas de AngularJS son imprescindibles en las vistas, pues gracias a ellas es posible comunicar la vista con el controlador. En la siguiente imagen se pueden ver las principales vistas de la aplicación:



Ilustración 30. Vistas principales de la aplicación y hoja de estilos

La vista home es la página principal, pacientes corresponde a la sección en la que se gestionan los pacientes y grabaciones es la página donde se gestionan las grabaciones. Además de estas vistas, la aplicación cuenta con varias más que forman parte de los controladores. El motivo por el que no son vistas independientes como las de la imagen es porque se muestran como ventanas emergentes y resulta más cómodo que estén definidas directamente en el controlador en forma de plantilla. Un ejemplo de estas plantillas sería la vista de recuperación de contraseña, que se muestra a continuación.

```
template: ' <md-dialog aria-label="Restaurar">\n' +
  <form id="passwordForm" name="passwordForm" ng-submit="restaurar()">\n' +
  <md-toolbar>\n' +
  <div class="md-toolbar-tools">\n' +
  <h2>Restaurar contraseña</h2>\n' +
  <span flex></span>\n' +
  <md-button class="md-icon-button" ng-click="close()">\n' +
  <md-icon md-svg-src="img/icons/ic_close_24px.svg" aria-label="Close dialog"></md-icon>\n' +
  </md-button>\n' +
  </div>\n' +
  </md-toolbar>\n' +
  <md-dialog-content>\n' +
  <div class="md-dialog-content" style="min-width: 350px; max-width: 80%; max-height: 60%;">\n' +
  <md-input-container class="md-block">\n' +
  <label>Usuario</label>\n' +
  <input name="username" ng-model="email" type="email" md-autofocus required title="email"\n' +
  autocomplete="username"/>\n' +
  <div ng-messages="passwordForm.username.$error" ng-show="passwordForm.username.$touched">\n' +
  <div ng-message="required">Es obligatorio introducir el correo electrónico</div>\n' +
  <div ng-message="email">Introduce una dirección de correo válida</div>\n' +
  </div>\n' +
  </md-input-container>\n' +
  <div id="error-submit" style="color:red"></div>\n' +
  </div>\n' +
  </md-dialog-content>\n' +
  <md-dialog-actions layout="row">\n' +
  <md-button type="submit" ng-disabled="passwordForm.$invalid" class="md-raised md-primary">Restaurar</md-button>\n' +
  </md-dialog-actions>\n' +
  </form>\n' +
  </md-dialog>',
```

Ilustración 31. Plantilla de una vista

La imagen anterior también sirve para ilustrar las directivas de AngularJS antes mencionadas. Por ejemplo, la directiva ng-model que se encuentra en la etiqueta input, sirve para que lo que el usuario escriba en ese input se almacene en una variable JavaScript de mismo nombre en su correspondiente controlador y viceversa. En el fragmento de código también se pueden ver directivas propias de AngularJS Material en forma de etiqueta HTML como md-dialog, con la que se define una ventana emergente.

Las hojas de estilo también forman parte de la Vista, en la aplicación se ha hecho uso de la librería AngularJS Material, que incluye directivas propias con elementos de Material Design y su correspondiente CSS. Además, se ha definido una hoja de estilos propia para complementar y mejorar lo proporcionado por la librería.

7.2. Controladores y Modelos

Tanto los Controladores como los modelos forman parte de ficheros JavaScript que se ejecutan en el lado cliente. Cada Vista está asociada como mínimo a un Controlador. Los Controladores se encargan de recibir las peticiones desde las Vistas y realizar, en caso de ser necesario, la petición de datos al Modelo correspondiente. En la aplicación desarrollada no existen ficheros que representen el Modelo, puesto que al trabajar con una base de datos NoSQL en Firebase, la información no está estructurada y no es necesario crear clases para representar el Modelo. Es por esto que el papel de los Modelos lo realizan variables de JavaScript y el \$scope de AngularJS; un objeto que hace referencia al Modelo de la aplicación y es capaz de almacenar tanto variables como funciones en el contexto de un controlador asociado a una vista. Los Controladores desarrollados son:

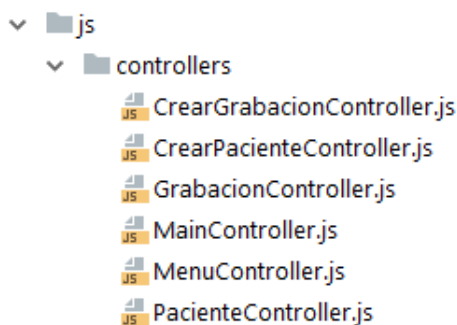


Ilustración 32. Controladores de la aplicación

El Controlador MainController es el Controlador encargado de la página principal de la aplicación y del inicio de sesión. MenuController es el responsable del menú lateral. CrearGrabacionController, tal y como su nombre indica se encarga de la creación de las grabaciones y su posterior encriptación y almacenamiento en la base de datos. CrearPacienteController realiza lo propio con los pacientes. GrabacionController se ocupa de la visualización de las grabaciones en una tabla, de su edición y borrado, también se encarga de la previsualización del fichero CSV de la grabación y la reproducción sincronizada de la grabación junto con el esqueleto y el vídeo. PacienteController se ocupa de la visualización de los pacientes en una tabla, de su edición y borrado.

En cuanto a los Modelos, estos aparecen representados por variables del \$scope, un ejemplo de ello sería la variable \$scope.paciente que se encuentra en el Controlador CrearPacienteController. Esta variable es un objeto de JavaScript, un objeto es una entidad independiente con propiedades y tipos. Los objetos pueden tener propiedades, que definen sus características. En este caso nombre, fecha, inicioEnfermedad..., son propiedades del objeto paciente.

```
$scope.paciente = {
  nombre: '',
  fecha: new Date(),
  inicioEnfermedad: (new Date()).getFullYear(),
  clasificacion: '',
  altura: '',
  peso: '',
  sexo: '',
  grabacion: '',
  atributosExtraNombre: [],
  atributosExtraValor: []
};
```

Código 1. Modelo paciente

A la hora de implementar el formulario en el que se crearán los pacientes, bastará con asociar en la Vista cada campo del formulario con una propiedad del objeto definido en el Controlador mediante la directiva ng-model de AngularJS. De esta forma, el objeto del Controlador tendrá siempre en cada una de sus propiedades los valores que estén en ese momento en el formulario. Una vez que el usuario haya completado el formulario, bastará con tomar los datos del objeto \$scope.paciente, cifrarlos y almacenarlos en la base de datos de Firebase. El cifrado de datos se explica en profundidad en la sección [7.5 Encriptado de los datos](#). La comunicación con Firebase se detalla a continuación.

7.3. Comunicación con Firebase Database

La comunicación con Firebase Database, la base de datos NoSQL de Firebase, se lleva a cabo desde los Controladores, estos son los encargados de solicitar y enviar los datos. Para ello es necesario definir una instancia de `firebase.database.Reference` que referencie la clave que se quiere consultar o sobre la que se quiere escribir. Por ejemplo, si el objetivo es trabajar con las grabaciones, será necesario definir la siguiente referencia:

```
let grabacionesRef =
  firebase.database().ref('grabaciones').child(getCookie('grupo');
```

Código 2. Definir instancia referencia Firebase

En el código mostrado se puede observar cómo se referencia a la clave grabaciones, y más concretamente a las grabaciones que pertenecen al grupo del usuario que actualmente tiene la sesión iniciada. El nombre del grupo del usuario conectado se almacena en una cookie mientras dure su sesión, de esa forma, la información es de fácil acceso para el desarrollador. Una vez definida la referencia sobre la clave deseada, ya es posible leer la información disponible en esa clave o escribir en ella.

7.3.1. Lectura de datos en Firebase

En caso de querer leer el contenido bastará con ejecutar el siguiente código, en él se puede observar cómo la función *once* carga la información relativa a las grabaciones en la variable *grabaciones*. A continuación, se recorre cada una de las grabaciones disponibles, es necesario aplicar la función *val()* a la grabación antes de poder trabajar con ella. Esto se debe a que la grabación está representada por un *DataSnapshot*, que es una instantánea de los datos. Una instantánea es una imagen de los datos en una referencia particular de la base de datos en un momento específico. Si se llama a *val()* en una instantánea, se obtiene la representación de los datos como objeto [22]:

```
$scope.promise = grabacionesRef.once('value', function (grabaciones) {
    grabaciones.forEach(function (grabacion) {
        let childData = grabacion.val();
        (...)
    });
});
```

Código 3. Lectura en Firebase

La variable *childData* será un objeto que contenga todas las propiedades de la grabación, hay que tener en cuenta que antes de poder trabajar con esa información es necesario descifrarla, pues se ha obtenido la información tal y como está almacenada en Firebase.

7.3.2. Escritura de datos en Firebase

Cuando se quieran introducir nuevos datos en Firebase, será necesario ejecutar el siguiente código tras haber encriptado los datos. Para llevar a cabo la inserción primero se aplica la función *push()*, esta función genera una clave única para el objeto que estamos insertando. De esta forma evitamos el problema de las claves duplicadas, cada paciente o grabación tendrá una clave única como identificador. La función *set()* se usa para guardar datos en la referencia especificada y reemplazar los datos existentes en esa ruta de acceso [23].

```
grabacionesRef.push().set(atributosObligatorios).then(function => {
    showToast('Grabación creada correctamente');
}).catch(er => {
    showToast('Error creando la grabación');
});
```

Código 4. Escritura en Firebase

La variable *atributosObligatorios* contiene un objeto JavaScript con todos los atributos de la grabación que se han obtenido del formulario. Tras la inserción se muestra una notificación Toast al usuario indicando si la inserción se ha realizado con éxito.

7.3.3. Actualización de datos en Firebase

Para llevar a cabo una actualización en Firebase hay que referenciar la clave sobre la que se quiere realizar la modificación y ejecutar la función `update()`. En el siguiente código se puede observar la actualización del `lugar` de una grabación. La grabación se referencia usando el atributo `key`, que contiene la clave de Firebase definida por la función `push()`. Una vez referenciada la grabación específica sobre la que se quiere realizar la modificación, se ejecuta la función `update()`, a la que se le pasa como parámetro un objeto JavaScript con las propiedades de la grabación que se van modificar junto con su nuevo valor. En caso de no existir esa propiedad en la grabación, esta se inserta como una propiedad nueva.

```
grabacionesRef.child(grabacion.key).update({  
  'lugar': grabacion.lugar  
});
```

Código 5. Actualización Firebase

7.3.4. Borrado de datos en Firebase

Para borrar datos de Firebase hay que llamar a `remove()` en una referencia a la ubicación de los datos que se quieren borrar. A continuación, se puede observar cómo se lleva a cabo el borrado de una grabación. Para ello se referencia la grabación que se desea eliminar y se llama a `remove()`.

```
grabacionesRef.child(grabacion.key).remove();
```

Código 6. Borrado en Firebase

7.4. Comunicación con Firebase Storage

La comunicación con Firebase Storage, el servicio para almacenar y procesar ficheros, se lleva a cabo desde los Controladores, estos son los encargados de solicitar y enviar los ficheros. Para ello es necesario definir una instancia de `firebase.storage.Reference` que referencie el fichero que se quiere consultar o la ubicación en la que se quiere almacenar. Por ejemplo, si se quiere crear una referencia al directorio donde se almacenan las grabaciones de un grupo:

```
let grabacionesStorageRef =  
  firebase.storage().ref('grabaciones').child(getCookie('grupo'));
```

Código 7. Definir instancia referencia Firebase Storage

Como se puede observar en el fragmento de código, la idea es exactamente la misma que cuando se trabaja con Firebase Database.

7.4.1. Lectura de ficheros en Firebase

Para leer un fichero de Firebase Storage hay que llamar a `getDownloadURL()` con la referencia al archivo que se quiere descargar. Esta función devuelve la URL de descarga del fichero. A continuación, se puede obtener el fichero mediante la API Fetch [24], esta API además de una interfaz JavaScript para acceder y manipular partes de la tubería HTTP, como peticiones y respuestas, también provee un método global `fetch()` que proporciona una forma fácil y lógica de obtener recursos de forma asíncrona por la red. En el código que se muestra a continuación se puede ver cómo se obtiene y se imprime por consola el fichero CSV de una grabación. Para este ejemplo no se ha tenido en cuenta que los ficheros se almacenan encriptados en Firebase, sería necesario desencriptar el contenido de la variable `reader.result` antes de poder trabajar con él. El encriptado se explica en más detalle en el apartado [7.5 Encriptado de los datos](#).

```
grabacionesStorageRef.child(grabacion).getDownloadURL().then(function (url) {
  fetch(url)
    .then(res => res.blob()) //Obtiene la respuesta y la devuelvo como blob
    .then(blob => {
      var reader = new FileReader();
      reader.onload = function () {
        console.log(reader.result);
      };
      reader.readAsText(blob);
    });
});
```

7.4.2. Almacenar ficheros en Firebase

Para almacenar un fichero en Firebase existen varias alternativas. En el proyecto se han utilizado los métodos `put()` y `putString()`. El primero se emplea cuando el fichero a almacenar no es un fichero de texto, en el caso del proyecto, los ficheros de vídeo se almacenan como *Blobs*. El segundo se utiliza con ficheros de texto, en el proyecto, los ficheros CSV y JSON. Para almacenar el fichero hay que llamar a una de estas dos funciones con la referencia a la ubicación en la que se quiera almacenar el fichero. A continuación, se muestra un ejemplo para cada tipo de fichero, de nuevo, sin tener en cuenta el encriptado previo al almacenamiento.

```
let reader = new FileReader();
reader.onload = function (e) {
  let blob = new Blob([e.target.result], {type: "data:application/octet-
  stream"});
  grabacionesStorageRef.put(blob);
};
reader.readAsDataURL(file);
```

Código 8. Almacenar fichero de vídeo

```
let reader = new FileReader();
reader.onload = function (e) {
  grabacionesStorageRef.putString(e.target.result);
};
reader.readAsText (file);
```

Código 9. Almacenar fichero de texto

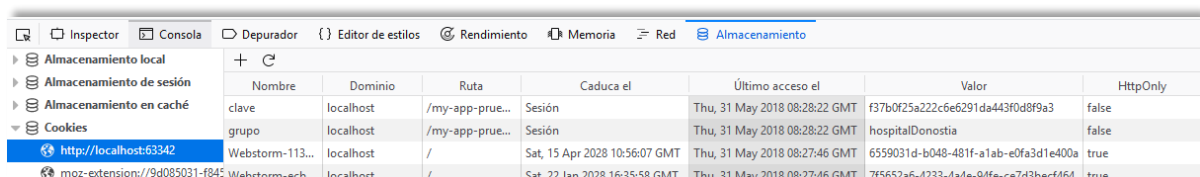
7.5. Encriptado de los datos

A continuación, se detalla el proceso seguido para obtener la clave de cifrado con la que se llevará a cabo el cifrado de los datos y su posterior uso en el proceso de cifrado. Ambos procesos se ilustran mediante su flujo de eventos, además, el proceso de cifrado/descifrado se acompaña de fragmentos de código donde se ve reflejado el flujo de eventos. La librería empleada para llevar a cabo la encriptación de los datos es CryptoJS [13], que consta de una colección de algoritmos criptográficos estandarizados y seguros implementados en JavaScript.

7.5.1. Proceso de obtención de la clave de cifrado

Tal y como se ha comentado en el capítulo [4. Tecnologías](#), la encriptación se lleva a cabo con los algoritmos proporcionados por la librería CryptoJS. El proceso de definición de la clave de encriptación de un grupo de usuarios se define a continuación:

1. El usuario inicia sesión en la aplicación web indicando su correo electrónico, su contraseña y la clave de cifrado.
2. Se comprueba a qué grupo de usuarios pertenece el nombre de usuario introducido.
 - 2.1. Si el usuario existe en la base de datos y pertenece a un grupo se procede a comprobar la clave de cifrado.
 - 2.1.1. Si no hay clave de cifrado asociada al grupo, se tomará la clave introducida como clave de cifrado del grupo y se almacenará en la base de datos el resultado de aplicar la función hash criptográfica SHA-256.
 - 2.1.2. Si existe una clave de cifrado para el grupo de usuarios, esta se compara con la introducida por el usuario tras aplicarle la función hash criptográfica SHA-256.
 - 2.1.2.1. Si coincide se procede al inicio de sesión.
 - 2.1.2.2. Si no coincide se muestra un mensaje de error y se vuelve al punto 1.
 - 2.2. Si el usuario no existe en la base de datos o no pertenece a ningún grupo se muestra un mensaje de error indicando que el usuario introducido no existe y se vuelve al punto 1.
3. Se almacena en una cookie el resultado de aplicar la función hash criptográfica MD5 a la clave introducida. Esta será la clave que se utilice para cifrar y descifrar todos los datos del grupo de usuarios.



Nombre	Dominio	Ruta	Caduca el	Último acceso el	Valor	HttpOnly
clave	localhost	/my-app-prue...	Sesión	Thu, 31 May 2018 08:28:22 GMT	f37b0f25a222c6e6291da443f0d8f9a3	false
grupo	localhost	/my-app-prue...	Sesión	Thu, 31 May 2018 08:28:22 GMT	hospitalDonostia	false
Webstorm-113...	localhost	/	Sat, 15 Apr 2028 10:56:07 GMT	Thu, 31 May 2018 08:27:46 GMT	6559031d-b048-481f-a1ab-e0fa3d1e400a	true
Webstorm-ecb...	localhost	/	Sat, 22 Jan 2028 16:35:58 GMT	Thu, 31 May 2018 08:27:46 GMT	7f5652a6-4233-4a4e-94fe-ce7d3becf464	true

Ilustración 33. Cookie con la clave de encriptación

7.5.2. Proceso de cifrado

Una vez que el usuario ha iniciado sesión correctamente y la clave de cifrado se encuentra en una cookie, el proceso de encriptación de los datos es el siguiente:

Si se están enviando datos a Firebase:

1. Si los datos son información relativa a pacientes o grabaciones, no son ficheros CSV ni vídeo.
 - 1.1. Para cada uno de los datos
 - 1.1.1. Se convierte a String en caso de no serlo.
 - 1.1.2. Se consulta la clave de cifrado almacenada en la cookie.
 - 1.1.3. Se aplica el algoritmo de encriptación AES con la clave obtenida.
 - 1.2. Se envía el conjunto de datos a la base de datos de Firebase.

```
let atributosObligatorios = {
  "id": CryptoJS.AES.encrypt($scope.paciente.nombre, getCookie('clave')).toString(),
  "sexo": CryptoJS.AES.encrypt($scope.paciente.sexo, getCookie('clave')).toString(),
};
pacientesRef.push().set(atributosObligatorios);
```

Código 10. Encriptación Strings

2. Si los datos son un fichero CSV
 - 2.1. Se lee el fichero usando la función *readAsText (fichero)* que devuelve el contenido del archivo como una cadena de texto.
 - 2.2. Se consulta la clave de cifrado almacenada en la cookie.
 - 2.3. Se aplica el algoritmo de encriptación AES con la clave obtenida.
 - 2.4. Se envía la cadena encriptada a Firebase Storage.

```
let reader = new FileReader();
reader.onload = function (e) {
  let fichero = reader.result;
  let encriptado = CryptoJS.AES.encrypt(fichero, getCookie('clave')).toString();
  storageRef.putString(CryptoJS.AES.encrypt(encriptado,
    getCookie('clave')).toString());
};
reader.readAsText(file);
```

Código 11. Encriptación CSV/JSON

3. Si los datos son un vídeo
 - 3.1. Se lee el fichero usando la función *readAsDataURL(vídeo)* que devuelve una URL que representa la información del archivo como una cadena de caracteres codificados en base64. De esta forma no se pierden metadatos, esenciales para su reproducción.
 - 3.2. Se consulta la clave de cifrado almacenada en la cookie.
 - 3.3. Se aplica el algoritmo de encriptación AES con la clave obtenida.
 - 3.4. Se envía la cadena encriptada a Firebase Storage.

```
let reader = new FileReader();
reader.onload = function (e) {
  let encriptado = CryptoJS.AES.encrypt(e.target.result, getCookie('clave'));
  let blobEncriptado = new Blob([encriptado], {type: "data:application/octet-stream"});
  let task = storageRef.put(blobEncriptado);
};
reader.readAsDataURL(file);
```

Código 12. Encriptación vídeo

7.5.2. Proceso de descifrado

Si se están leyendo datos de Firebase es necesario tener en cuenta que al descifrar con CryptoJS este devuelve un objeto de tipo *WordArray* [25], que representa un array de palabras de 32 bits. Como consecuencia es necesario convertir el objeto a un String con codificación UTF-8 antes de poder trabajar con la información:

1. Si los datos son información relativa a pacientes o grabaciones, no son ficheros CSV ni vídeos.
 - 1.1. Para cada uno de los datos
 - 1.1.1. Se obtiene el dato como un String encriptado.
 - 1.1.2. Se consulta la clave de cifrado almacenada en la cookie.
 - 1.1.3. Se aplica el algoritmo AES de descifrado con la clave obtenida.
 - 1.1.4. Se convierte a String codificado en UTF-8.
 - 1.1.5. Los datos ya son legibles y se pueden mostrar al usuario.

```
paciente.forEach(function (pacientes) { //Caso general, algún atributo se
trata de forma diferente
  let childData = pacientes.val();
  for (let clave in childData) {
    childData[clave] = CryptoJS.AES.decrypt(childData[clave],
      getCookie('clave')).toString(CryptoJS.enc.Utf8);
  }
}
```

Código 13. Descifrado Strings

En el ejemplo se leen las propiedades del paciente y se sobrescriben los datos encriptados por los datos descifrados.

2. Si los datos son un fichero CSV
 - 2.1. Se obtiene el fichero almacenado en Firebase mediante su url y se convierte a Blob, a continuación, se lee el fichero usando la función *readAsText (fichero)*, que devuelve el contenido del archivo como una cadena de texto.
 - 2.2. Se consulta la clave de cifrado almacenada en la cookie.
 - 2.3. Se aplica el algoritmo de encriptación AES con la clave obtenida.
 - 2.4. Se convierte a String codificado en UTF-8.
 - 2.5. En este momento ya se puede leer el contenido del fichero CSV original, bien para previsualizar el fichero o para visualizar los esqueletos.

```
pathReference.child(grabacion).getDownloadURL().then(function (url) {
  fetch(url)
    .then(res => res.blob()) // Gets the response and returns it as a blob
    .then(blob => {
      var reader = new FileReader();
      reader.onload = function () {
        let ficheroDescifrado = CryptoJS.AES.decrypt(reader.result,
          getCookie('clave')).toString(CryptoJS.enc.Utf8);
      };
      reader.readAsText(blob);
    });
});
```

Código 14. Descifrado CSV/JSON

3. Si los datos son un vídeo

- 3.1. Se obtiene el archivo de vídeo mediante el objeto JavaScript *XMLHttpRequest*, este objeto permite obtener cualquier tipo de dato de una URL de forma asíncrona, sin tener que recargar la página completa.
- 3.2. Se consulta la clave de cifrado almacenada en la cookie.
- 3.3. Se aplica el algoritmo de desencriptado AES con la clave obtenida.
- 3.4. Se convierte a String codificado en Latin-1.
- 3.5. En este momento ya se puede reproducir el vídeo. Bastaría con añadir el String obtenido como atributo *href* a una etiqueta vídeo HTML.

```
pathReference.child(grabacion.vídeo).getDownloadURL().then(function (url) {  
  let xhr = new XMLHttpRequest();  
  xhr.onload = function (event) {  
    let videoEncriptado = xhr.response;  
    let videoDesencriptado = CryptoJS.AES.decrypt(videoEncriptado,  
      getCookie('clave')).toString(CryptoJS.enc.Latin1);  
    video.setAttribute('href', videoDesencriptado);  
    addSourceToVideo(video, videoDesencriptado);  
  };  
  xhr.open('GET', url);  
  xhr.send();  
});
```

Código 15. Desencriptado vídeo

7.6. Visualización de las grabaciones

Una de las funcionalidades más importantes de la aplicación es la visualización de las grabaciones de marcha de los pacientes. Consiste en visualizar el esqueleto del paciente obtenido con Kinect de forma sincronizada con el vídeo correspondiente a esa grabación para poder analizar los movimientos del paciente. También permite controlar la reproducción, permitiendo pausar, reproducir, retroceder y avanzar, además de la posibilidad de añadir notas a la reproducción. En este apartado se explican cuáles han sido los pasos para desarrollar esta funcionalidad.

7.6.1. Herramientas utilizadas

La principal herramienta utilizada para implementar la visualización ha sido el *<canvas>* de HTML5, un elemento HTML que puede ser usado para dibujar gráficos usando scripts (normalmente JavaScript). Este puede, por ejemplo, ser usado para dibujar gráficos, realizar composición de fotos o animaciones [26]. Además, se ha usado la librería Papaparse [12] para poder manejar el fichero CSV de la grabación de forma sencilla. La reproducción, al igual que el formulario de inserción de notas, se muestra en el elemento *mdDialog* proporcionado por la librería AngularJS Material [9]. Estas herramientas se explican con más detalle en el capítulo [4. Tecnologías](#), dedicado a las tecnologías utilizadas durante el desarrollo del proyecto.

7.6.2. Dibujado del esqueleto

El primer paso para llevar a cabo la visualización de los esqueletos es cargar el fichero CSV que contiene la posición de cada una de las articulaciones en cada instante de tiempo. Una vez cargado el archivo, este es tratado por Papapase mediante la función `parse()`. A esta función se le pasa el texto del fichero como parámetro y devuelve un objeto que contiene como principal propiedad un array en el que cada una de las filas del fichero está representada en forma de array. Cada array contiene la información del esqueleto para un instante concreto.

Una vez que toda la información del fichero está contenida en el array, la idea es ir recorriendo cada una de las filas del fichero y dibujando en el canvas la posición del esqueleto en ese instante. Puesto que es necesario actualizar la imagen del canvas alrededor de 30 veces por segundo (esto se debe a que Kinect registra la posición del usuario con esa frecuencia) se opta por emplear la función `setInterval` [27] de JavaScript, que permite llamar a una función o evaluar una expresión a intervalos específicos (en milisegundos).

`setInterval` llamará a la función que dibuja el esqueleto unas 30 veces por segundo. Esta función deberá leer la siguiente línea del fichero (el siguiente array del array, pues se está trabajando con el objeto devuelto por `Papaparse`), comprobando primero que no se haya alcanzado el final del fichero, e ir analizando cada una de las articulaciones. Para cada articulación se comprueba si existe información sobre ella, en caso de no existir se le asignarán las coordenadas (0, 0), después se consulta cuáles son las articulaciones que dependen de ella, por ejemplo, la cadera derecha, la cadera izquierda y la columna vertebral dependen de la base de la columna vertebral. Esto se debe a que la base de la columna vertebral se toma como la raíz de la jerarquía.

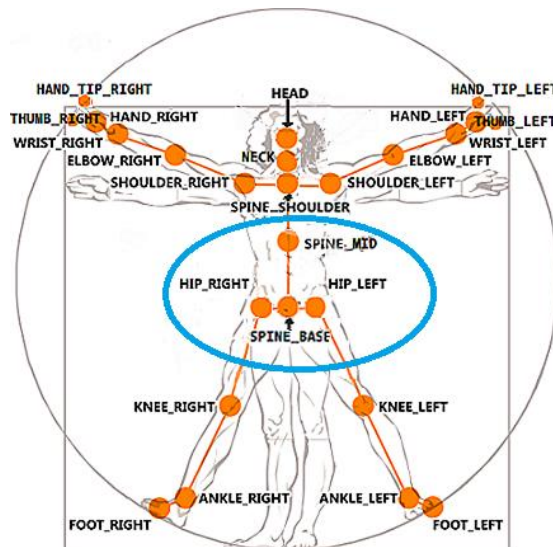


Ilustración 34. Dependencias columna vertebral

Estas dependencias se consultan para dibujar una línea entre la articulación “padre” y sus dependientes. Para poder realizar estas consultas se ha creado un vector que representa la jerarquía de las articulaciones. Cada posición del vector, la posición coincide con la posición en la que la articulación aparece en el CSV, contiene las articulaciones que dependen directamente de ella en la jerarquía. Estas aparecen como enteros que representan a las articulaciones, coincidiendo otra vez con la posición en la que aparecen en el fichero CSV.

A continuación, se muestran las primeras posiciones del vector con el objetivo de esclarecer la estructura:

```
[[1, 12, 16], [20], [3], [], ...]
```

Código 16. Vector jerarquía de articulaciones

En el vector del ejemplo la primera posición representa la base de la columna, y tal y como se ha mencionado anteriormente, de ella dependen tres articulaciones, que en este caso son las identificadas con los números 1, 12 y 16. Lo mismo ocurre con el resto de articulaciones.

Tras haber pintado las líneas entre la articulación y sus dependientes, el siguiente paso es pintar la articulación. Para ello se comprueba si la articulación ha sido inferida, en caso de no haber sido inferida se representa mediante un círculo verde, en caso contrario mediante uno rojo. El esqueleto se dibuja en dos canvases, en uno se tienen en cuenta las posiciones X e Y, y en el otro Z e Y, de esta forma se consigue representar la visión frontal y lateral del esqueleto.

En el siguiente fragmento de pseudocódigo se refleja el proceso seguido para dibujar los esqueletos:

```
fichero = read(file);
ficheroParseado = Papa.parse(fichero);
linea = 0;
temporizador = setInterval(pintarEsqueleto, 33);
function pintarEsqueleto () {
    if (linea < ficheroParseado.length){
        for (articulacion in ficheroParseado[linea]){
            pintarLineasArticulacionesDependientes();
            if (articulacion is inferida){
                pintarArticulacionRoja();
            }else{
                pintarArticulacionVerde();
            }
        }
        linea = linea + 1;
    }else{
        parar temporizador;
    }
}
```

Código 17. Pseudocódigo pintado esqueletos

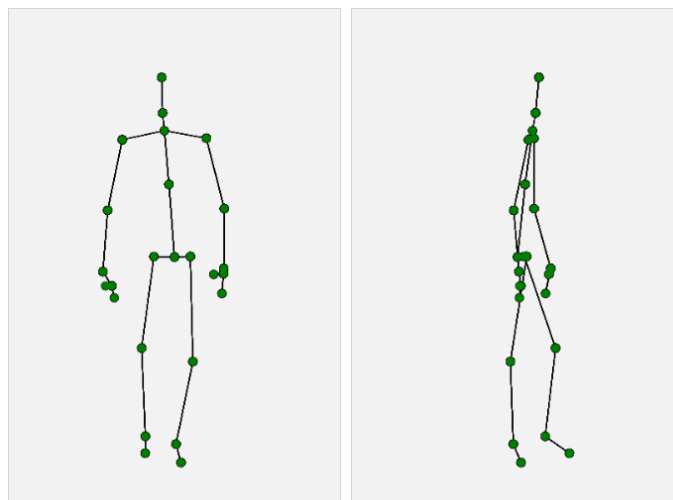


Ilustración 35. Visualización frontal y lateral del esqueleto

7.6.3. Visualización sincronizada de vídeo y esqueleto

El siguiente paso en la implementación de la visualización es conseguir que el vídeo y el esqueleto se reproduzcan sincronizados. Para ello, el principal objetivo es lograr que el esqueleto se actualice a la misma velocidad que el vídeo avanza, es decir, que ambos estén siempre en el mismo punto temporal. En un principio el esqueleto se actualizará 30 veces por segundo, tal y como se había establecido en su visualización. Como Kinect no asegura que conseguirá información del paciente 30 veces por segundo es necesario realizar comprobaciones extra con el objetivo de ajustar la línea del fichero que debe dibujar en cada momento.

Estas comprobaciones se realizan en base a la reproducción del vídeo, cada vez que este avance un segundo se activará el evento *ontimeupdate*. Este evento ejecuta una función en la que se comprueba si el vídeo y el esqueleto están sincronizados. Para ello se compara el minuto de reproducción actual del vídeo con el tiempo al que corresponde la línea actual del esqueleto (el tiempo se encuentra en la primera columna de cada línea del fichero).

Si ambos tiempos son iguales o muy cercanos, la reproducción continua como hasta ese momento. En caso contrario, se busca la línea del fichero que tenga la marca de tiempo más cercana al tiempo actual del vídeo y se actualiza el número de la línea a visualizar.

Puesto que a la hora de registrar una grabación no es necesario añadir un fichero de vídeo, únicamente el fichero CSV, será necesario distinguir ambos casos a la hora de elegir la estrategia de sincronización. Si se dispone tanto de vídeo como de fichero CSV, la sincronización se controla de la forma explicada. En caso de no disponer del fichero de vídeo la visualización se lleva a cabo como se ha explicado en el apartado anterior, actualizando el canvas 30 veces por segundo.

También es necesario tener en cuenta qué ocurre si la grabación y la animación del esqueleto no tienen la misma duración. La casuística es la siguiente:

- Si el vídeo es más largo que el esqueleto, la reproducción la controlará el vídeo, una vez acabe el esqueleto este seguirá reproduciéndose hasta finalizar.
- Si el vídeo es más corto que el esqueleto, la reproducción la controlará el vídeo hasta que finalice. Una vez finalizado el vídeo, el esqueleto se actualizará de forma independiente 30 veces por segundo, tal y como se ha definido en el apartado anterior.

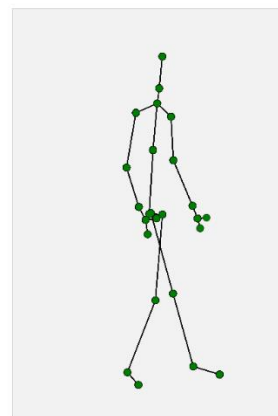
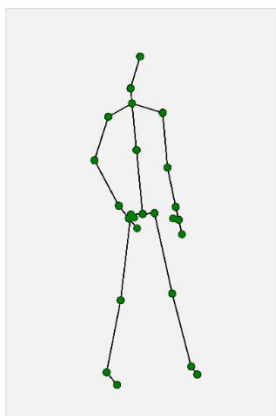


Ilustración 36. Visualización sincronizada de vídeo y esqueleto

7.6.4. Herramientas de reproducción

El último paso para completar el visualizador es desarrollar las herramientas de reproducción. Por un lado, se crearán unos controles compuestos por una barra de progreso, un botón de pausa/reproducción y un controlador de volumen. Mediante estos controles se posibilitará el pausado, el avance o retroceso a cualquier punto y el control del nivel del volumen de la grabación. Por otro lado, se posibilitará al usuario elegir qué quiere ver, el vídeo, la vista frontal del esqueleto o la vista lateral. Pudiendo alternar entre ellas, o ver todas a la vez. También se incluye la posibilidad de añadir notas pulsando en cualquier punto del esqueleto. Las notas incluidas se pueden gestionar desde la vista *Grabaciones*.

Para el pausado de la reproducción se ha declarado una variable booleana que indica si la reproducción está o no pausada. También se ha definido un evento que se activa cuando el usuario pulsa sobre el botón de pausa/reproducción. Este evento otorga a la variable el valor opuesto al que había en el momento de ser pulsado, y pausa o reproduce el vídeo en base al valor de la variable. Además, al inicio de la función que gestiona la visualización del esqueleto se comprueba el valor de la variable definida. De esta forma la visualización del esqueleto solo avanzará cuando la variable indique que la reproducción no está pausada.

En cuanto al control de volumen, solamente afecta al vídeo, ya que el esqueleto no dispone de ningún tipo de audio. Este controlador permite tanto silenciar el vídeo como controlar su nivel. Esto se consigue definiendo un evento para el botón de silenciado y otro para la barra de control de nivel.

El control más complejo es el de la barra de progreso, con la cual se podrá establecer qué punto de la grabación se quiere reproducir. La barra está implementada mediante un input de tipo rango que activa un evento cuando se desplaza o cuando se cambia su valor. El valor del input es actualizado a la par que la reproducción avanza. Cuando se lanza el evento, es necesario analizar el desplazamiento, ya que este puede haber sido actualizado debido al avance natural de la reproducción o debido a que el usuario lo ha desplazado. Si el desplazamiento ha sido por parte del usuario será necesario actualizar el minuto de reproducción del vídeo y la línea de lectura actual del visualizador del esqueleto.

El funcionamiento de la barra de progreso depende tanto de la disponibilidad del fichero de vídeo y del CSV, como de cuál de ellos tiene una mayor duración en caso de contar con ambos. La casuística es la siguiente:

- Tanto el fichero de vídeo como el fichero CSV están disponibles:
 - Si el vídeo es más largo que el esqueleto, durante toda la reproducción la barra de progreso se actualiza a la par que el vídeo avanza.
 - Si el vídeo es más corto que el esqueleto, la barra de progreso se actualiza a la par que el vídeo mientras esté dentro de los límites del vídeo, si no, se actualiza en función del tiempo asociado a la línea del esqueleto que se esté visualizando en ese momento.
- Solo el fichero CSV está disponible:
 - La barra de progreso se actualiza en base al tiempo de reproducción que indique la línea actual del fichero.

A continuación, se muestran imágenes de la reproducción junto con las herramientas. También se incluyen imágenes de la creación de notas y su posterior visualización.

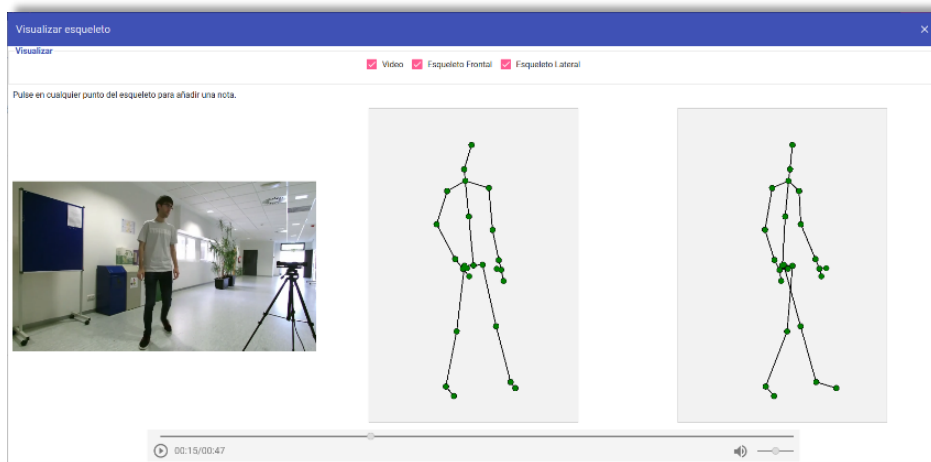


Ilustración 37. Visualización grabación con herramientas

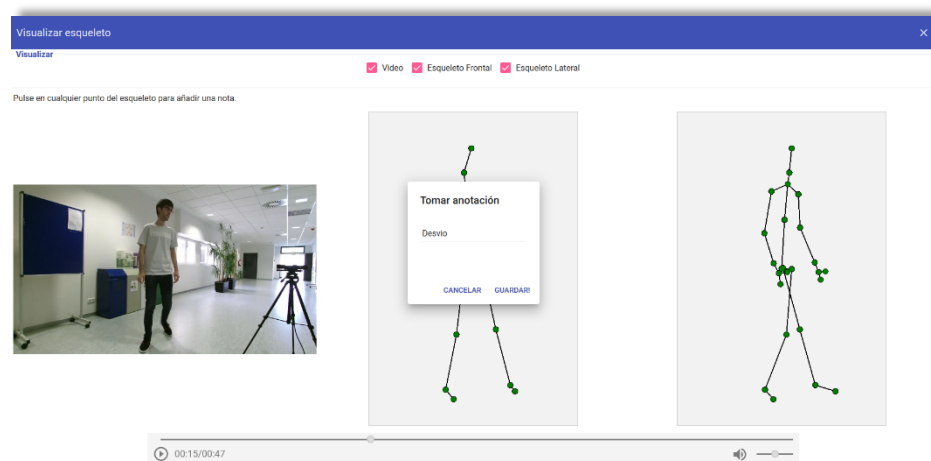


Ilustración 38. Visualización grabación. Añadir nota

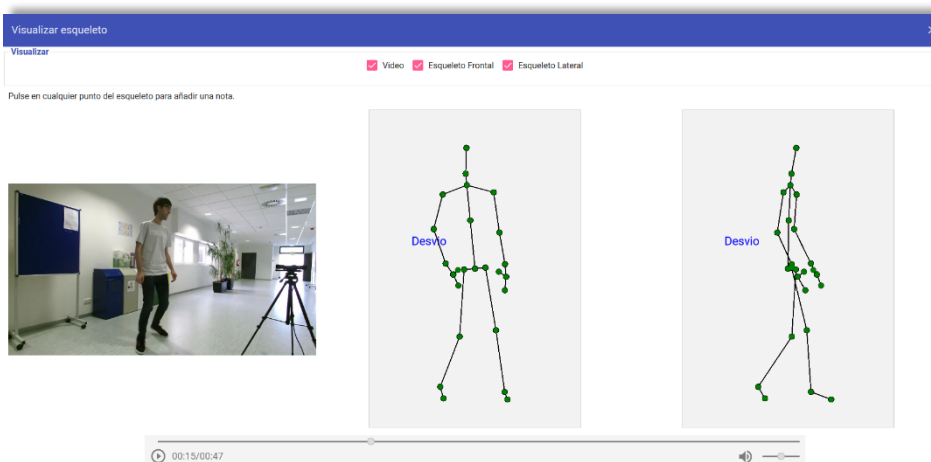


Ilustración 39. Visualización grabación. Visualizar nota

7.7. Dificultades durante el desarrollo

En este apartado se describen las dificultades encontradas a lo largo del desarrollo del proyecto y cómo se han superado.

La mayor dificultad del proyecto surgió en torno a la implementación de la visualización de los esqueletos de las grabaciones. Al comienzo del proyecto se estudiaron las alternativas disponibles para llevar a cabo el dibujo del esqueleto. Por un lado, estaba la forma más simple, dibujar el esqueleto de forma bidimensional en un Canvas de HTML5, de forma similar a como se había implementado en la aplicación anterior. Por otro lado, se encontró la herramienta three.js [28], una biblioteca escrita en JavaScript para crear y mostrar gráficos animados en 3D desde el navegador Web. Más concretamente se encontró este ejemplo [29], disponible en su página web, dónde se puede observar a una persona caminando, permitiendo acercar y alejar la cámara del esqueleto al antojo del usuario e incluso rotar la cámara alrededor del mismo. También permite limitar el visionado únicamente a su esqueleto, tal y como se pretendía implementar en la visualización. Además, el ejemplo permite pausar la reproducción, acelerarla y modificar varias características más.

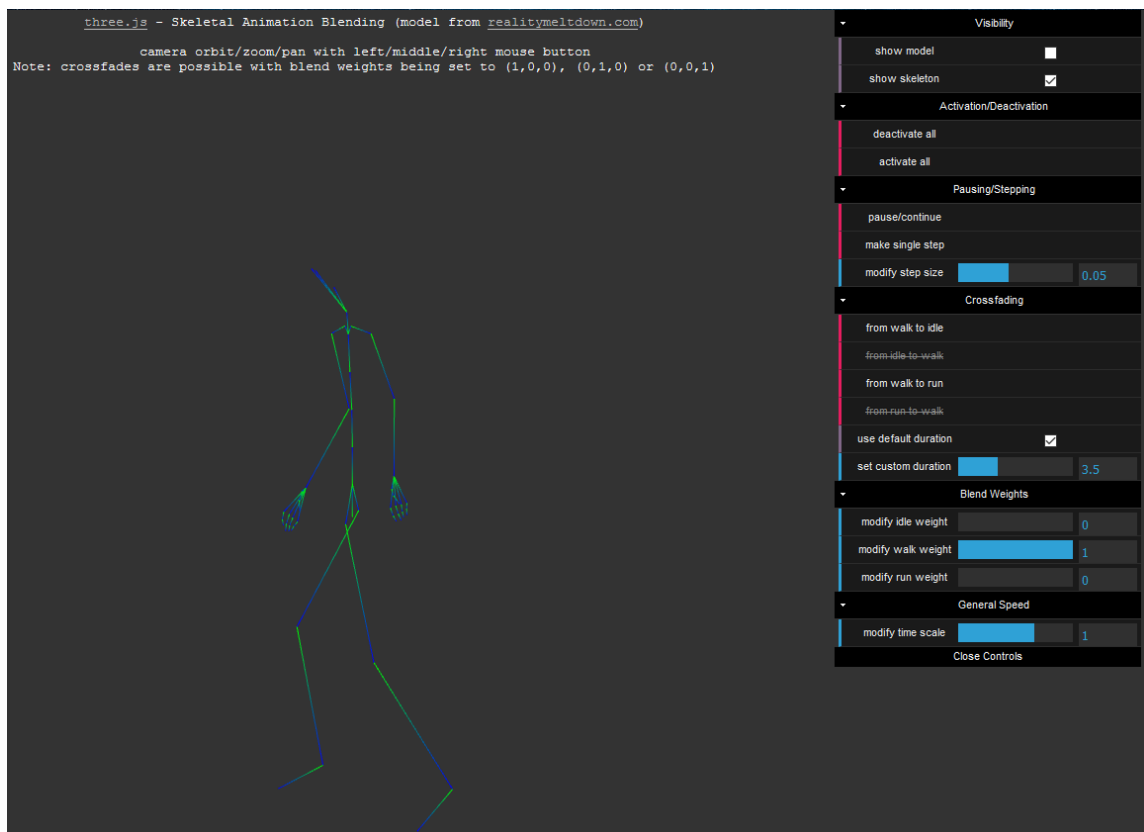


Ilustración 40. Visualización esqueleto con ThreeJS

Tras ver el gran potencial de esta biblioteca se optó por llevar a cabo el desarrollo de la visualización del esqueleto con ella. Las primeras dificultades surgieron con el formato del fichero desde el que se obtenía la información del esqueleto a visualizar. Este fichero era un JSON que se dividía en dos partes. La primera de ellas contenía la definición de cada una de las articulaciones del esqueleto seguido de la jerarquía que formaban entre ellas.

La segunda parte del fichero se componía de las animaciones del esqueleto, dónde se definía para cada unidad de tiempo la posición de cada una de las articulaciones. Además de estas dos grandes secciones había otras dedicadas a el modelo que se muestra en el ejemplo y otras configuraciones que no eran relevantes para el uso que se quería hacer en el proyecto. El formato de este fichero no está explicado ni en el ejemplo ni en la documentación de la librería, por lo que el significado de cada parte de la estructura se intuyó en base a pruebas e investigación.

Debido a que la librería espera como entrada un JSON y Kinect devuelve un CSV, fue necesario crear un conversor que transformase el fichero CSV al formato JSON descrito. El formato que debe tener el fichero JSON para que la visualización se realice correctamente no está descrito en la documentación oficial de la librería. La documentación de la misma es muy superficial y escasa la mayoría de las veces. Una vez implementado el conversor se probó la visualización con el fichero convertido, pero la visualización del esqueleto no fue la esperada. Mientras que la jerarquía de huesos parecía que se visualizaba correctamente, había alguna articulación que no se visualizaba en su posición correcta, además algunas de ellas se visualizaban desproporcionadas. En la siguiente imagen se puede comparar cómo se visualizaba con three.js y como se debería estar visualizando. Para comprobar cómo se debía visualizar se tomaron las posiciones del esqueleto para ese instante concreto y se dibujaron con GeoGebra [30], un software matemático compuesto por un procesador geométrico y un procesador algebraico.

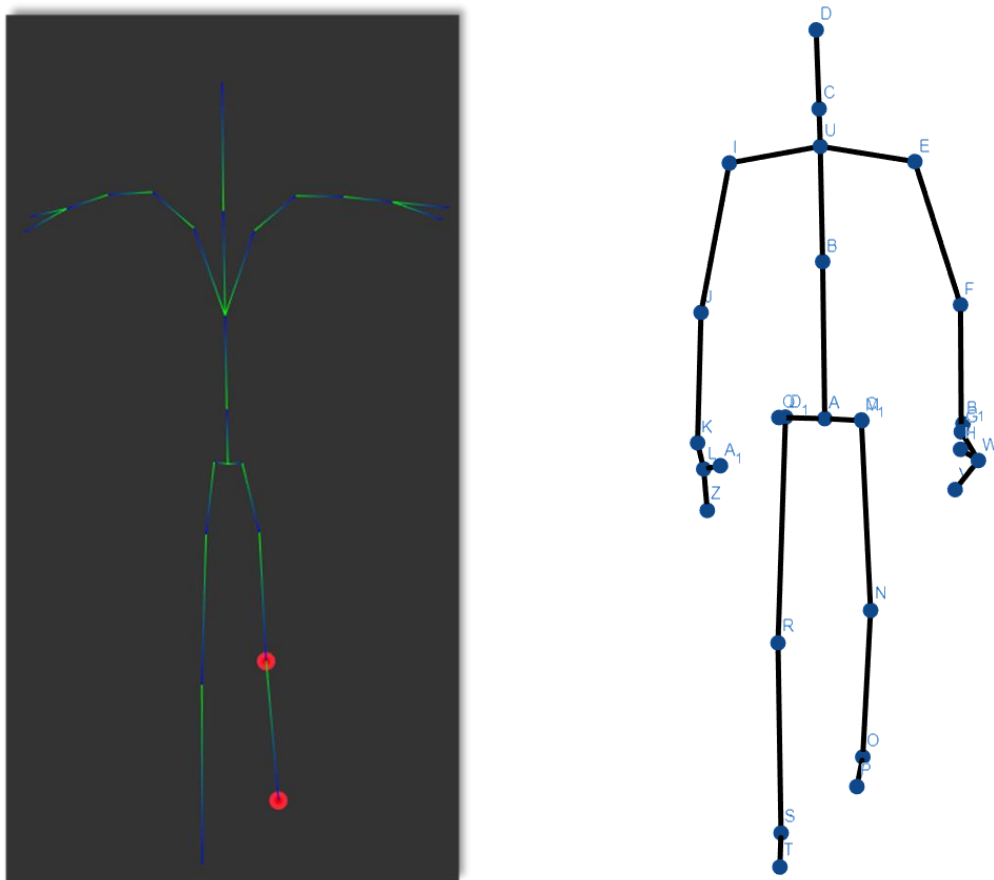


Ilustración 41. Visualización en ThreeJS (izquierda) con errores y esqueleto en GeoGebra (derecha)

Tras llevar a cabo diversas pruebas e investigar el problema de visualización en profundidad, e incluso preguntar en foros oficiales fue imposible solventarlo. Ni siquiera los desarrolladores de la biblioteca, que contestaron a la consulta planteada en el foro, fueron capaces de dar una respuesta a cuál era el origen del problema.

Por la imposibilidad de implementar la visualización con three.js y tras haberle dedicado gran cantidad de horas y alrededor de dos iteraciones, se decidió abandonar la herramienta y optar por desarrollarlo con el Canvas de HTML, de forma similar a como se había implementado en la aplicación anterior. La implementación del visualizador se define en detalle en la sección [7.6 Visualización de las grabaciones](#).

Otra de las dificultades está relacionada con la encriptación, más concretamente con la encriptación de los ficheros de vídeo. El desarrollo de la encriptación de estos ficheros fue especialmente costosa a diferencia de la encriptación de los ficheros CSV y JSON. La gran diferencia entre ambos ficheros es que los dos últimos son ficheros de texto plano, mientras que el fichero de vídeo tiene un formato más complejo, además de metadatos necesarios para su reproducción. Es por ello que ambos tipos de ficheros no pueden ser tratados de la misma forma a la hora del encriptado y desencriptado. Al inicio del desarrollo de la encriptación, no se tuvieron en cuenta estas diferencias y todos los ficheros se encriptaban igual. Mientras que los ficheros de texto se visualizaban correctamente tras ser desencriptados, los de vídeo no se reproducían y un error indicaba que el archivo estaba corrupto. Fueron necesarias cantidad de pruebas para encontrar la forma correcta de leer el fichero de vídeo para que toda la información necesaria para reproducirse se mantuviese una vez encriptado, y que al desencriptarlo se pudiese reproducir. La solución también se encontró gracias a [31], una herramienta para encriptar ficheros implementada en JavaScript usando la librería CryptoJS, que, aunque limitado a ficheros de máximo 1MB serviría de inspiración para encontrar la solución al problema.

Durante el desarrollo se encontraron más dificultades que no tienen la relevancia de las citadas hasta ahora. Estos problemas se solventaron a base de investigación y pruebas, o consultado y preguntando en foros como StackOverflow [32].

8. Pruebas

En este capítulo se presentan las pruebas realizadas para asegurar que la aplicación web funciona correctamente y cumple con los requisitos establecidos. Es necesario asegurar que todas las funcionalidades funcionan tal y como se había previsto, sin dar lugar a errores o fallos que impidan una experiencia satisfactoria a los usuarios. Las pruebas se han siguiendo las pautas definidas en las Historias de Usuario.

Las pruebas que se presentan en este capítulo se han llevado a cabo tanto a lo largo del proyecto, verificando al final de cada Sprint las funcionalidades desarrolladas durante el mismo, como al final del proyecto, con el objetivo de asegurar que todo funciona de la forma esperada.

Todas las pruebas se han efectuado tanto en un servidor local como en Firebase. Para ello se ha empleado el enfoque de caja negra, donde la aplicación es estudiada desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno [33].

A continuación, se presentan varias de las pruebas realizadas a cada una de las Historias de Usuario que forman parte del Product Backlog del proyecto. Para cada test definido en las Historias de Usuario se han realizado las pruebas necesarias para verificar que funciona tal y como estaba previsto. El resto de pruebas se encuentran en el [Anexo C. Pruebas](#).

HU6. Visualización de esqueletos en sincronía con el vídeo

Como cliente quiero visualizar los vídeos de las grabaciones desde la aplicación web.

1. Comprobar que el vídeo se ve correctamente sincronizado con el esqueleto.

Prueba 1: El usuario intenta visualizar el esqueleto y el vídeo de una grabación.

Entrada: El usuario pulsa sobre el icono de  la columna *Grabaciones*.

Salida esperada: La reproducción del esqueleto y el vídeo avanza sincronizada.

Salida real: Salida esperada.

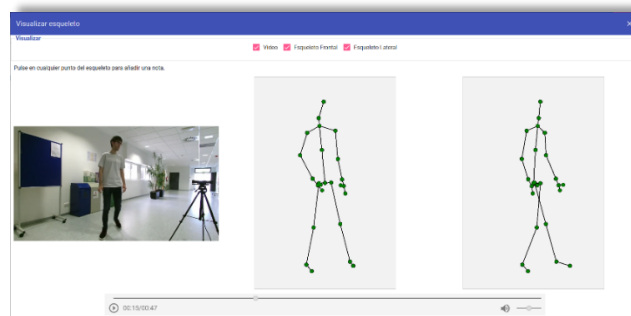


Ilustración 42. Test HU6.1 Prueba 1

2. Visualizar solamente el vídeo.

Prueba 1: El usuario intenta reproducir únicamente el vídeo de una grabación.

Entrada: El usuario pulsa sobre el icono de  la columna *Grabaciones* y seleccionar el vídeo.

Salida esperada: La reproducción del vídeo de la grabación.

Salida real: Salida esperada.

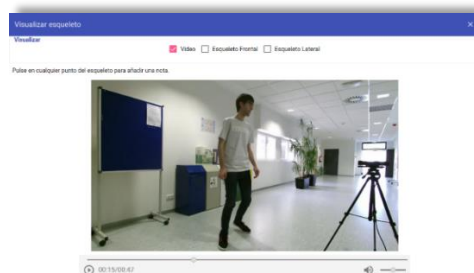


Ilustración 43. Test HU6.2 Prueba 1

3. Visualizar solamente la vista frontal del esqueleto.

Prueba 1: El usuario intenta reproducir únicamente la vista frontal del esqueleto de una grabación.

Entrada: El usuario pulsa sobre el icono de ▶ la columna *Grabaciones* y deselecciona la vista lateral del esqueleto y el vídeo.

Salida esperada: La reproducción de la vista frontal de la grabación.

Salida real: Salida esperada.

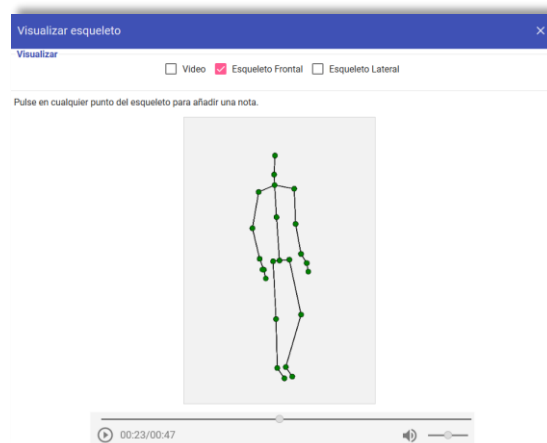


Ilustración 44. Test HU6.3 Prueba 1

4. Visualizar solamente la vista lateral del esqueleto.

Prueba 1: El usuario intenta reproducir únicamente la vista lateral del esqueleto de una grabación.

Entrada: El usuario pulsa sobre el icono de ▶ la columna *Grabaciones* y deselecciona la vista frontal del esqueleto y el vídeo.

Salida esperada: La reproducción de la vista lateral de la grabación.

Salida real: Salida esperada.

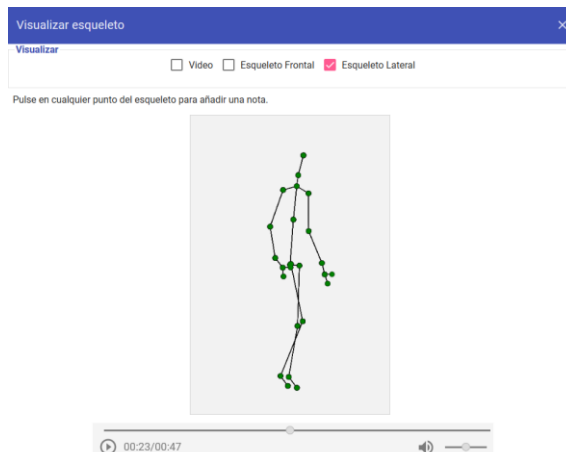


Ilustración 45. Test HU6.4 Prueba 1

5. Visualizar tanto el vídeo como el esqueleto.

Prueba 1: El usuario intenta reproducir tanto la vista frontal y lateral del esqueleto como el vídeo de una grabación.

Entrada: El usuario pulsa sobre el icono de  la columna *Grabaciones* y selecciona la vista frontal y lateral del esqueleto y el vídeo.

Salida esperada: La reproducción de la vista frontal y lateral de la grabación y del vídeo.

Salida real: Salida esperada.

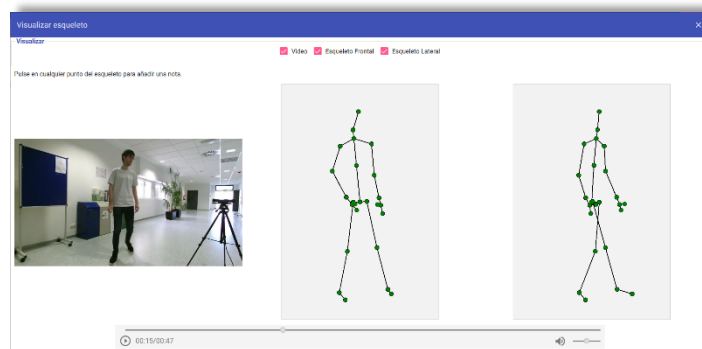


Ilustración 46. Test HU6.5 Prueba 1

HU8. Encriptación de la información

Como cliente quiero que tanto los ficheros de datos y vídeos como la información se almacene cifrada para cumplir con las indicaciones de la LOPD. Además, quiero evitar que el administrador de Firebase pueda visualizar los datos de los pacientes y de las grabaciones.

1. Comprobar que los ficheros csv se almacenan encriptados y que no se puede interpretar su contenido sin ser descifrado previamente.

Prueba 1: El usuario crea una grabación con un fichero CSV. El administrador de Firebase descarga el fichero e intenta leerlo.

Entrada: El fichero CSV encriptado de una grabación.

Salida esperada: El fichero CSV es ilegible.

Salida real: Salida esperada.

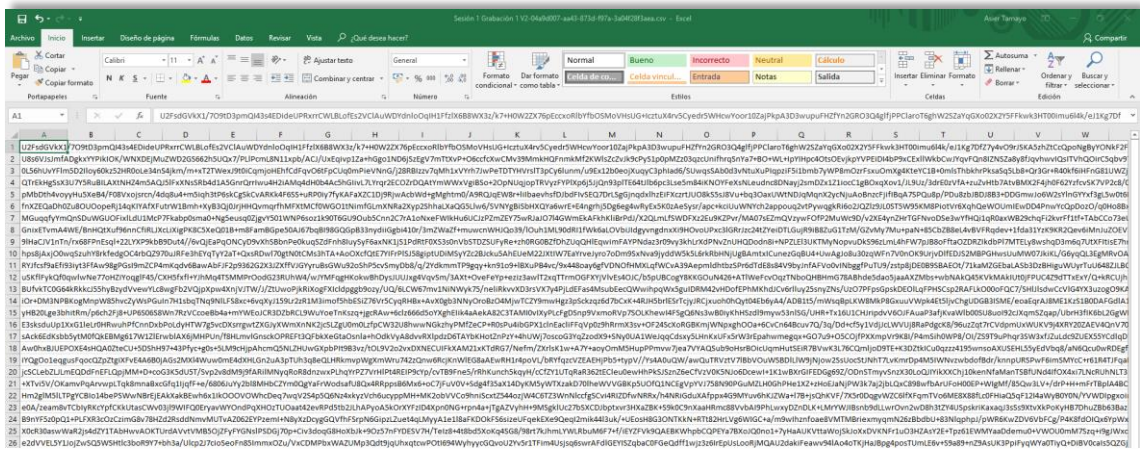
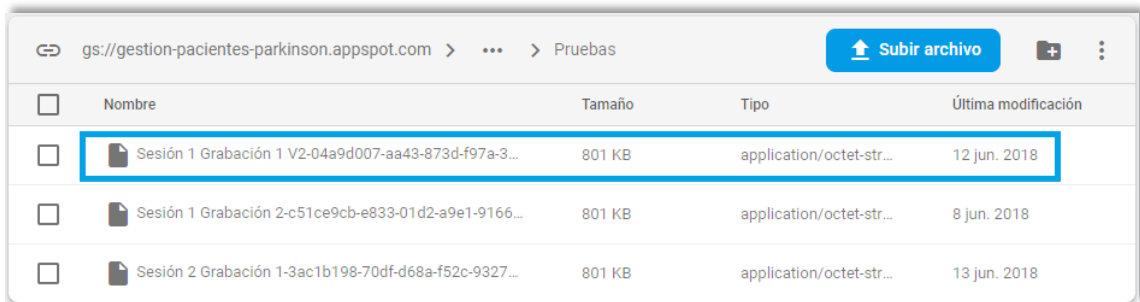


Ilustración 47. Test HU8.1 Prueba 1

2. Comprobar que los ficheros de vídeo se almacenan encriptados y que no se puede interpretar su contenido sin ser descifrado previamente.

Prueba 1: El usuario crea una grabación con un fichero de vídeo. El administrador de Firebase descarga el fichero e intenta reproducirlo.

Entrada: El fichero de vídeo encriptado de una grabación.

Salida esperada: El fichero de vídeo es irreproducible e ilegible.

Salida real: Salida esperada.

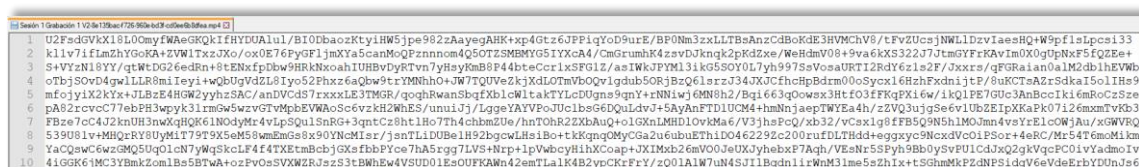
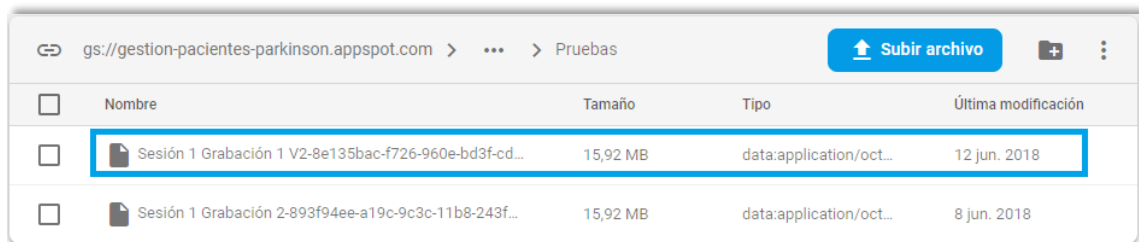


Ilustración 48. Test HUB.2 Prueba 1

3. Comprobar que la información de pacientes y grabaciones se almacena encriptada y que no se puede interpretar su contenido sin ser descifrada previamente.

Prueba 1: El usuario crea un paciente. El administrador de Firebase intenta ver la información del paciente registrado.

Entrada: Un nuevo paciente.

Salida esperada: La información del paciente es ilegible.

Salida real: Salida esperada.

```
-LEJYc2R_FSgUEqOV0kU
  altura: "U2FsdGVkX1+Q/JZ0XC0aB3L1veig+ywAiY80gapvT
  clasificacion: "U2FsdGVkX19ANYezveWJjIoK0X0j9QiAE0t1bUe3EI
  extra
    Telefono: "U2FsdGVkX18yYxoGw11z+8yGsBBkCrKMGJK/8wwnF
  fechaNacimiento: "U2FsdGVkX1+kDivKf0g6heRGMqkK4jxCy6RzEoKXP
  grabaciones
    -LETbE63e4Jk4YGs0rAB: true
  id: "U2FsdGVkX18nc5KExXz31WPhZhwMfb67Pv5PuHgXT
  inicioEnfermedad: "U2FsdGVkX19mtLyvkKyvFHAzqbDfS8X9uQF8+ZLo8
  sexo: "U2FsdGVkX18UjRP4AfXGfc0iaAJQqJF8MefRpy5N8
```

Ilustración 49. Test HU8.3 Prueba 1

Prueba 2: El usuario crea una grabación. El administrador de Firebase intenta ver la información de la grabación registrada.

Entrada: Una nueva grabación.

Salida esperada: La información de la grabación es ilegible.

Salida real: Salida esperada.

```
-LETbE63e4Jk4YGs0rAB
  edadGrabacion: "U2FsdGVkX1//XU+bLs5X/LTZx0MIzSRRB0BUReqWf
  edadPaciente: "U2FsdGVkX184R5Tk6VkB1jyoSLUxtBK5ExYB.JopmC
  fechaGrabacion: "U2FsdGVkX18Ykz1f4JggRzk+C/XhsH0v60F0637Qv
  grabacion: "Sesión 1 Grabación 1 V2-04a9d007-aa43-873d-f97e
  id: "U2FsdGVkX1/1kVE001zUnFLdZtxwzqLqnKdPt/qADbtE9Df
  lugar: "U2FsdGVkX1/5pxVh0jZoCk7Xi5muSMwLNt10fzPs7
  notasVideo
    -LEs7bolvjhplvWwH_t5
      segundo: "U2FsdGVkX1+qIhVJK2t7ssskctyZvgwsxdyoA08189bbRR7
      texto: "U2FsdGVkX1/1fAac4K1Yq9VgLfuyXKNhNXgYU1aPv
      x: "U2FsdGVkX1+7Fr2oo/GYcMkCo0uugMAvfgiid9gWM
      y: "U2FsdGVkX1+8Zcty33A7QkUDzCBo76PnxTzTDYUgamSsH1t
    -LEs7cluwM_EYoNOeAWi
      segundo: "U2FsdGVkX1/F5jadY5cKaTYBoGhalqyUWpwPsoxJK7fxVOF
      texto: "U2FsdGVkX18GzDVkXpaAnWm6X0DK3TaweJY+0Le8G
      x: "U2FsdGVkX19F4JJvdNucwU2PpnTbuZ1mr9JEjrXoG
      y: "U2FsdGVkX1/mpMNBzzTAC5z9g6/gbopVZvixPJKreKRplsc
  paciente: "U2FsdGVkX18nc5KExXz31WPhZhwMfb67Pv5PuHgXT
  pacienteKey: "U2FsdGVkX1+R8kSb0J17TpLLV+jXs7k8CvGCyH7Rf4L9mRF
  video: "Sesión 1 Grabación 1 V2-8e135bac-f726-960e-bd3f
```

Ilustración 50. Test HU8.3 Prueba 2

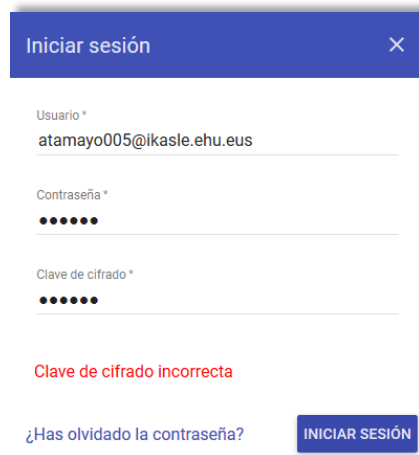
4. Comprobar que no deja acceder a la aplicación sin introducir la clave de cifrado correcta

Prueba 1: El usuario rellena el formulario de inicio de sesión completamente, pero introduce una clave de cifrado incorrecta.

Entrada: Usuario, contraseña y clave de cifrado. Clave de cifrado incorrecta.

Salida esperada: Mensaje de error indicando que la clave de cifrado es incorrecta.

Salida real: Salida esperada.



The screenshot shows a login form titled "Iniciar sesión" with a close button (X). The form contains three input fields: "Usuario*" with the value "atamayo005@ikasle.ehu.eus", "Contraseña*" with masked characters, and "Clave de cifrado*" with masked characters. Below the fields, a red error message reads "Clave de cifrado incorrecta". At the bottom, there is a link "¿Has olvidado la contraseña?" and a blue button labeled "INICIAR SESIÓN".

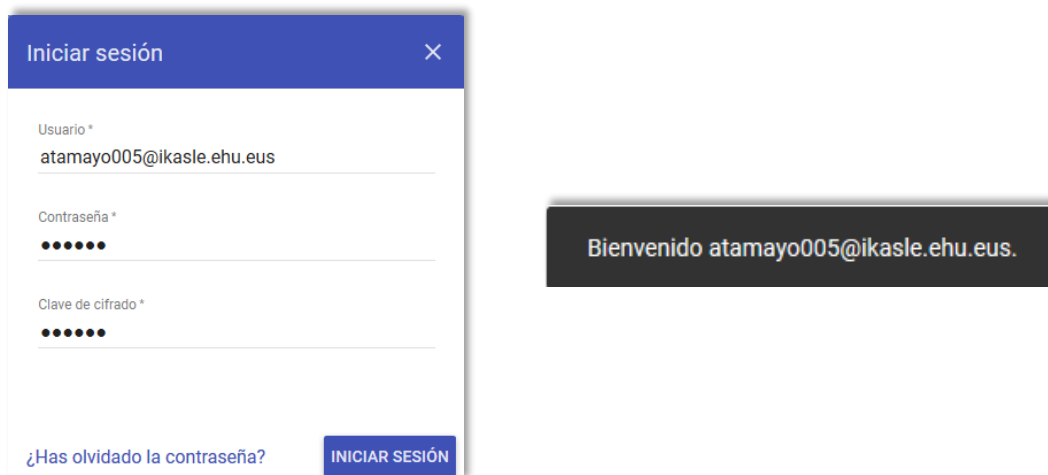
Ilustración 51. Test HU8.4 Prueba 1

Prueba 2: El usuario rellena el formulario de inicio de sesión completo de forma correcta.

Entrada: Usuario, contraseña y clave de cifrado.

Salida esperada: Sesión iniciada.

Salida real: Salida esperada.



The screenshot shows the same login form as in Illustration 51, but with the "INICIAR SESIÓN" button highlighted. To the right of the form, a dark grey banner displays the message "Bienvenido atamayo005@ikasle.ehu.eus.".

Ilustración 52. Test HU8.4 Prueba 2

9. Seguimiento del proyecto

Este capítulo está dedicado al seguimiento del proyecto. En él se expone el diagrama de Gantt que representa lo trabajado durante las distintas iteraciones del proyecto. Además, se muestran las dedicaciones a las distintas áreas del proyecto.

9.1. Diagrama de Gantt final

A continuación, se muestra el diagrama de Gantt que se ha obtenido tras finalizar el desarrollo del proyecto. A diferencia del [diagrama de Gantt original](#), el de la planificación inicial, en este se muestran las tareas desarrolladas en cada una de las iteraciones.

Puesto que el proyecto se ha desarrollado siguiendo la metodología Scrum, a la hora de realizar la planificación inicial no se sabía con detalle qué se haría en cada iteración, pues en base a las reuniones de cierre de las iteraciones se definen que tareas se realizarán en la siguiente iteración. Como se puede observar en el diagrama, las fechas de las iteraciones no se han cumplido con exactitud, ni tampoco la duración de cada una de ellas. Sin embargo, el número de iteraciones se ha mantenido intacto.

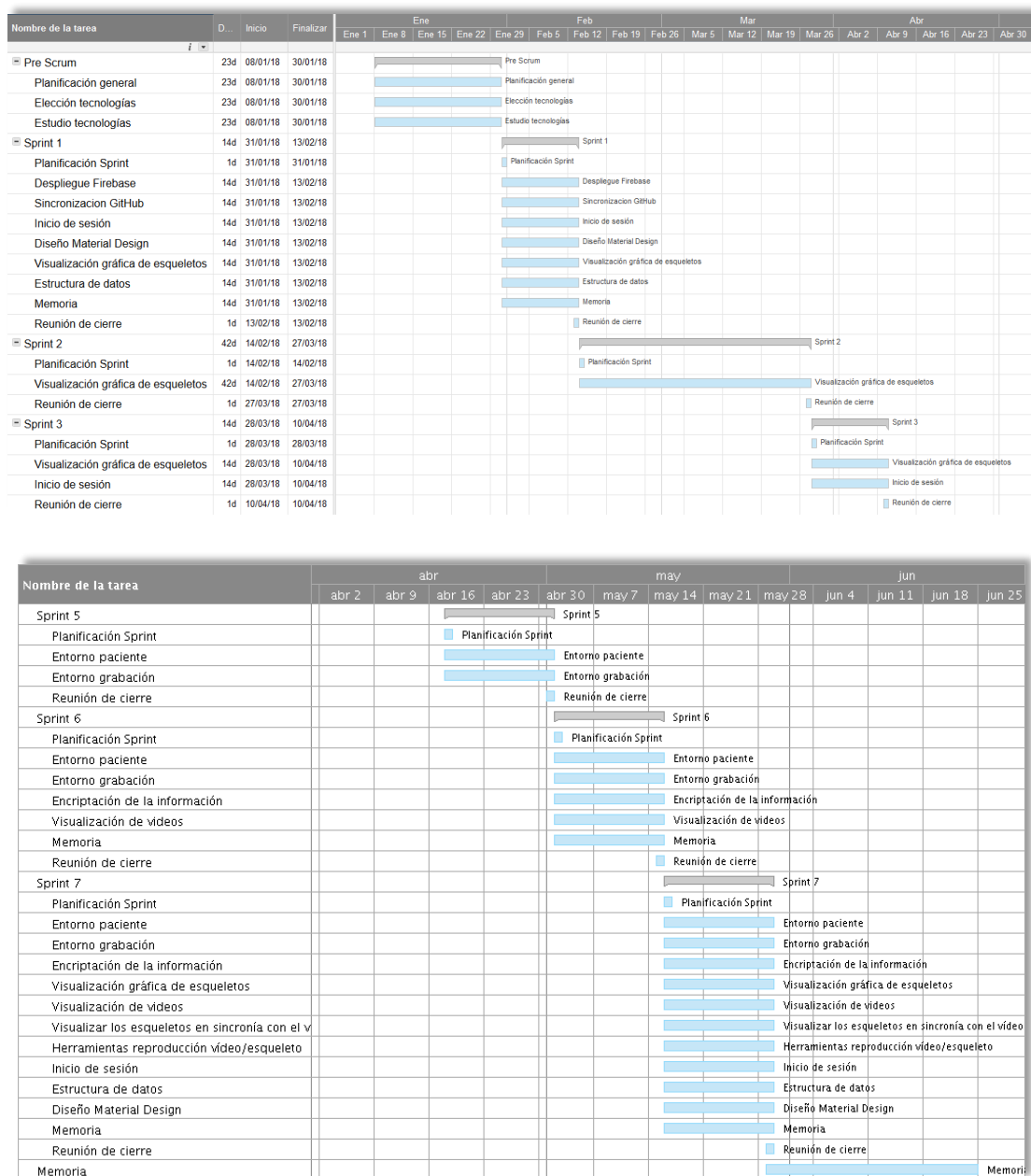


Ilustración 53. Versión final diagrama de Gantt

9.2. Dedicaciones

A continuación, se muestran las dedicaciones globales del proyecto y las dedicaciones de cada una de las iteraciones. Las horas de preparación de la defensa no están incluidas.

	Tiempo (Minutos)	Tiempo (Horas)
Planificación	210	3,5
Desarrollo	9360	156
Estudio	2310	38,5
Memoria	6570	109,5
Reuniones	750	12,5
Total	19200	320
Total Previsto	18000	300

Tabla 7. Dedicación total

	Pre SCRUM	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Memoria
Planificación	60	120					30		
Desarrollo		990	600	540	330	1020	1710	3630	540
Estudio	60	840	330	270	90	60	240	420	0
Memoria		30					300	510	5730
Reuniones	60	30	180	90	120	30	60	60	120
Total minutos	180	2010	1110	900	540	1110	2340	4620	6390
Horas	3	33,5	18,5	15	9	18,5	39	77	106.5

Tabla 8. Dedicaciones por iteración

Las dedicaciones a cada una de las tareas a lo largo de las iteraciones están reflejadas en las actas de las reuniones de cierre de iteración disponibles en el [Anexo D. Actas reuniones](#).

10. Conclusiones

En este capítulo se presentan las conclusiones del proyecto, para ello se analizan varios aspectos del proyecto y su desarrollo. Primero se hace una valoración de la metodología empleada y de las tecnologías utilizadas. Posteriormente, se valoran los conocimientos adquiridos y la experiencia personal. Finalmente se proponen puntos de extensión y mejora para la aplicación web. La aplicación desarrollada está disponible a través del siguiente enlace: <https://gestion-pacientes-parkinson.firebaseio.com/>.

10.1. Metodología y tecnologías empleadas

La metodología utilizada para desarrollar el proyecto ha sido Scrum. Haber aplicado una metodología ágil ha sido beneficioso para el desarrollo. Entre otras cosas, el hecho de que al final de cada iteración los tres miembros del equipo, entre ellos el cliente, se reuniese para valorar lo realizado en esa iteración y definir los objetivos de la siguiente, ha ayudado a cumplir con los objetivos iniciales del proyecto. De esta forma se ha logrado que la aplicación haya ido evolucionando a lo largo del desarrollo hasta plasmar las necesidades del cliente.

También ha sido de gran ayuda que en cada una de las iteraciones se focalizase en un número reducido de tareas. De este modo, ha sido más sencillo cumplir los objetivos de cada iteración.

En cuanto a las tecnologías empleadas, ha sido un gran acierto la elección de AngularJS como framework para desarrollar el lado cliente de la aplicación. Aunque la curva de aprendizaje es elevada, una vez interiorizadas la filosofía y las peculiaridades de la tecnología, el desarrollo se simplifica enormemente. Cabe destacar la cantidad de recursos disponibles a través de Internet para la adquisición de los conocimientos necesarios para trabajar con AngularJS. Además, la existencia de la librería AngularJS Material ha facilitado enormemente el apartado visual de la aplicación.

La tecnología escogida para la parte servidor también ha sido un acierto. Firebase ha servido como hosting de la aplicación, como gestor de usuarios, base de datos y almacenamiento de ficheros. Es un todo en uno increíblemente útil. Aunque inicialmente resultaba extraño no tener control sobre el código de la parte servidor, enseguida se vio el gran potencial de Firebase. Otra de las facilidades ofrecidas por Firebase es la sencillez con la que desde el cliente se puede llevar a cabo la comunicación con Firebase.

Otras tecnologías como CryptoJS, para encriptar la información, o Papaparse, para trabajar cómodamente con los ficheros CSV, también han sido de gran ayuda en el desarrollo.

Sin embargo, tal y como se cuenta en el capítulo [7.7. Dificultades durante el desarrollo](#), la librería three.js no fue un acierto. La falta de documentación hizo que fuese imposible desarrollar la visualización de esqueletos, teniendo que recurrir al uso de herramientas propias de HTML. La elección de esta tecnología tiene como resultado que una gran cantidad de horas de trabajo no se vean reflejadas en la aplicación final.

10.2. Conocimientos adquiridos y experiencia personal

Gracias al proyecto se han adquirido conocimientos de tecnologías sobre las que no se trabaja en el grado. La profundización en las tecnologías utilizadas capacita poder comenzar proyectos de una envergadura igual a mayor a la de este proyecto. Se ha adquirido un mayor bagaje sobre el desarrollo en JavaScript y HTML, más concretamente en AngularJS. Además de conocimientos en torno a seguridad y cifrado de datos.

También se han reforzado los conocimientos adquiridos en torno a la planificación y gestión de proyectos. Esta vez con un proyecto real y de larga duración.

En resumen, los conocimientos adquiridos en el Trabajo de Fin de Grado resultan enriquecedores de cara a la formación personal y futura inserción en el mundo laboral.

10.3. Puntos de extensión y mejora

Aunque los requisitos y objetivos definidos al principio del proyecto se han alcanzado satisfactoriamente, el tiempo a dedicar al Trabajo de Fin de Grado está acotado. Es por ello que a continuación se proponen mejoras que se podrían incluir en la aplicación:

- **Gestión de claves en la nube.** Tal y como se comenta en el capítulo [6.3.2.3. Gestión de las claves de cifrado](#), la gestión de claves ideal se realizaría usando un servicio de gestión de claves en la nube como el que ofrecen Google o Amazon.
- **Recuperación clave de cifrado.** Actualmente si un grupo de usuarios olvida la clave de cifrado asociada a su grupo esto supone la pérdida de toda la información sobre pacientes y grabaciones relativa a ese grupo.
- **Visualización mejorada de los esqueletos.** Sería un gran avance y de gran ayuda para el grupo de investigación implementar la visualización de los esqueletos en 3 dimensiones, tal y como se intentó hacer al comienzo del proyecto. Como parte de la visualización tridimensional se podría permitir al usuario la rotación visual en torno al esqueleto del paciente y la posibilidad de acercar o alejar la vista.
- **Reproducción mejorada.** Otra mejora en torno a la visualización sería permitir escoger la velocidad a la que se quiere visualizar el esqueleto junto al vídeo.
- **Anotaciones mejoradas.** En un futuro se podría permitir asociar cada nota que se tome con una articulación para que posteriormente se muestre exactamente en el lugar en el que se encuentre la articulación.
- **Idiomas:** Traducir la aplicación a otros idiomas, como el euskera o el inglés.
- **Crear pacientes:** Permitir crear un nuevo paciente durante el registro de una grabación.
- **Borrar grabaciones:** Preguntar al usuario si al borrar un paciente desea eliminar las grabaciones asignadas a ese paciente. Actualmente cuando se elimina un paciente las grabaciones asociadas pasan a estar sin asignar a ningún paciente.

11. Bibliografía

En este capítulo se muestran las referencias bibliográficas más destacadas en relación al Trabajo de Fin de Grado. No se muestran consultas realizadas para la formación o con el objetivo de solucionar problemas durante el desarrollo.

- [1] BDI, «BDI,» [En línea]. Available: <http://bdi.si.ehu.es/bdi/>.
- [2] COM SALUD, «eSalud,» [En línea]. Available: <http://laesalud.com/esalud/>.
- [3] Microsoft, «Microsoft Kinect,» [En línea]. Available: <https://msdn.microsoft.com/en-us/library/hh438998.aspx>.
- [4] R, «R,» [En línea]. Available: [https://en.wikipedia.org/wiki/R_\(programming_language\)](https://en.wikipedia.org/wiki/R_(programming_language)).
- [5] Scrum, «Scrum,» [En línea]. Available: <https://www.scrum.org/>.
- [6] «Pinterest,» [En línea]. Available: <https://www.pinterest.es/pin/2264446687496242066/>.
- [7] Google, «AngularJS,» [En línea]. Available: <https://angularjs.org/>.
- [8] Facebook, «ReactJS,» [En línea]. Available: <https://reactjs.org/>.
- [9] AngularJS Material, «AngularJS Material,» [En línea]. Available: <https://material.angularjs.org>.
- [10] «Materializa,» [En línea]. Available: <https://materializecss.com/>.
- [11] Google, «Google Plus,» [En línea]. Available: <https://plus.google.com/+AngularJS/posts/aZNVhj355G2> .
- [12] Papaparse, «Papaparse,» [En línea]. Available: <https://www.papaparse.com/>.
- [13] CryptoJS, «CryptoJS,» [En línea]. Available: <https://code.google.com/archive/p/crypto-js/>.
- [14] MomentJS, «MomentJS,» [En línea]. Available: <https://momentjs.com/>.
- [15] daniel-nagy, «GitHub,» [En línea]. Available: <https://github.com/daniel-nagy/md-data-table>.
- [16] Google, «Firebase default encryption,» [En línea]. Available: <https://cloud.google.com/security/encryption-at-rest/default-encryption/>.
- [17] Google, «Capas encriptación Firebase,» [En línea]. Available: <https://cloud.google.com/images/security/whitepaper-chart-01.png>.
- [18] Google, «Gestión de datos en Google,» [En línea]. Available: <https://cloud.google.com/images/security/whitepaper-chart-02.png>.
- [19] Google, «Google Gestión Claves,» [En línea]. Available: <https://cloud.google.com/images/security/whitepaper-chart-03.png>.
- [20] «Entropysecurity,» [En línea]. Available: <http://entropysecurity.com/benchmarking-algoritmos-simetricos/>.

- [21] @encryb, «medium,» [En línea]. Available: <https://medium.com/@encryb/comparing-performance-of-javascript-cryptography-libraries-42fb138116f3>.
- [22] Google, «Firebase,» [En línea]. Available: <https://firebase.google.com/docs/database/admin/retrieve-data?hl=es-419>.
- [23] Google, «Firebase,» [En línea]. Available: https://firebase.google.com/docs/database/web/read-and-write?hl=es-419#basic_write.
- [24] Mozilla, «Mozilla Developer,» [En línea]. Available: https://developer.mozilla.org/es/docs/Web/API/Fetch_API/Utilizando_Fetch.
- [25] joecliff, «GitHub,» [En línea]. Available: <https://gist.github.com/joecliff/10948592>.
- [26] Mozilla, «Mozilla Developer,» Mozilla, [En línea]. Available: https://developer.mozilla.org/es/docs/Web/Guide/HTML/Canvas_tutorial.
- [27] W3C, «w3schools,» [En línea]. Available: https://www.w3schools.com/jsref/met_win_setinterval.asp.
- [28] «Three.js,» [En línea]. Available: <https://threejs.org/>.
- [29] «ThreeJS ejemplo esqueleto,» [En línea]. Available: https://threejs.org/examples/#webgl_animation_skinning_blending.
- [30] «Geogebra,» [En línea]. Available: <https://www.geogebra.org/>.
- [31] M. Angelov, «Tutorialzine,» [En línea]. Available: <https://tutorialzine.com/2013/11/javascript-file-encrypter> .
- [32] «Stack Overflow,» [En línea]. Available: <https://stackoverflow.com/>.
- [33] Wikipedia, «Wikipedia,» [En línea]. Available: [https://es.wikipedia.org/wiki/Caja_negra_\(sistemas\)](https://es.wikipedia.org/wiki/Caja_negra_(sistemas)).
- [34] [En línea]. Available: <https://cloud.google.com/images/security/whitepaper-chart-01.png>.

Anexo A.

Historias de Usuario

En este capítulo se presenta la versión final de las historias de usuario del proyecto. Al principio todas las historias no contaban con el nivel de detalle actual.

Historia de usuario

Número: 0

Usuario: Cliente

Nombre historia: Memoria del proyecto

Descripción:

Como cliente quiero una memoria detallada del proyecto para poder realizar un seguimiento del mismo. Además de poder conocer todas las características de la aplicación resultante del proyecto.

Historia de usuario

Número: 1

Usuario: Cliente

Nombre historia: Usuarios

Descripción:

Como cliente quiero que solamente puedan acceder a la aplicación personas autorizadas, la información solo será accesible una vez se haya iniciado sesión. Los usuarios se crearán de forma manual por el administrador del sistema y habrá grupos de usuarios. Cada grupo representará una institución, como por ejemplo el Hospital Donostia.

Notas:

- Solo el administrador podrá crear usuarios.
- Se creará un formulario de inicio de sesión.
- Se crearán grupos de usuarios en la base de datos.
- Se permitirá la recuperación de contraseñas.

Test:

- Comprobar que solo pueden iniciar sesión los usuarios autorizados.
- No se puede acceder a información de pacientes o grabaciones sin iniciar sesión.
- Un usuario puede cambiar de contraseña.

Historia de usuario

Número: 3

Usuario: Cliente

Nombre historia: Entorno grabaciones

Descripción:

Como cliente quiero poder ver información sobre las grabaciones, así como poder añadir nuevas grabaciones y gestionar las existentes. Además, quiero poder previsualizar el fichero CSV de la grabación y descargarlo tanto en formato CSV como en JSON (en este formato estarán incluidas las notas asociadas a la grabación).

Notas:

- Puedo añadir nuevas grabaciones indicando un identificador, la fecha de grabación, el paciente al que corresponde, el fichero csv que contiene los datos de la grabación y el vídeo de la misma. Además, puedo añadir tantos atributos extra como quiera.
- Solo puedo ver las grabaciones que formen parte de mi grupo.
- Las grabaciones se muestran ordenadas en una tabla.
- Puedo editar cualquier información sobre la grabación pulsando sobre ese dato en la tabla.
- Puedo ordenar las grabaciones por cualquier campo.
- Puedo seleccionar una o más grabaciones para eliminarlas.
- Puedo filtrar las grabaciones por su identificador.
- Puedo seleccionar el número de pacientes que se muestra por página.

Test:

- Añadir una nueva grabación y comprobar que se ha añadido correctamente a la base de datos.
- Las grabaciones pueden incluir tantos atributos extra como se crea conveniente.
- Asignar un paciente a una grabación al paciente a la hora de crear una grabación.
- Comprobar que las nuevas grabaciones solamente son visibles para los usuarios que pertenecen al mismo grupo que el usuario que la ha añadido.
- Eliminar una grabación y comprobar que se ha eliminado en la base de datos y no se muestra más en la tabla.
- Eliminar varias grabaciones y comprobar que se han eliminado en la base de datos y no se muestran más en la tabla.
- Editar todos los campos y comprobar que los cambios se ven reflejados en la base de datos.
- Filtrar las grabaciones por el identificador.
- Reordenar la tabla por columnas.
- Ocultar columnas
- Previsualizar el fichero csv.
- Descargar el fichero csv.
- Descargar el fichero json y comprobar que incluye las notas de la grabación.

Historia de usuario

Número: 5

Usuario: Cliente

Nombre historia: Visualización de vídeos

Descripción:

Como cliente quiero visualizar los vídeos de las grabaciones desde la aplicación web.

Notas:

- Puedo ver el vídeo correspondiente a una grabación.
- Puedo pausar/reproducir el vídeo.
- Puedo retroceder/avanzar el vídeo.

Test:

- Comprobar que el vídeo se ve correctamente.
- Pausar y retomar el vídeo.
- Avanzar y retroceder en la reproducción del vídeo.

Historia de usuario

Número: 6

Usuario: Cliente

Nombre historia: Visualización de esqueletos en sincronía con el vídeo

Descripción:

Como cliente quiero visualizar el esqueleto y el vídeo correspondiente a una grabación de forma sincronizada con él para poder analizar los movimientos del paciente. También quiero poder elegir si veo el vídeo, el esqueleto o ambas cosas.

Notas:

- Puedo ver el vídeo y el esqueleto correspondiente a una grabación sincronizados.
- Puedo elegir si quiero ver el vídeo, la grabación o ambos.

Test:

- Comprobar que el vídeo se ve correctamente sincronizado con el esqueleto.
- Visualizar solamente el vídeo.
- Visualizar solamente la vista frontal del esqueleto.
- Visualizar solamente la vista lateral del esqueleto.
- Visualizar tanto el vídeo como el esqueleto.

Historia de usuario

Número: 7

Usuario: Cliente

Nombre historia: Herramientas reproducción vídeo/esqueleto

Descripción:

Como cliente quiero poder controlar la reproducción del vídeo/esqueleto para poder pausar, reproducir, retroceder y avanzar.

Notas:

- Puedo pausar/reproducir la visualización.
- Puedo retroceder/avanzar la visualización.

Test:

- Pausar y retomar la reproducción, comprobando que continúan sincronizados.
- Avanzar en la reproducción, comprobando que continúan sincronizados.
- Retroceder en la reproducción, comprobando que continúan sincronizados.
- Quitar el sonido al vídeo.
- Regular el volumen del vídeo.

Historia de usuario

Número: 8

Usuario: Cliente

Nombre historia: Encriptación de la información

Descripción:

Como cliente quiero que tanto los ficheros de datos y vídeos como la información se almacene cifrada para cumplir con las indicaciones de la LOPD. Además, quiero evitar que el administrador de Firebase pueda visualizar los datos de los pacientes y de las grabaciones.

Notas:

- Los ficheros csv son encriptados antes de ser almacenados en la nube.
- Los ficheros de vídeo son encriptados antes de ser almacenados en la nube.
- Toda la información de pacientes y grabaciones es encriptada antes de almacenarse.
- Se pedirá a los usuarios una clave de cifrado a la hora de iniciar sesión.

Test:

- Comprobar que los ficheros csv se almacenan encriptados y que no se puede interpretar su contenido sin ser descifrado previamente.
- Comprobar que los ficheros de vídeo se almacenan encriptados y que no se puede interpretar su contenido sin ser descifrado previamente.
- Comprobar que la información de pacientes y grabaciones se almacena encriptada y que no se puede interpretar su contenido sin ser descifrada previamente.
- Comprobar que no deja acceder a la aplicación sin introducir la clave de cifrado correcta.

Historia de usuario

Número: 9

Usuario: Cliente

Nombre historia: Despliegue en Firebase

Descripción:

Como cliente quiero poder acceder a la aplicación web a través de Firebase para poder usar la aplicación y ver su evolución durante el desarrollo.

Notas:

- La aplicación web estará accesible a través de Internet desde una URL.

Test:

- Comprobar que el cliente puede acceder a la aplicación web desde la URL facilitada por Firebase.

Historia de usuario

Número: 10

Usuario: Desarrollador

Nombre historia: Sincronización GitHub

Descripción:

Como desarrollador quiero que se sincronice el proyecto en un repositorio de GitHub mediante Git para poder realizar un control de versiones.

Notas:

- Se realizará control de versiones del proyecto en GitHub.
- Se actualizará el repositorio de GitHub como mínimo al final de cada iteración.

Test:

- Comprobar que el repositorio es accesible desde GitHub y que el control de versiones se realiza correctamente.
- Se ha actualizado el proyecto mínimo una vez por iteración.

Historia de usuario

Número: 11

Usuario: Cliente

Nombre historia: Estructura fichero datos

Descripción:

Como cliente quiero que se diseñe una nueva estructura del fichero de datos para que, en un futuro, si es necesario añadir más información en el fichero, la aplicación pueda seguir funcionando sin realizar demasiados cambios.

Notas:

- Se diseñará una nueva estructura para los datos.

Test:

- Comprobar que la estructura propuesta cumple con los requisitos y que es adecuada para el propósito definido.

Anexo B.

Flujo de eventos

En este anexo se muestran el resto de flujos de eventos del capítulo [5.2 Flujo de eventos](#).

1. Iniciar sesión

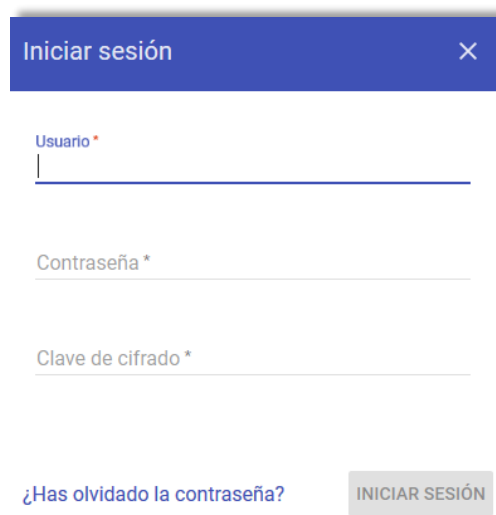
Descripción: Permite al usuario iniciar sesión en la aplicación.

Actores: Usuario.

Precondiciones: Disponer de un usuario y contraseña para acceder a la aplicación.

Flujo de eventos:

1. El usuario pulsa en el botón “Iniciar sesión” que se encuentra en la esquina superior derecha de la página principal.
2. El usuario introduce su nombre de usuario (correo electrónico).
 - 2.1. Si el nombre de usuario no tiene un formato de correo válido el sistema muestra un mensaje de error.
3. El usuario introduce su contraseña.
 - 3.1. Si la contraseña tiene una longitud inferior a 6 caracteres el sistema muestra un mensaje de error.
4. El usuario introduce la clave de cifrado de su grupo.
 - 4.1. Si la clave de cifrado tiene una longitud inferior a 6 caracteres el sistema muestra un mensaje de error.
5. El usuario pulsa el botón “Iniciar sesión”.
6. El sistema verifica que los datos introducidos son correctos e inicia la sesión del usuario.
 - 6.1. Si ha habido algún problema verificando los datos, el sistema muestra un mensaje de error.



El formulario de inicio de sesión contiene los siguientes elementos:

- Encabezado: "Iniciar sesión" con un botón de cerrar "X".
- Campo de texto: "Usuario *".
- Campo de texto: "Contraseña *".
- Campo de texto: "Clave de cifrado *".
- Enlace: "¿Has olvidado la contraseña?".
- Botón: "INICIAR SESIÓN".

Ilustración 54. Inicio de sesión

2. Recuperar contraseña

Descripción: Permite al usuario recuperar la contraseña.

Actores: Usuario.

Precondiciones: Disponer de un usuario y contraseña para acceder a la aplicación.

Flujo de eventos:

1. El usuario pulsa sobre “¿Has olvidado la contraseña?” en el formulario de inicio de sesión.
2. El usuario introduce su nombre de usuario.
 - 2.1 Si el nombre de usuario no tiene un formato de correo válido el sistema muestra un mensaje de error.
3. El usuario pulsa sobre el botón “Restaurar”.
4. El sistema envía un correo electrónico con un enlace para crear una contraseña nueva.
 - 4.1 Si el nombre de usuario no está registrado, el sistema informa al usuario de que no se ha podido enviar el correo de recuperación. Continúa en el paso 1.
5. El usuario accede al correo electrónico y pulsa en el enlace.
6. El usuario introduce la nueva contraseña.
7. El sistema actualiza la contraseña del usuario.
 - 7.1 Si ha habido algún error el sistema informa al usuario. Continúa en el paso 3.

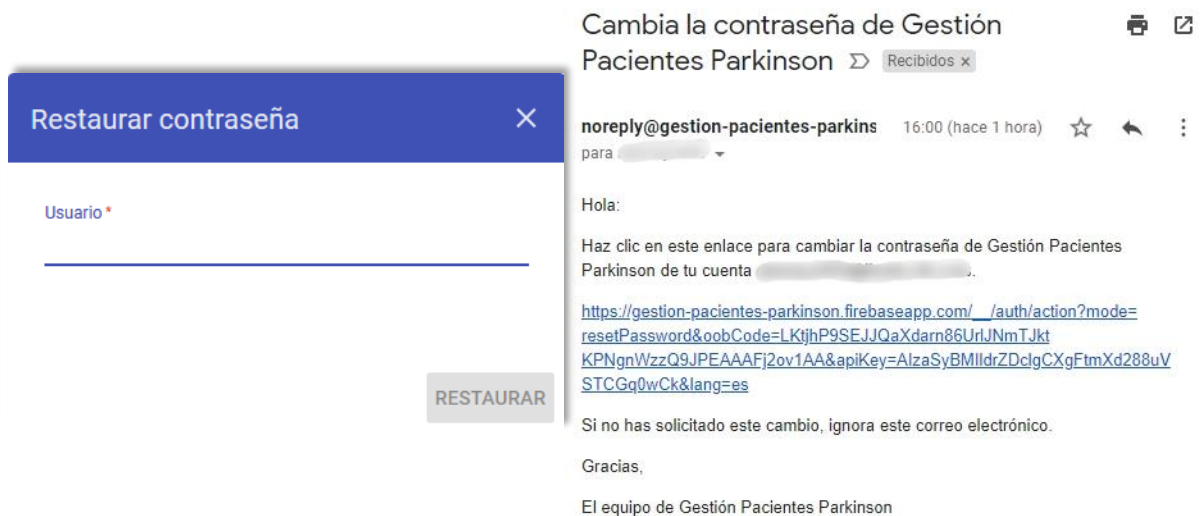


Ilustración 55. Recuperar contraseña

3. Crear paciente

Descripción: Permite al usuario registrar un nuevo paciente en el sistema.

Actores: Usuario con sesión iniciada.

Precondiciones: Haber iniciado sesión en el sistema.

Flujo de eventos:

1. El usuario accede a la sección “*Pacientes*” y pulsa sobre el botón con símbolo + que se encuentra en la esquina superior derecha de la tabla.
2. El usuario introduce información sobre el paciente; identificador, fecha de nacimiento, año de inicio de la enfermedad, clasificación y sexo.
 - 2.1. Si el usuario no introduce todos los campos, el sistema muestra un mensaje indicando qué campos son obligatorios y el botón “*Crear Paciente*” bloqueado.
 - 2.2. Si alguno de los datos introducidos no es correcto, el sistema informa al usuario mediante un mensaje de error. Continúa en el paso 2.
3. Si lo desea, el usuario incluye el peso y/o la altura.
 - 3.1. Si alguno de los datos introducidos no es correcto, el sistema informa al usuario mediante un mensaje de error. Continúa en el paso 3.
4. Si hay grabaciones que no tienen un paciente asignado:
 - 4.1. El sistema muestra las grabaciones disponibles.
 - 4.2. Si lo desea, el usuario asigna al paciente una de estas grabaciones.
5. Si lo desea, el usuario pulsa sobre el botón con el símbolo + para añadir un atributo extra.
 - 5.1. El usuario indica el identificador y el valor del nuevo atributo.
 - 5.1.1. Si no se introduce el identificador y el valor, el sistema muestra un mensaje indicando que ambos campos son obligatorios.
 - 5.2. Si lo desea, el usuario pulsa sobre el botón con el símbolo – para eliminar el último atributo extra añadido. Continúa en el paso 5.
6. El usuario pulsa sobre el botón “*Crear paciente*”.
7. El sistema verifica que los datos introducidos son correctos, los encripta y almacena el paciente en Firebase. Muestra un mensaje indicando que el paciente se ha creado con éxito.
 - 7.1. Si hay algún error durante la encriptación de los datos o al almacenar los datos en la base de datos, el sistema muestra un mensaje de error.

Crear paciente

Identificador * F. nacimiento * 22/06/2018

Peso (kg) Altura (cm)

Año inicio enfermedad * 2018 Clasificación *

Sexo * Grabación

Atributos extra +

CREAR PACIENTE

Paciente creado correctamente

Ilustración 56. Crear paciente

4. Editar paciente

Descripción: Permite al usuario editar cualquier dato acerca del usuario.

Actores: Usuario con sesión iniciada.

Precondiciones: Haber iniciado sesión en el sistema y que haya al menos un paciente asociado al grupo de usuarios al que pertenece el actor.

Flujo principal:

1. El usuario accede a la sección “*Pacientes*”.
2. El usuario pulsa sobre el atributo del paciente que quiere editar.
3. El sistema muestra en una ventana emergente el valor actual del atributo.
4. El usuario escribe el nuevo valor del atributo en la ventana mostrada.
 - 4.1. Si el dato introducido no es correcto el sistema muestra un mensaje de error.
5. El usuario pulsa sobre el botón “*Guardar*”.
6. El sistema verifica que el nuevo valor del atributo es correcto, lo encripta y actualiza el paciente en la base de datos.
 - 6.1. Si hay algún error el sistema muestra un mensaje indicando qué ha ocurrido.

<input type="checkbox"/>	Identificador ↑	Fecha Nacimiento
<input type="checkbox"/>	Identificador	24/05/1959
<input type="checkbox"/>	Snape	21/09/1979
<input type="checkbox"/>	CANCELAR GUARDAR	09/06/2018

Ilustración 57. Editar paciente

5. Ver pacientes

Descripción: Permite al usuario ver la lista de pacientes, además de poder elegir la cantidad de pacientes que quiere ver, filtrarlos por su identificador u ordenarlos por algún atributo.

Actores: Usuario con sesión iniciada.

Precondiciones: Haber iniciado sesión en el sistema y que haya al menos un paciente asociado al grupo de usuarios al que pertenece el actor.

Flujo principal:

1. El usuario accede a la sección “*Pacientes*”.
2. El sistema obtiene de la base de datos, descripta y muestra en una tabla todos los pacientes pertenecientes al grupo del usuario.
 - 2.1. Si el paciente tiene al menos una grabación asignada:
 - 2.1.1. El sistema muestra en la columna *Grabaciones* el botón “*Ver*”.
 - 2.1.2. Si el usuario pulsa el botón “*Ver*”:
 - 2.1.2.1. El sistema redirige al usuario a la sección “*Grabaciones*” mostrando únicamente las grabaciones asociadas al paciente.
 - 2.2. Si el paciente tiene al menos un atributo extra:
 - 2.2.1. El sistema muestra el botón “*Ver*” en la columna *Atributos extra*.
 - 2.2.2. Si el usuario pulsa el botón “*Ver*”:
 - 2.2.2.1. El sistema le muestra los atributos extra asignados al paciente.
 - 2.3. Si el usuario pulsa sobre la cabecera de una columna de la tabla.
 - 2.3.1. El sistema reordena la tabla en base al atributo pulsado.
 - 2.4. Si el usuario pulsa sobre el icono ☰ puede filtrar los pacientes:
 - 2.4.1. El sistema muestra una barra de búsqueda en la parte superior de la tabla.
 - 2.4.2. El usuario escribe en la barra de búsqueda.
 - 2.4.3. El sistema muestra los pacientes que contienen en su identificador el texto introducido.
 - 2.5. Si el usuario desmarca el nombre de una columna en la lista de atributos que aparece en la parte superior de la pantalla:
 - 2.5.1. El sistema oculta esa columna de la tabla
 - 2.6. Si el usuario marca el nombre de una columna en la lista de atributos que aparece en la parte superior de la pantalla:
 - 2.6.1. El sistema muestra esa columna en la tabla.
 - 2.7. Si el usuario pulsa en la lista desplegable “*Página*”:
 - 2.7.1. Si hay más pacientes que el número indicado en “*Filas por página*”:
 - 2.7.1.1. El sistema muestra las páginas disponibles.
 - 2.7.1.2. El usuario selecciona el número de la página en la lista desplegable.
 - 2.7.1.3. El sistema muestra los pacientes de esa página.
 - 2.8. Si el usuario pulsa en la lista desplegable “*Filas por página*”:
 - 2.8.1. El sistema muestra los valores de filas por página disponibles.
 - 2.8.2. Si el usuario selecciona un valor de la lista:
 - 2.8.2.1. Si el número de pacientes del grupo del usuario es menor o igual al valor seleccionado:
 - 2.8.2.1.1. El sistema muestra todos los pacientes en la primera página.

- 2.8.2.2. Si el número de pacientes es superior al valor seleccionado:
 - 2.8.2.2.1. El sistema muestra el valor seleccionado de pacientes en la primera página y el resto los divide en tantas páginas como sea necesario.
- 2.9. Si el usuario pulsa en la flecha de navegación izquierda de la parte inferior derecha de la tabla:
 - 2.9.1. Si hay más de una página de pacientes:
 - 2.9.1.1. El sistema mostrará la primera página de pacientes.
- 2.10. Si el usuario pulsa en la flecha de navegación central izquierda de la parte inferior derecha de la tabla:
 - 2.10.1. Si hay más de una página de pacientes:
 - 2.10.1.1. El sistema mostrará la página de pacientes anterior a la actual.
- 2.11. Si el usuario pulsa en la flecha de navegación central derecha de la parte inferior derecha de la tabla:
 - 2.11.1. Si hay más de una página de pacientes:
 - 2.11.1.1. El sistema mostrará la página de pacientes siguiente a la actual.
- 2.12. Si el usuario pulsa en la flecha de navegación derecha de la parte inferior derecha de la tabla:
 - 2.12.1. Si hay más de una página de pacientes:
 - 2.12.1.1. El sistema mostrará la última página de pacientes.

Configuración

De-Seleccionar todas
 Fecha nacimiento
 Edad
 Sexo
 Altura
 Peso
 Inicio enfermedad
 Clasificación
 Grabaciones
 Atributos extra

Pacientes

<input type="checkbox"/> Identificador ↑	Fecha Nacimiento	Edad	Sexo	Altura(cm)	Peso	Inicio enfermedad	Clasificación	Grabaciones	Atributos extra
<input type="checkbox"/> Han Solo	20/10/1951	66	Hombre -	190	Añadir peso	2000	2 -	VER	VER
<input type="checkbox"/> Leia	21/09/1979	38	Mujer -	Añadir altura	58	2012	1 -		AÑADIR
<input type="checkbox"/> Snape	21/04/1962	56	Mujer -	158	85	2015	2 -		AÑADIR

Página: 1 Filas por página: 10 1 - 3 de 3 < > >>

Ilustración 58. Ver pacientes

6. Eliminar paciente/s

Descripción: Permite al usuario eliminar uno o más pacientes.

Actores: Usuario con sesión iniciada.

Precondiciones: Haber iniciado sesión en el sistema y que haya al menos un paciente asociado al grupo de usuarios al que pertenece el actor.

Flujo principal:

1. El usuario accede a la sección “*Pacientes*”.
2. El usuario selecciona los pacientes que quiere borrar.
3. El sistema solicita la confirmación del usuario:
 - 3.1. Si el usuario confirma que quiere borrar los pacientes pulsando el botón “*Continuar*”:
 - 3.1.1. El sistema elimina los pacientes de la base de datos. Muestra un mensaje indicando que los pacientes se han borrado correctamente.
 - 3.1.2. Si ha ocurrido algún error durante el borrado, el sistema informa al usuario mediante un mensaje de error y vuelve a la vista general de la sección “*Pacientes*”.
 - 3.2. Si el usuario cancela el borrado de pacientes:
 - 3.2.1. El sistema vuelve a la vista general de la sección “*Pacientes*”.



Paciente borrado correctamente

Ilustración 59. Eliminar paciente/s

7. Crear grabación

Descripción: Permite al usuario registrar una nueva grabación en el sistema.

Actores: Usuario con sesión iniciada.

Precondiciones: Haber iniciado sesión en el sistema.

Flujo de eventos:

1. El usuario accede a la sección “Grabaciones” y pulsa sobre el botón con símbolo + que se encuentra en la esquina superior derecha de la tabla.
2. El usuario introduce información sobre la grabación; identificador, lugar, fecha de grabación, y fichero CSV de la grabación.
 - 2.1. Si el usuario no introduce todos los campos, el sistema muestra un mensaje indicando qué campos son obligatorios y el botón Crear grabación aparece bloqueado.
 - 2.2. Si alguno de los datos introducidos no es correcto, el sistema informa al usuario mediante un mensaje de error. Continúa en el paso 2.
3. Si lo desea, el usuario incluye el vídeo de la grabación.
 - 3.1. Si el fichero introducido no es un vídeo, el sistema informa al usuario mediante un mensaje de error. Continúa en el paso 3.
4. Si hay pacientes registrados en el grupo del usuario:
 - 4.1. Si lo desea, el usuario asigna la grabación a un paciente.
5. Si lo desea, el usuario pulsa sobre el botón con el símbolo + para añadir un atributo extra.
 - 5.1. El usuario indica el identificador y el valor del nuevo atributo.
 - 5.1.1. Si no se introduce el identificador y el valor, el sistema indica que ambos campos son obligatorios y el botón Crear grabación aparece bloqueado.
 - 5.2. Si lo desea, el usuario pulsa sobre el botón con el símbolo – para eliminar el último atributo extra añadido. Continúa en el paso 5.
6. El usuario pulsa sobre el botón “Crear grabación”.
7. El sistema verifica que los datos introducidos son correctos, los encripta y almacena la grabación en la base de datos de Firebase. Muestra un mensaje indicando que la grabación se ha creado con éxito.
 - 7.1. Si hay algún error durante la encriptación de los datos o al almacenar los datos en la base de datos, el sistema muestra un mensaje de error.

The screenshot shows a web form titled "Crear grabación" with a close button (X) in the top right corner. The form contains several input fields and buttons:

- A red circular button with a white plus sign (+) in the top left.
- Input fields for "Identificador *" and "Lugar *".
- A date picker for "F. grabación *" showing "22/06/2018".
- A dropdown menu for "Paciente".
- Two buttons labeled "ELEGIR FICHERO" with text "CSV no seleccionado" and "Vídeo no seleccionado" respectively.
- An "Atributos extra" section with a red circular button with a white plus sign (+) and a text input field.
- A "CREAR GRABACIÓN" button at the bottom.

To the right of the form, a black rectangular box contains the white text "Grabacion creada correctamente".

Ilustración 60. Crear grabación

8. Editar grabación

Descripción: Permite al usuario editar cualquier dato acerca de la grabación.

Actores: Usuario con sesión iniciada.

Precondiciones: Haber iniciado sesión en el sistema y que haya al menos una grabación asociada al grupo de usuarios al que pertenece el actor.

Flujo principal:

1. El usuario accede a la sección “Grabaciones”.
2. El usuario pulsa sobre el atributo de la grabación que quiere editar.
3. El sistema muestra en una ventana emergente el valor actual del atributo.
4. El usuario escribe el nuevo valor del atributo en la ventana mostrada.
 - 4.1. Si el dato introducido no es correcto el sistema muestra un mensaje de error.
5. El usuario pulsa sobre el botón “Guardar”.
6. El sistema verifica que el nuevo valor del atributo es correcto, lo encripta y actualiza la grabación en la base de datos.
 - 6.1. Si hay algún error el sistema muestra un mensaje indicando qué ha ocurrido.

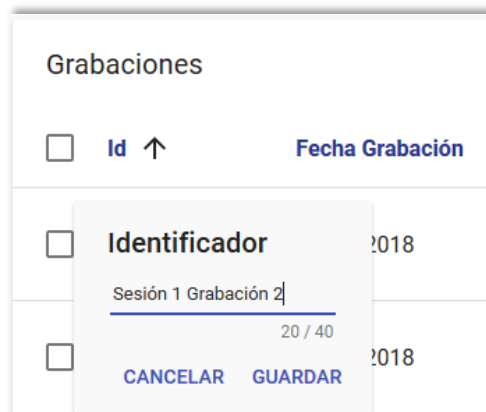


Ilustración 61. Editar grabación

9. Ver grabaciones

Descripción: Permite al usuario ver la lista de grabaciones, además de poder elegir la cantidad de grabaciones que quiere ver, filtrarlas por su identificador u ordenarlas por algún atributo.

Actores: Usuario con sesión iniciada.

Precondiciones: Haber iniciado sesión en el sistema y que haya al menos una grabación asociada al grupo de usuarios al que pertenece el actor.

Flujo principal:

1. El usuario accede a la sección “Grabaciones”.
2. El sistema obtiene de la base de datos, descripta y muestra en una tabla todas las grabaciones pertenecientes al grupo del usuario.
 - 2.1. Si la grabación tiene al menos un atributo extra:
 - 2.1.1. El sistema muestra el botón “Ver” en la columna *Atributos extra*.
 - 2.1.2. Si el usuario pulsa el botón “Ver”:
 - 2.1.2.1. El sistema le muestra los atributos extra asignados a la grabación.
 - 2.2. Si la grabación tiene al menos una nota:
 - 2.2.1. El sistema muestra el botón “Ver” en la columna *Notas*.
 - 2.2.2. Si el usuario pulsa el botón “Ver”:
 - 2.2.2.1. El sistema le muestra las notas de la grabación.
 - 2.3. Si el usuario pulsa sobre la cabecera de una columna de la tabla.
 - 2.3.1. El sistema reordena la tabla en base al atributo pulsado.
 - 2.4. Si el usuario pulsa sobre el icono ☰ puede filtrar las grabaciones:
 - 2.4.1. El sistema muestra una barra de búsqueda en la parte superior de la tabla.
 - 2.4.2. El usuario escribe en la barra de búsqueda.
 - 2.4.3. El sistema muestra las grabaciones que contienen en su identificador el texto introducido
 - 2.5. Si el usuario desmarca el nombre de una columna en la lista de atributos que aparece en la parte superior de la pantalla.
 - 2.5.1. El sistema oculta esa columna de la tabla
 - 2.6. Si el usuario marca el nombre de una columna en la lista de atributos que aparece en la parte superior de la pantalla.
 - 2.6.1. El sistema muestra esa columna en la tabla.
 - 2.7. Si el usuario pulsa en la lista desplegable “Página”:
 - 2.7.1. Si hay más grabaciones que el número indicado en “Filas por página”:
 - 2.7.1.1. El sistema muestra las páginas disponibles.
 - 2.7.1.2. El usuario selecciona el número de la página en la lista desplegable.
 - 2.7.1.3. El sistema muestra las grabaciones de esa página.
 - 2.8. Si el usuario pulsa en la lista desplegable “Filas por página”.
 - 2.8.1. El sistema muestra los valores de filas por página disponibles.
 - 2.8.2. Si el usuario selecciona un valor de la lista:
 - 2.8.2.1. Si el número de grabaciones del grupo del usuario es menor o igual al valor seleccionado:
 - 2.8.2.1.1. El sistema muestra todas las grabaciones en la primera página.
 - 2.8.2.2. Si el número de grabaciones es superior al valor seleccionado:

- 2.8.2.2.1. El sistema muestra el valor seleccionado de grabaciones en la primera página y el resto las divide en tantas páginas como sea necesario.
- 2.9. Si el usuario pulsa en la flecha de navegación izquierda de la parte inferior derecha de la tabla:
- 2.9.1. Si hay más de una página de grabaciones:
 - 2.9.1.1. El sistema mostrará la primera página de grabaciones.
- 2.10. Si el usuario pulsa en la flecha de navegación central izquierda de la parte inferior derecha de la tabla:
- 2.10.1. Si hay más de una página de grabaciones:
 - 2.10.1.1. El sistema mostrará la página de grabaciones anterior a la actual.
- 2.11. Si el usuario pulsa en la flecha de navegación central derecha de la parte inferior derecha de la tabla:
- 2.11.1. Si hay más de una página de grabaciones:
 - 2.11.1.1. El sistema mostrará la página de grabaciones siguiente a la actual.
- 2.12. Si el usuario pulsa en la flecha de navegación derecha de la parte inferior derecha de la tabla:
- 2.12.1. Si hay más de una página de grabaciones:
 - 2.12.1.1. El sistema mostrará la última página de grabaciones.

Configuración

De-Seleccionar todas
 Fecha grabación
 Lugar
 Paciente
 Edad
 Edad grabación
 Grabación
 CSV
 Video
 Atributos extra
 Notas

Grabaciones

<input type="checkbox"/>	Id ↑	Fecha Grabación	Lugar	Paciente	Edad	Edad grabación	Grabación	CSV	Video	Atributos extra	Notas
<input type="checkbox"/>	Sesión 1 Grabación 1	10/03/2018	Hospital Cruces	Han Solo -	66	66	<input type="button" value="🔍"/> <input type="button" value="👁"/> <input type="button" value="📄"/> <input type="button" value="📄"/>	Sesión 1 Grabación 1 V2.csv	Sesión 1 Grabación 1 V2.mp4	<input type="button" value="AÑADIR"/>	<input type="button" value="VER"/>
<input type="checkbox"/>	Sesión 1 Grabación 3	06/04/2018	Hospital Donostia	Sin Asignar -			<input type="button" value="🔍"/> <input type="button" value="👁"/> <input type="button" value="📄"/> <input type="button" value="📄"/>	Sesión 1 Grabación 2.csv	Sesión 1 Grabación 2.mp4	<input type="button" value="AÑADIR"/>	

Página: 1 Filas por página: 10 1 - 2 de 2 < > >>

Ilustración 62. Ver grabaciones

10. Editar nota

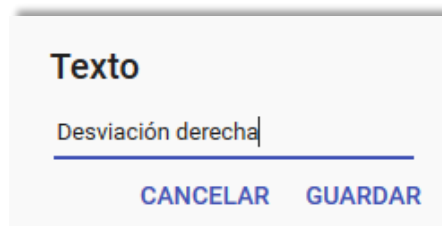
Descripción: Permite al usuario editar el texto de una nota.

Actores: Usuario con sesión iniciada.

Precondiciones: Haber iniciado sesión en el sistema y que haya al menos una grabación con al menos una nota asociada al grupo de usuarios al que pertenece el actor.

Flujo principal:

1. El usuario accede a la sección “Grabaciones”.
2. El usuario pulsa sobre el botón “Ver” de la columna Notas.
3. El sistema muestra todas las notas de la grabación.
4. El usuario pulsa sobre el texto de la grabación que quiere editar.
5. El sistema muestra en una ventana emergente el valor actual del atributo.
6. El usuario escribe el nuevo valor del atributo en la ventana mostrada.
 - 6.1. Si el dato introducido no es correcto el sistema muestra un mensaje de error.
7. El usuario pulsa sobre el botón “Guardar”.
8. El sistema verifica que el nuevo valor del atributo es correcto, lo encripta y actualiza la nota en la base de datos.
 - 8.1. Si hay algún error el sistema muestra un mensaje indicando qué ha ocurrido.



Una ventana emergente con un fondo gris claro y un borde gris. En la parte superior, el título "Texto" está en negrita. Debajo, el texto "Desviación derecha" está escrito en un campo de entrada con un cursor al final. Una línea horizontal azul separa el campo de entrada de los botones. En la parte inferior, los botones "CANCELAR" y "GUARDAR" están escritos en azul.

Ilustración 63. Editar grabación

11. Eliminar nota/s

Descripción: Permite al usuario eliminar una o más notas.

Actores: Usuario con sesión iniciada

Precondiciones: Haber iniciado sesión en el sistema y que haya al menos una grabación con al menos una nota asociada al grupo de usuarios al que pertenece el actor.

Flujo principal:

1. El usuario accede a la sección “Grabaciones”.
2. El usuario pulsa sobre el botón “Ver” de la columna Notas.
3. El sistema muestra todas las notas de la grabación.
4. El usuario selecciona las notas que quiere borrar.
5. El usuario pulsa sobre el icono de la papelera de la esquina superior derecha de la tabla.
 - 5.1. El sistema solicita al usuario que confirme que desea borrar las notas seleccionadas.
 - 5.2. Si el usuario confirma que quiere borrar las notas pulsando el botón “Continuar”:
 - 5.2.1. El sistema elimina las notas de la base de datos. Muestra un mensaje indicando que las notas se han borrado correctamente.
 - 5.2.2. Si ha ocurrido algún error durante el borrado, el sistema informa al usuario con un mensaje de error y vuelve a la vista general de la sección “Grabaciones”.
 - 5.3. Si el usuario cancela el borrado de notas:
 - 5.3.1. El sistema vuelve a la vista general de la sección “Grabaciones”.

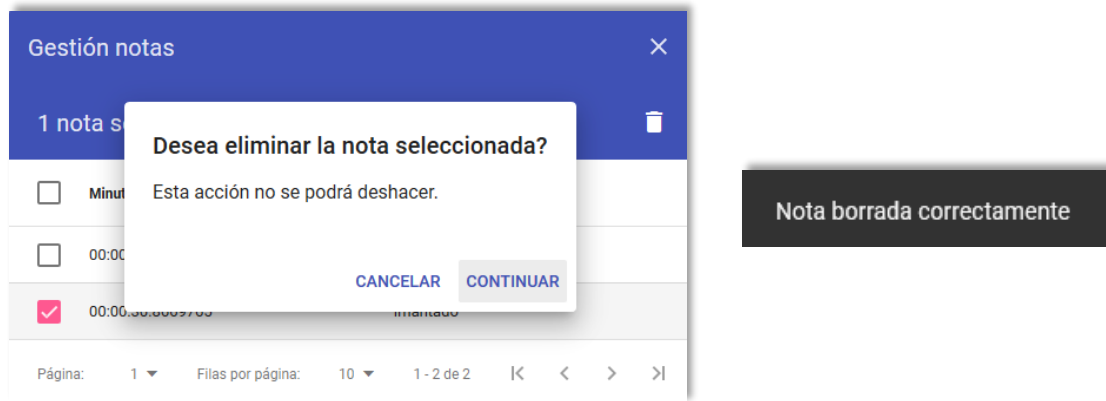


Ilustración 64. Eliminar nota/s

12. Eliminar grabación/es

Descripción: Permite al usuario eliminar una o más grabaciones.

Actores: Usuario con sesión iniciada

Precondiciones: Haber iniciado sesión en el sistema y que haya al menos una grabación asociada al grupo de usuarios al que pertenece el actor.

Flujo principal:

1. El usuario accede a la sección “Grabaciones”.
2. El usuario selecciona las grabaciones que quiere borrar.
3. El usuario pulsa sobre el icono de la papelera de la esquina superior derecha de la tabla.
 - 3.1. El sistema solicita al usuario que confirme que desea borrar las grabaciones seleccionadas.
 - 3.1.1. Si el usuario confirma que quiere borrar las grabaciones pulsando el botón “Continuar”:
 - 3.1.1.1. El sistema elimina las grabaciones de la base de datos. Muestra un mensaje indicando que las grabaciones se han borrado correctamente.
 - 3.1.1.2. Si ha ocurrido algún error durante el borrado, el sistema informa al usuario con un mensaje de error y vuelve a la vista general de la sección “Grabaciones”.
 - 3.1.2. Si el usuario cancela el borrado de grabaciones:
 - 3.1.3. El sistema vuelve a la vista general de la sección “Grabaciones”.



Grabación borrada correctamente

Ilustración 65. Eliminar grabación


13. Visualizar grabación

Descripción: Permite al usuario visualizar una grabación, se visualizará el esqueleto del paciente junto con el vídeo (si está disponible). Además, permitirá tomar notas al usuario.

Actores: Usuario con sesión iniciada

Precondiciones: Haber iniciado sesión en el sistema y que haya al menos una grabación asociada al grupo de usuarios al que pertenece el actor.

Flujo principal:

1. El usuario accede a la sección “Grabaciones”.
2. El usuario pulsa sobre el icono  de la grabación que quiere visualizar.
3. Si la grabación cuenta tanto con fichero CSV como con vídeo:
 - 3.1. El sistema obtiene los ficheros de la nube, los descripta y comienza a reproducir el vídeo y la vista frontal y lateral del esqueleto de forma sincronizada.
 - 3.1.1. Si ha habido algún problema durante el proceso, el sistema muestra el error.
 - 3.1.2. Si hay una nota asociada al segundo de reproducción actual el sistema la muestra sobre el esqueleto durante 5 segundos.
 - 3.2. Si el usuario pulsa sobre el botón de pausa:
 - 3.2.1. El sistema pausa la reproducción.
 - 3.3. Si el usuario pulsa sobre el botón de reproducción:
 - 3.3.1. El sistema retoma la reproducción en el punto en el que se había pausado.
 - 3.4. Si el usuario pulsa sobre cualquier punto de la barra de progreso:
 - 3.4.1. El sistema retoma la reproducción en el minuto asociado al punto seleccionado.
 - 3.5. Si el usuario pulsa sobre cualquier punto del esqueleto:
 - 3.5.1. El sistema pausa el vídeo y muestra una ventana emergente en la que el usuario puede introducir una nota.
 - 3.5.1.1. Si el usuario introduce una nota:
 - 3.5.1.1.1. El sistema la encripta, la guarda en la base de datos y la muestra sobre el esqueleto durante 5 segundos.
 - 3.5.1.2. Se retoma la reproducción.
 - 3.6. Si el usuario pulsa sobre cualquier punto de la barra de control de volumen:
 - 3.6.1. El sistema ajusta el volumen del vídeo.
 - 3.7. Si el usuario pulsa sobre el botón de silenciado:
 - 3.7.1. El sistema silencia el vídeo.
 - 3.8. Si el usuario pulsa sobre el botón de restaurar sonido:
 - 3.8.1. El sistema reproduce el vídeo con sonido.
 - 3.9. Si el usuario desmarca el nombre de una vista en la lista de la parte superior.
 - 3.9.1. El sistema oculta esa vista.
 - 3.10. Si el usuario marca el nombre de una vista en la lista de la parte superior.
 - 3.10.1. El sistema muestra esa columna en la tabla.
4. Si la grabación cuenta solo con el fichero CSV:
 - 4.1. El sistema obtiene el fichero de la nube, lo descripta y comienza a reproducir la vista frontal y lateral del esqueleto de forma sincronizada.

- 4.1.1. Si ha habido algún problema durante el proceso, el sistema muestra el error.
 - 4.1.2. Si hay una nota asociada al segundo de reproducción actual el sistema la muestra sobre el esqueleto durante 5 segundos.
 - 4.2. Si el usuario pulsa sobre el botón de pausa:
 - 4.2.1. El sistema pausa la reproducción.
 - 4.3. Si el usuario pulsa sobre el botón de reproducción:
 - 4.3.1. El sistema retoma la reproducción en el punto en el que se había pausado.
 - 4.4. Si el usuario pulsa sobre cualquier punto de la barra de progreso:
 - 4.4.1. El sistema retoma la reproducción en el minuto asociado al punto seleccionado.
 - 4.5. Si el usuario pulsa sobre cualquier punto del esqueleto:
 - 4.5.1. El sistema pausa la reproducción y muestra una ventana emergente en la que el usuario puede introducir una nota.
 - 4.5.1.1. Si el usuario introduce una nota:
 - 4.5.1.1.1. El sistema la encripta, la guarda en la base de datos y la muestra sobre el esqueleto durante 5 segundos.
 - 4.5.1.2. El sistema retoma la reproducción.
 - 4.6. Si el usuario desmarca el nombre de una vista en la lista de la parte superior.
 - 4.6.1. El sistema oculta esa vista.
 - 4.7. Si el usuario marca el nombre de una vista en la lista de la parte superior.
5. El sistema muestra esa columna en la tabla.

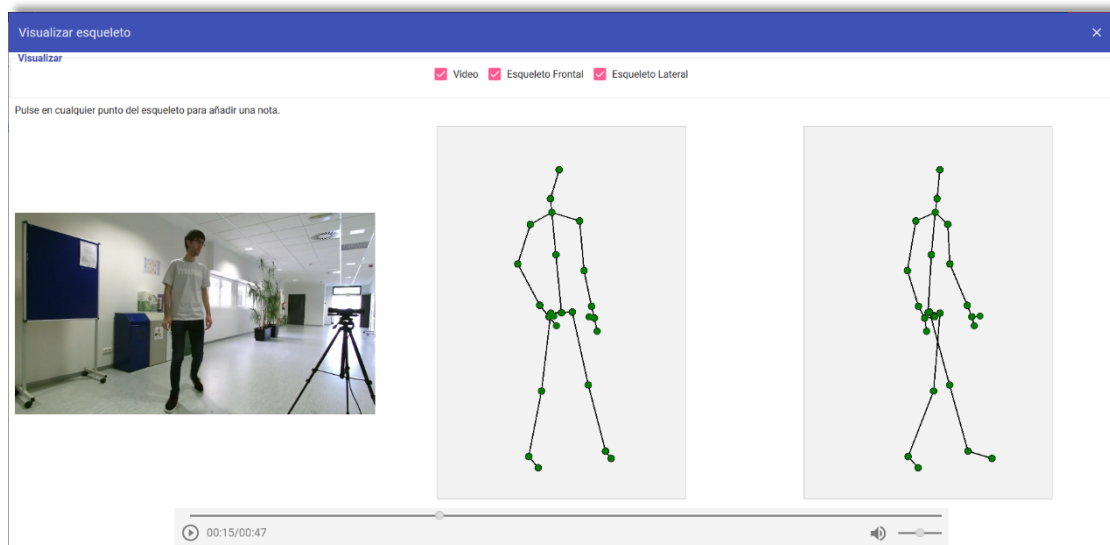


Ilustración 66. Visualizar grabación


14. Descargar grabación

Descripción: Permite al usuario descargar una grabación, la descarga puede hacerse tanto en formato CSV como en JSON.

Actores: Usuario con sesión iniciada

Precondiciones: Haber iniciado sesión en el sistema y que haya al menos una grabación asociada al grupo de usuarios al que pertenece el actor.

Flujo principal:

1. El usuario accede a la sección “Grabaciones”.
2. El usuario pulsa sobre uno de los iconos  de la grabación que quiere descargar.
 - 2.1. Si el usuario pulsa el icono asociado al formato CSV:
 - 2.1.1. El sistema descarga la grabación en el formato CSV.
 - 2.2. Si el usuario pulsa el icono asociado al formato JSON:
 - 2.2.1. El sistema descarga la grabación en el formato JSON.

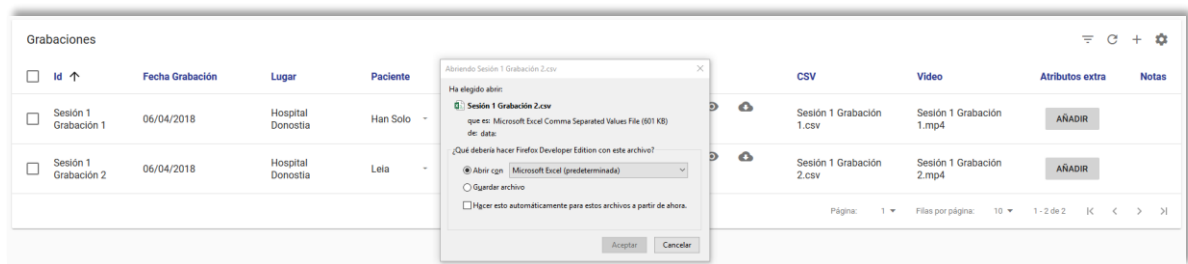


Ilustración 67. Descargar grabación

Anexo C. Pruebas

En este anexo se incluyen el resto de pruebas correspondientes al capítulo [8. Pruebas.](#)

HU1. Usuarios

Como cliente quiero que solamente puedan acceder a la aplicación personas autorizadas, la información solo será accesible una vez se haya iniciado sesión. Los usuarios se crearán de forma manual por el administrador del sistema y habrá grupos de usuarios. Cada grupo representará una institución, como por ejemplo el Hospital Donostia.

1. Comprobar que solo pueden iniciar sesión los usuarios autorizados.

Prueba 1: El usuario intenta iniciar sesión sin rellenar ningún campo del formulario de inicio de sesión. La aplicación no debería permitir al usuario pulsar el botón de inicio de sesión.

Entrada: Vacía.

Salida esperada: El botón de inicio de sesión bloqueado y mensaje de error en los campos sobre los que se haya pulsado.

Salida real: Salida esperada.

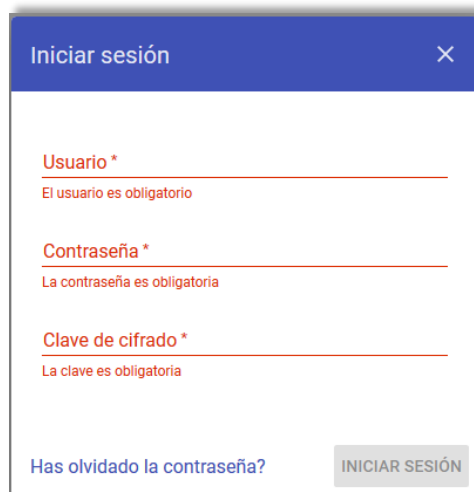


Ilustración 68. Test HU1.1 Prueba 1

Prueba 2: El usuario intenta iniciar sesión sin rellenar el nombre de usuario. La aplicación no debería permitir al usuario pulsar el botón de inicio de sesión.

Entrada: Nombre de usuario vacío, el resto de campos es irrelevante.

Salida esperada: El botón de inicio de sesión bloqueado y mensaje de error en el campo de nombre de usuario y en el resto de campos sobre los que se haya pulsado.

Salida real: Salida esperada.

Iniciar sesión

Usuario *

El usuario es obligatorio

Contraseña *

Clave de cifrado *

Has olvidado la contraseña?

INICIAR SESIÓN

Ilustración 69. Test HU1.1 Prueba 2

Prueba 3: El usuario intenta iniciar sesión sin indicar la contraseña. La aplicación no debería permitir al usuario pulsar el botón de inicio de sesión.

Entrada: Contraseña vacía, el resto de campos es irrelevante.

Salida esperada: El botón de inicio de sesión bloqueado y mensaje de error en el campo de contraseña y en el resto de campos sobre los que se haya pulsado.

Salida real: Salida esperada.

Iniciar sesión

Usuario *

atamayo005@ikasle.ehu.eus

Contraseña *

La contraseña es obligatoria

Clave de cifrado *

●●●●●

Has olvidado la contraseña?

INICIAR SESIÓN

Ilustración 70. Test HU1.1 Prueba 3

Prueba 4: El usuario intenta iniciar sesión sin indicar la clave de cifrado. La aplicación no debería permitir al usuario pulsar el botón de inicio de sesión.

Entrada: Clave de cifrado vacía, el resto de campos es irrelevante.

Salida esperada: El botón de inicio de sesión bloqueado y mensaje de error en el campo de clave de cifrado y en el resto de campos sobre los que se haya pulsado.

Salida real: Salida esperada.

Iniciar sesión

Usuario*
atamayo005@ikasle.ehu.eus

Contraseña*
●●●●●●

Clave de cifrado*
La clave es obligatoria

Has olvidado la contraseña? INICIAR SESIÓN

Ilustración 71. Test HU1.1 Prueba 4

Prueba 5: El usuario rellena el formulario de inicio de sesión completamente, pero introduce el usuario o la contraseña de forma incorrecta.

Entrada: Usuario, contraseña y clave de cifrado. Nombre de usuario o contraseña incorrecta.

Salida esperada: Mensaje de error indicando que el nombre de usuario y/o la contraseña es incorrecta.

Salida real: Salida esperada.

Iniciar sesión

Usuario*
atamayo005@ikasle.ehu.eus

Contraseña*
●●●●●●

Clave de cifrado*
●●●●●●

Usuario y/o contraseña incorrectos

Has olvidado la contraseña? INICIAR SESIÓN

Ilustración 72. Test HU1.1 Prueba 5

Prueba 6: El usuario rellena el formulario de inicio de sesión completo de forma correcta.

Entrada: Usuario, contraseña y clave de cifrado correcta.

Salida esperada: Mensaje dando la bienvenida al usuario.

Salida real: Salida esperada.

Bienvenido atamayo005@ikasle.ehu.eus.

Ilustración 73. Test HU1.1 Prueba 6

2. No se puede acceder a información de pacientes o grabaciones sin iniciar sesión.

Prueba 1: El usuario intenta acceder a la sección de pacientes o grabaciones sin haber iniciado sesión. Puede intentar acceder bien desde el menú o introduciendo la URL correspondiente a la sección. La aplicación no debería permitir al usuario acceder.

Entrada: No hay entrada para esta prueba.

Salida esperada: Mensaje indicando que debe iniciar sesión para poder acceder.

Salida real: Salida esperada.

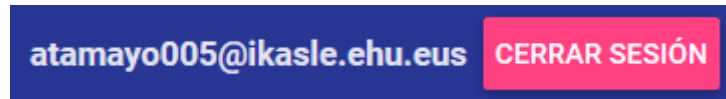


Ilustración 74. Test HU1.2 Prueba 1

Prueba 2: El usuario intenta acceder a la sección de pacientes o grabaciones tras haber iniciado sesión. Puede acceder bien desde el menú o introduciendo la URL correspondiente a la sección. La aplicación debería permitir al usuario acceder.

Entrada: No hay entrada para esta prueba.

Salida esperada: Acceso correcto a la página de pacientes o grabaciones.

Salida real: Salida esperada.

3. Un usuario puede cambiar de contraseña.

Prueba 1: El usuario trata de iniciar sesión, pero ha olvidado la contraseña. Pulsa sobre el enlace *¿Has olvidado la contraseña?* y rellena el formulario con un correo que no está registrado en la base de datos. La aplicación debería mostrar un mensaje de error.

Entrada: Correo que no está registrado en la base de datos.

Salida esperada: Mensaje indicando que hubo un problema al enviar el correo.

Salida real: Salida esperada.

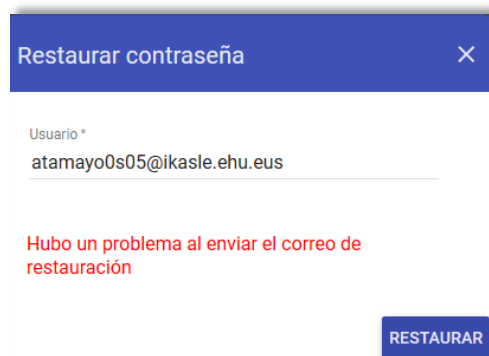


Ilustración 75. Test HU1.3 Prueba 1

Prueba 2: El usuario trata de iniciar sesión, pero ha olvidado la contraseña. Pulsa sobre el enlace *¿Has olvidado la contraseña?* y rellena el formulario con un correo registrado en la base de datos. La aplicación debería enviar un correo electrónico para restaurar la contraseña y mostrar una notificación indicando que se ha enviado con éxito.

Entrada: Correo registrado en la base de datos.

Salida esperada: Mensaje indicando que se ha enviado el correo de recuperación. El usuario recibe el mensaje de recuperación.

Salida real: Salida esperada.



Ilustración 77. Test HU1.3 Prueba 2. Email recuperación

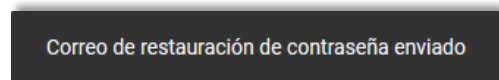


Ilustración 76. Test HU1.3 Prueba 2. Notificación

HU2. Entorno pacientes

Como cliente quiero añadir pacientes para posteriormente ver información sobre ellos y gestionarlos.

1. Añadir un nuevo paciente y comprobar que se ha añadido correctamente a la base de datos.

Prueba 1: El usuario intenta registrar un nuevo paciente sin rellenar ningún campo del formulario. La aplicación no debería permitir al usuario pulsar el botón *Crear paciente*.

Entrada: Vacía.

Salida esperada: El botón *Crear paciente* bloqueado y mensaje de error en los campos pulsados.

Salida real: Salida esperada.

Prueba 2: El usuario intenta registrar un nuevo paciente sin rellenar todos los campos obligatorios. La aplicación no debería permitir al usuario pulsar el botón *Crear paciente*.

Entrada: Algún campo del formulario.

Salida esperada: El botón *Crear paciente* bloqueado y mensaje de error en los campos pulsados.

Salida real: Salida esperada.

Ilustración 78. Test HU2.1 Prueba 1 y Prueba 2

Prueba 3: El usuario intenta registrar un nuevo paciente indicando cómo fecha de nacimiento una fecha posterior a la actual. La aplicación no debería permitir al usuario introducir una fecha posterior.

Entrada: Fecha posterior a la actual.

Salida esperada: El sistema no permite introducir la fecha, en el calendario solo se pueden seleccionar fechas anteriores a la actual y no es posible modificar el campo manualmente.

Salida real: Salida esperada.

Ilustración 79. Test HU2.1 Prueba 3

Prueba 4: El usuario intenta registrar un nuevo paciente indicando cómo año de inicio de la enfermedad un año posterior al actual. La aplicación debería mostrar un error si el usuario introduce un año posterior.

Entrada: Año posterior al actual.

Salida esperada: El sistema muestra un mensaje indicando que el año de inicio de la enfermedad debe ser anterior al año actual.

Salida real: Salida esperada.



Ilustración 80. Test HU2.2 Prueba 4

Prueba 5: El usuario intenta registrar un nuevo paciente introduciendo en el campo de peso o altura caracteres no numéricos. La aplicación mostrará un error si se introducen.

Entrada: Caracteres no numéricos en los campos de altura y peso.

Salida esperada: El sistema muestra un error indicando que espera valores numéricos.

Salida real: Salida esperada.



Ilustración 81. Test HU2.2 Prueba 5

Prueba 6: El usuario rellena los campos obligatorios del formulario de creación de paciente de forma correcta.

Entrada: Identificador, fecha de nacimiento, año de inicio de la enfermedad, clasificación y sexo.

Salida esperada: Paciente registrado en la base de datos y mensaje indicando que el paciente se ha creado correctamente.

Salida real: Salida esperada.

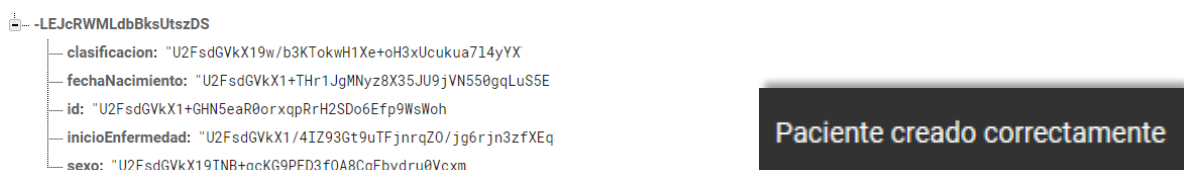


Ilustración 82. Test HU2.2 Prueba 6

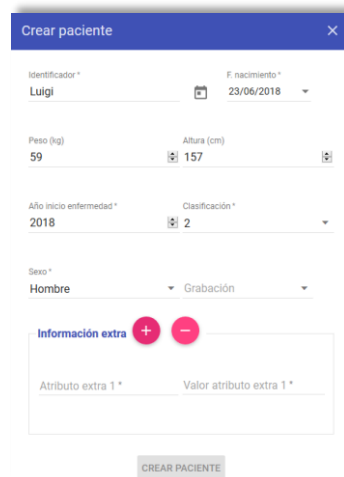
2. Los pacientes pueden incluir tantos atributos extra como se crea conveniente.

Prueba 1: El usuario añade un nuevo campo extra pulsando el botón + de la parte superior izquierda del formulario. No rellena ningún campo. La aplicación no debería permitir al usuario pulsar el botón *Crear paciente*.

Entrada: Identificador, fecha de nacimiento, año de inicio de la enfermedad, clasificación y sexo.

Salida esperada: El botón *Crear paciente* bloqueado.

Salida real: Salida esperada.



The screenshot shows a mobile application form titled 'Crear paciente'. The form contains several input fields: 'Identificador*' with the value 'Luigi', 'F. nacimiento*' with the value '23/06/2018', 'Peso (kg)' with '59', 'Altura (cm)' with '157', 'Año inicio enfermedad*' with '2018', 'Clasificación*' with '2', 'Sexo*' with 'Hombre', and 'Grabación'. Below these fields is a section titled 'Información extra' with a '+' button and a '-' button. Underneath, there are two input fields: 'Atributo extra 1*' and 'Valor atributo extra 1*', both of which are empty. At the bottom of the form is a button labeled 'CREAR PACIENTE'.

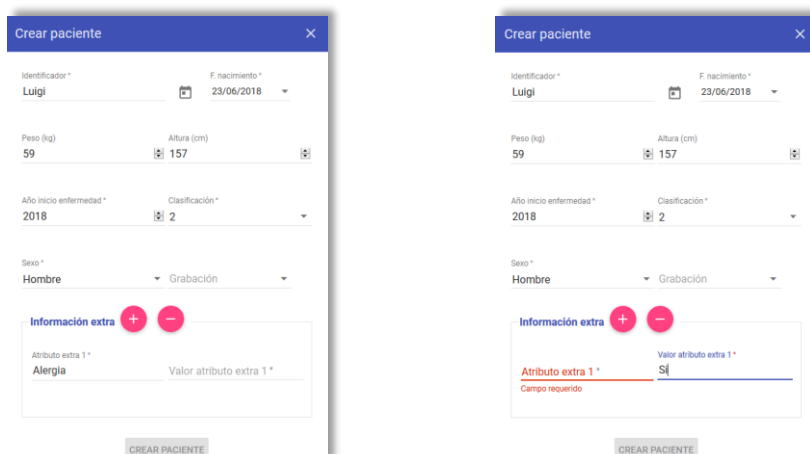
Ilustración 83. Test HU2.2 Prueba 1

Prueba 2: El usuario intenta añadir un nuevo campo extra rellenando únicamente el identificador o el valor. La aplicación no debería permitir al usuario pulsar el botón *Crear paciente*.

Entrada: Identificador, fecha de nacimiento, año de inicio de la enfermedad, clasificación y sexo. Además, el Identificador o valor del nuevo atributo extra.

Salida esperada: El botón *Crear paciente* bloqueado.

Salida real: Salida esperada.



The image shows two side-by-side screenshots of the 'Crear paciente' form. The left screenshot shows the 'Información extra' section with the '+' button highlighted. The 'Atributo extra 1*' field contains the text 'Alergia'. The right screenshot shows the 'Información extra' section with the '-' button highlighted. The 'Valor atributo extra 1*' field contains the text 'SI'. Both screenshots show the 'CREAR PACIENTE' button at the bottom, which is disabled (greyed out).

Ilustración 84. Test HU2.2 Prueba 2

Prueba 3: El usuario rellena los campos obligatorios del formulario de creación de paciente de forma correcta y completa tanto el identificador como el valor del atributo extra.

Entrada: Identificador, fecha de nacimiento, año de inicio de la enfermedad, clasificación, sexo, identificador y valor del nuevo atributo extra.

Salida esperada: Paciente registrado en la base de datos con el atributo extra definido y mensaje indicando que el usuario se ha creado correctamente.

Salida real: Salida esperada.

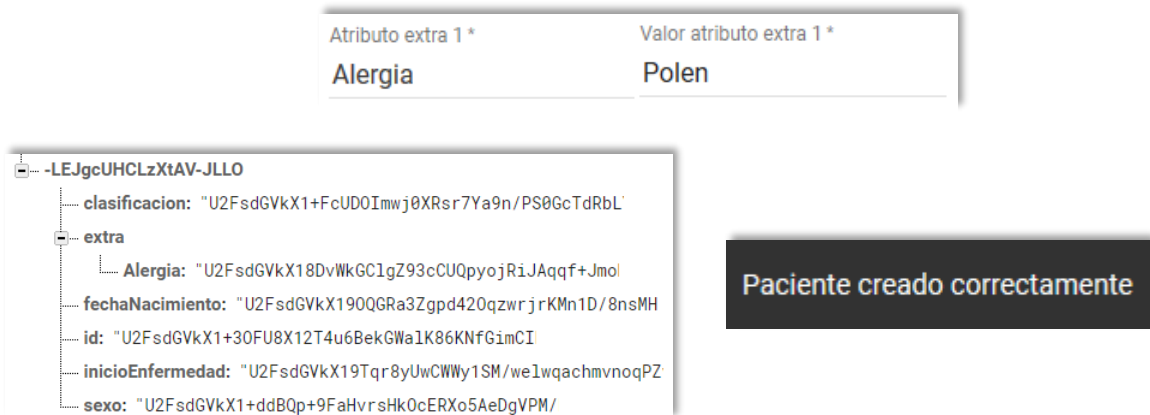


Ilustración 85. Test HU2.2 Prueba 3

Prueba 4: El usuario elimina el último atributo extra creado pulsando sobre el botón con el símbolo - de la parte superior izquierda del formulario.

Entrada: Al menos debe haber un atributo extra creado, el usuario pulsa el botón -.

Salida esperada: El último atributo extra creado es eliminado.

Salida real: Salida esperada.

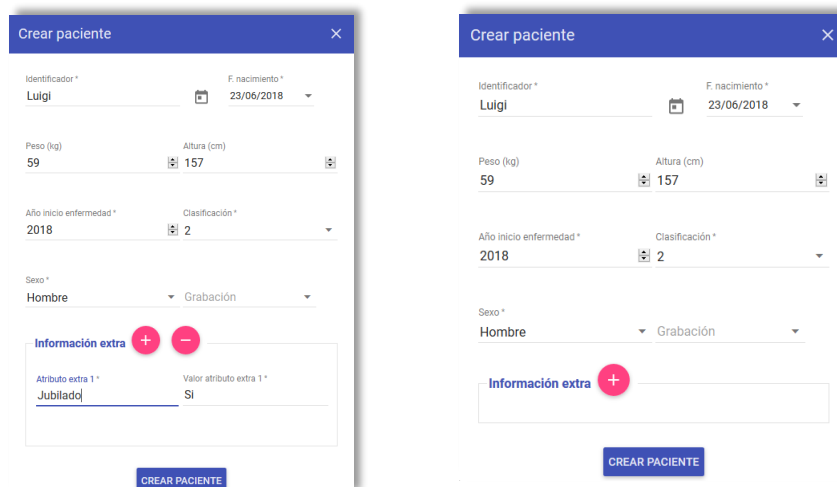


Ilustración 86. Test HU2.2 Prueba 4

3. Asignar una grabación al paciente a la hora de registrarlo.

Prueba 1: El usuario registra un nuevo paciente introduciendo todos los campos obligatorios de forma correcta. Existe al menos una grabación perteneciente al mismo grupo sin paciente asignado. El usuario asigna una grabación al paciente durante el registro. La grabación queda asignada al nuevo paciente.

Entrada: El usuario introduce el identificador, fecha de nacimiento, año de inicio de la enfermedad, clasificación, sexo y grabación de un paciente.

Salida esperada: El nuevo paciente se crea correctamente y la grabación queda asignada a ese paciente.

Salida real: Salida esperada.

The image shows a 'Crear paciente' form on the left and a table of recordings on the right. The form has the following fields:

- Identificador*: Luiqi
- F. nacimiento*: 23/06/2018
- Peso (kg): 59
- Altura (cm): 157
- Año inicio enfermedad*: 2018
- Clasificación*: 2
- Sexo*: Hombre
- Grabación: Sesión 3 Grabación 1
- Información extra: (empty field with a red plus icon)
- CREAR PACIENTE button

The table on the right is titled 'Grabaciones' and has the following columns: Id, Fecha Grabación, Lugar, Paciente, Edad, and Edad grabación.

<input type="checkbox"/>	Id ↑	Fecha Grabación	Lugar	Paciente	Edad	Edad grabación
<input type="checkbox"/>	Sesión 1 Grabación 1	06/04/2018	Hospital Donostia	Paquita Salas -	29	29
<input type="checkbox"/>	Sesión 1 Grabación 2	06/04/2018	Hospital Donostia	Luke -	73	73

Ilustración 87. Test HU2.3 Prueba 1

4. Comprobar que los nuevos pacientes solamente son visibles para los usuarios que pertenecen al mismo grupo que el usuario que lo ha añadido.

Prueba 1: El usuario1 perteneciente al grupo1 introduce un nuevo paciente. El nuevo paciente es visible para usuario1. El usuario2, perteneciente a un grupo distinto al grupo1, inicia sesión y accede a la sección *Pacientes*. El paciente creado por usuario1 no es visible para el usuario2.

Entrada: Usuario1 introduce el identificador, fecha de nacimiento, año de inicio de la enfermedad, clasificación y sexo de un paciente.

Salida esperada: El usuario1 puede ver el paciente creado por él mismo, mientras que el usuario2 no tiene acceso al nuevo paciente.

Salida real: Salida esperada.

Crear paciente [X]

Identificador* F. nacimiento*

Peso (kg) Altura (cm)

Año inicio enfermedad* Clasificación*

Sexo* Grabación

Información extra

CREAR PACIENTE

Pacientes

Identificador	Fecha Nacimiento	Edad	Sexo	Altura(cm)	Peso	Inicio enfermedad	Clasificación	Grabaciones	Atributos extra
<input type="checkbox"/> Han Solo	24/05/1959	59	Hombre -	180	Añadir peso	2000	2	-	VER AÑADIR
<input type="checkbox"/> Leia	21/09/1979	38	Mujer -	Añadir altura	65	2012	3	-	VER AÑADIR
<input type="checkbox"/> Paquita Salas	25/03/1989	29	Mujer -	175	90	2016	3	-	VER AÑADIR
<input type="checkbox"/> Snape	12/11/1960	57	Hombre -	182	78	2015	2	-	AÑADIR

Página: 1 Filas por página: 10 1-4 de 4

Ilustración 88. Test HU2.4 Prueba 1. Paciente creado por usuario1

Pacientes

Identificador	Fecha Nacimiento	Edad	Sexo	Altura(cm)	Peso	Inicio enfermedad	Clasificación	Grabaciones	Atributos extra
<input type="checkbox"/> Hulk	13/08/1976	41	Otro -	210	150	2012	2	-	VER AÑADIR
<input type="checkbox"/> Link	06/03/1981	37	Hombre -	174	87	2018	1	-	VER VER
<input type="checkbox"/> Mario	15/09/1960	57	Hombre -	Añadir altura	Añadir peso	2018	3	-	VER VER

Página: 1 Filas por página: 10 1-3 de 3

Ilustración 89. Test HU2.4. Prueba 1. Sección pacientes del usuario2

Prueba 2: El usuario1 perteneciente al grupo1 introduce un nuevo paciente. El usuario2, perteneciente al mismo grupo, inicia sesión y accede a la sección *Pacientes*. El paciente creado por usuario1 es visible para el usuario2, además del resto de pacientes registrados en ese grupo.

Entrada: Usuario1 introduce el identificador, fecha de nacimiento, año de inicio de la enfermedad, clasificación y sexo de un paciente.

Salida esperada: Tanto el usuario1 como el usuario2 tienen acceso al nuevo paciente y al resto de pacientes de ese grupo.

Salida real: Salida esperada.

Crear paciente [X]

Identificador* F. nacimiento*

Peso (kg) Altura (cm)

Año inicio enfermedad* Clasificación*

Sexo* Grabación

Información extra

CREAR PACIENTE

Pacientes

Identificador	Fecha Nacimiento	Edad	Sexo	Altura(cm)	Peso	Inicio enfermedad	Clasificación	Grabaciones	Atributos extra
<input type="checkbox"/> Han Solo	24/05/1959	59	Hombre -	180	Añadir peso	2000	2	-	VER AÑADIR
<input type="checkbox"/> Leia	21/09/1979	38	Mujer -	Añadir altura	65	2012	3	-	VER AÑADIR
<input type="checkbox"/> Paquita Salas	25/03/1989	29	Mujer -	175	90	2016	3	-	VER AÑADIR
<input type="checkbox"/> Snape	12/11/1960	57	Hombre -	182	78	2015	2	-	AÑADIR

Página: 1 Filas por página: 10 1-4 de 4

Ilustración 91. Test HU2.4 Prueba 2. Paciente creado por usuario1

Pacientes

Identificador	Fecha Nacimiento	Edad	Sexo	Altura(cm)	Peso	Inicio enfermedad	Clasificación	Grabaciones	Atributos extra
<input type="checkbox"/> Han Solo	24/05/1959	59	Hombre -	180	Añadir peso	2000	2	-	VER AÑADIR
<input type="checkbox"/> Leia	21/09/1979	38	Mujer -	Añadir altura	65	2012	3	-	VER AÑADIR
<input type="checkbox"/> Paquita Salas	25/03/1989	29	Mujer -	175	90	2016	3	-	VER AÑADIR
<input type="checkbox"/> Snape	12/11/1960	57	Hombre -	182	78	2015	2	-	AÑADIR

Página: 1 Filas por página: 10 1-4 de 4

Ilustración 90. Test HU2.4 Prueba 2. Sección pacientes del usuario2

5. *Eliminar un paciente y comprobar que se ha eliminado en la base de datos y no se muestra más en la tabla.*

Prueba 1: El usuario selecciona un paciente para ser eliminado, pero cancela su borrado. Para ello selecciona el paciente pulsando en el cuadrado que se encuentra en el lateral izquierdo del paciente y seguidamente pulsa sobre el icono con forma de papelera. El usuario cancela el borrado del paciente pulsando en el botón *Cancelar*.

Entrada: El usuario selecciona un paciente para ser eliminado, pero cuando el sistema solicita su confirmación el usuario cancela el borrado.

Salida esperada: Los pacientes se mantienen intactos.

Salida real: Salida esperada.



Ilustración 92. Test HU2.5 Prueba 1

Prueba 2: El usuario selecciona un paciente y lo elimina. Para ello selecciona el paciente pulsando en el cuadrado que se encuentra en el lateral izquierdo del paciente y seguidamente pulsa sobre el icono con forma de papelera. El usuario pulsa el botón *Continuar*.

Entrada: El usuario selecciona un paciente para ser eliminado, y confirma que desea eliminarlo.

Salida esperada: El paciente seleccionado es borrado de la base de datos, no aparece en la tabla.

Salida real: Salida esperada.



Ilustración 93. Test HU2.5 Prueba 2

6. *Eliminar varios pacientes y comprobar que se han eliminado en la base de datos y no se muestran más en la tabla.*

Prueba 1: El usuario selecciona varios pacientes y los elimina. Para ello selecciona los pacientes pulsando en los cuadrados que se encuentra en el lateral izquierdo de cada paciente y seguidamente pulsa sobre el icono con forma de papelera. El usuario pulsa el botón *Continuar*.

Entrada: El usuario selecciona dos pacientes para ser eliminados, y confirma que desea eliminarlos.

Salida esperada: Los pacientes seleccionados son borrados de la base de datos, no aparecen en la tabla.

Salida real: Salida esperada.



Ilustración 94. Test HU2.6 Prueba 1

7. *Eliminar un paciente con grabación asociada y comprobar que la grabación queda sin asignar tras borrar el paciente.*

Prueba 1: El usuario selecciona un paciente que tiene al menos una grabación asociada y lo elimina. Para ello selecciona el paciente pulsando en el cuadrado que se encuentra en el lateral izquierdo del paciente y seguidamente pulsa sobre el icono con forma de papelera. El usuario pulsa el botón *Continuar*.

Entrada: El usuario selecciona un paciente para ser eliminado, y confirma que desea eliminarlo.

Salida esperada: El paciente seleccionado (Luke) es borrado de la base de datos, no aparece en la tabla. La grabación o grabaciones asignadas al paciente pasan a estar Sin Asignar.

Salida real: Salida esperada.

Grabaciones				
<input type="checkbox"/>	Id ↑	Fecha Grabación	Lugar	Paciente
<input type="checkbox"/>	Sesión 1 Grabación 1	06/04/2018	Hospital Donostia	Leia
<input type="checkbox"/>	Sesión 1 Grabación 2	06/04/2018	Hospital Donostia	Luke

Grabaciones				
<input type="checkbox"/>	Id ↑	Fecha Grabación	Lugar	Paciente
<input type="checkbox"/>	Sesión 1 Grabación 1	06/04/2018	Hospital Donostia	Leia
<input type="checkbox"/>	Sesión 1 Grabación 2	06/04/2018	Hospital Donostia	Sin Asignar

Ilustración 95. Test HU2.7 Prueba 1

8. Editar todos los campos y comprobar que los cambios se ven reflejados en la base de datos.

Prueba 1: El usuario edita el identificador de un paciente. Para ello pulsa sobre el identificador, se muestra una ventana con el identificador actual y el usuario lo edita. El usuario pulsa el botón *Guardar*.

Entrada: El nuevo identificador del paciente.

Salida esperada: Se actualiza el identificador del paciente.

Salida real: Salida esperada.

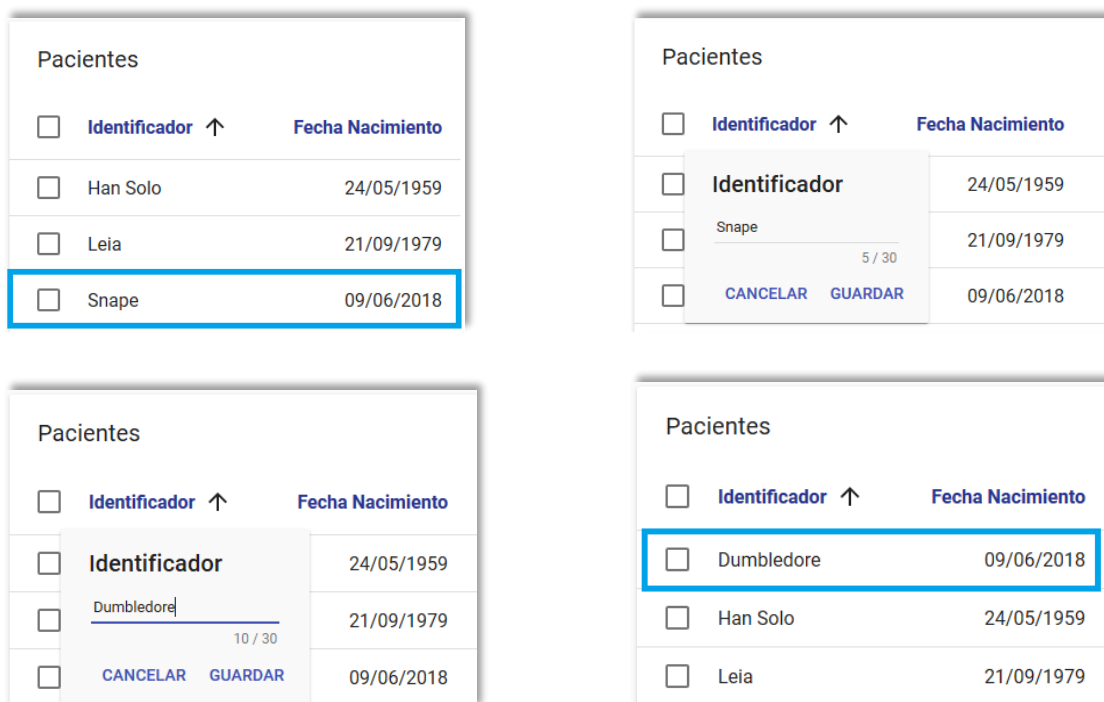


Ilustración 96. Test HU2.8 Prueba 1

Prueba 2: El usuario edita la fecha de nacimiento de un paciente. Para ello pulsa sobre la fecha, se muestra una ventana con la fecha de nacimiento del paciente y el usuario la edita. El usuario pulsa el botón *Guardar*.

Entrada: La nueva fecha de nacimiento del paciente.

Salida esperada: Se actualiza la fecha de nacimiento del paciente.

Salida real: Salida esperada.

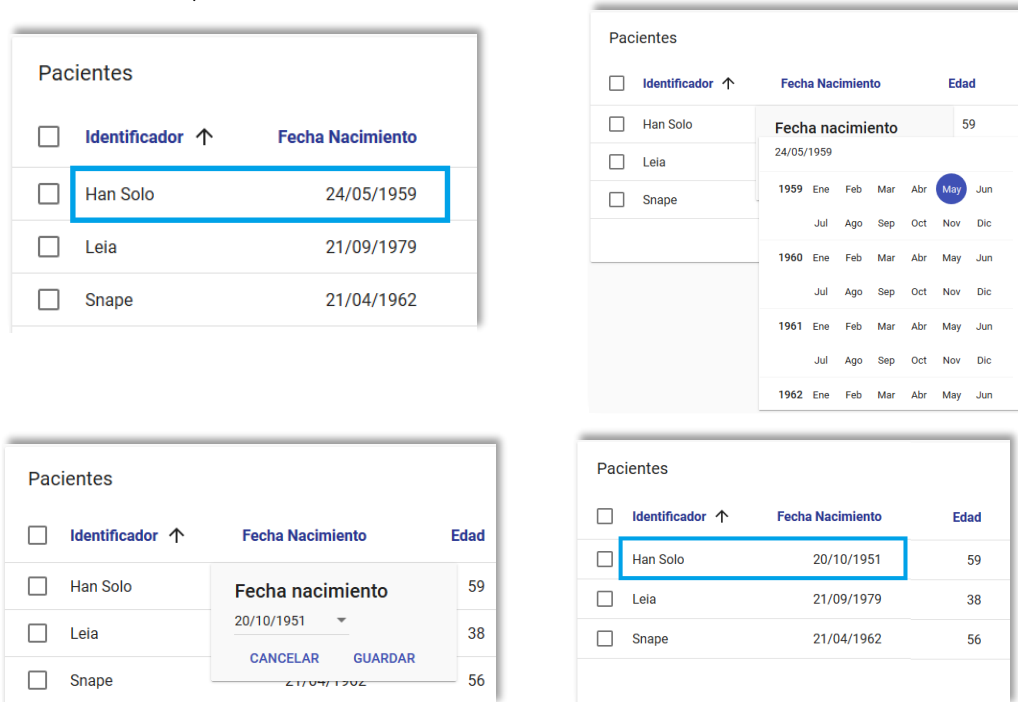


Ilustración 97. Test HU2.8 Prueba 2

Prueba 3: El usuario edita el sexo de un paciente. Para ello pulsa sobre el sexo, se muestra una lista desplegable con el valor actual seleccionado y el usuario selecciona otro valor.

Entrada: El nuevo sexo del paciente.

Salida esperada: Se actualiza el sexo del paciente.

Salida real: Salida esperada.

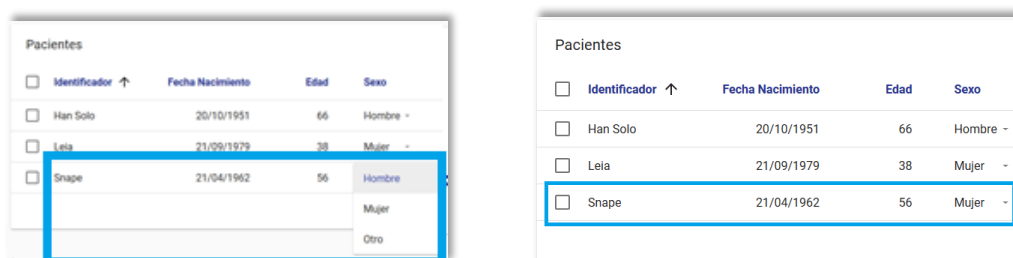


Ilustración 98. Test HU2.8 Prueba 3

Prueba 4: El usuario edita la altura de un paciente. Para ello pulsa sobre la altura, se muestra una ventana emergente con el valor actual de la altura (si no se ha registrado la altura del paciente todavía, aparece vacío) y el usuario introduce el nuevo valor.

Entrada: La nueva altura del paciente.

Salida esperada: Se actualiza la altura del paciente.

Salida real: Salida esperada.

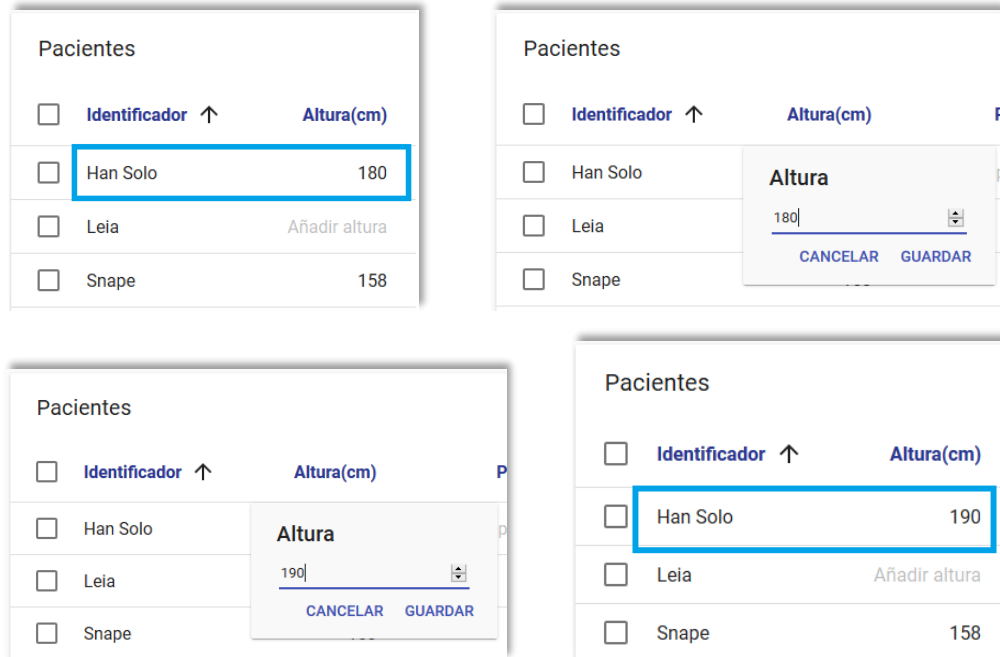


Ilustración 99. Test HU2.8 Prueba 4

Prueba 5: El usuario edita el peso de un paciente. Para ello pulsa sobre el peso, se muestra una ventana emergente con el valor actual del peso (si no se ha registrado el peso del paciente todavía, aparece vacío) y el usuario introduce el nuevo valor.

Entrada: El nuevo peso del paciente.

Salida esperada: Se actualiza el peso del paciente.

Salida real: Salida esperada.

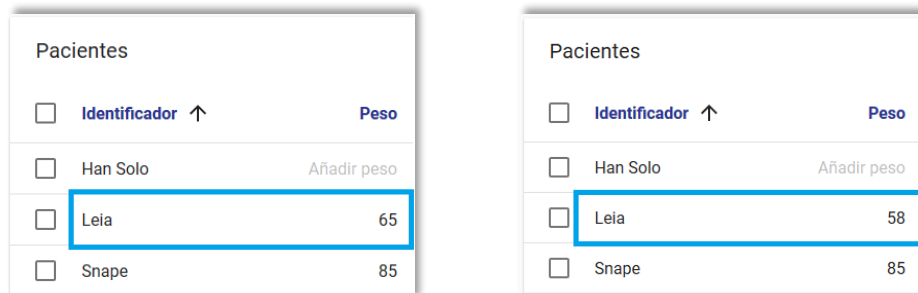


Ilustración 100. Test HU2.8 Prueba 5

Prueba 6: El usuario edita el año de inicio de la enfermedad de un paciente. Para ello pulsa sobre la fecha, se muestra una ventana con el año de inicio de la enfermedad del paciente y el usuario la edita. El usuario pulsa el botón *Guardar*.

Entrada: El nuevo año de inicio de la enfermedad del paciente.

Salida esperada: Se actualiza el año de inicio de la enfermedad del paciente.

Salida real: Salida esperada.

The image shows two side-by-side screenshots of a web application interface titled 'Pacientes'. Each screenshot displays a table with columns for 'Identificador' (checkbox), 'Identificador' (text), and 'Inicio enfermedad' (text). The table contains three rows: Han Solo (2000), Leia (2012), and Snape (2018 in the first screenshot, 2015 in the second). A blue rectangular box highlights the 'Inicio enfermedad' cell for the 'Snape' row in both screenshots, indicating the data being edited.

<input type="checkbox"/>	Identificador ↑	Inicio enfermedad
<input type="checkbox"/>	Han Solo	2000
<input type="checkbox"/>	Leia	2012
<input type="checkbox"/>	Snape	2018

<input type="checkbox"/>	Identificador ↑	Inicio enfermedad
<input type="checkbox"/>	Han Solo	2000
<input type="checkbox"/>	Leia	2012
<input type="checkbox"/>	Snape	2015

Ilustración 101. Test HU2.8 Prueba 6

Prueba 7: El usuario edita la clasificación de un paciente. Para ello pulsa sobre la clasificación, se muestra una lista desplegable con el valor actual seleccionado y el usuario selecciona el nuevo valor.

Entrada: La nueva clasificación del paciente.

Salida esperada: Se actualiza la clasificación del paciente.

Salida real: Salida esperada.


The image shows two side-by-side screenshots of a web application interface titled 'Pacientes'. Each screenshot displays a table with columns for 'Identificador' (checkbox), 'Identificador' (text), and 'Clasificación' (text with a dropdown arrow). The table contains three rows: Han Solo (2), Leia (3 in the first screenshot, 1 in the second), and Snape (2). A blue rectangular box highlights the 'Clasificación' cell for the 'Leia' row in both screenshots, indicating the data being edited.

<input type="checkbox"/>	Identificador ↑	Clasificación
<input type="checkbox"/>	Han Solo	2
<input type="checkbox"/>	Leia	3
<input type="checkbox"/>	Snape	2

<input type="checkbox"/>	Identificador ↑	Clasificación
<input type="checkbox"/>	Han Solo	2
<input type="checkbox"/>	Leia	1
<input type="checkbox"/>	Snape	2

Ilustración 102. Test HU2.8 Prueba 7

9. Filtrar los pacientes por el identificador.

Prueba 1: El usuario filtra los pacientes por el identificador introduciendo un identificador que no tiene ningún paciente. Para ello pulsa sobre el icono , se muestra una barra de búsqueda e introduce el identificador en la barra.

Entrada: Un identificador que no coincida con el de ningún paciente.

Salida esperada: La lista de pacientes vacía.

Salida real: Salida esperada.

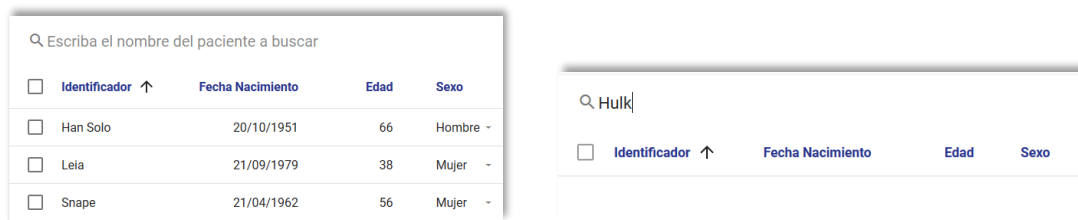



Ilustración 103. Test HU2.9 Prueba 1

Prueba 2: El usuario filtra los pacientes por el identificador introduciendo el identificador de un paciente. Para ello pulsa sobre el icono , se muestra una barra de búsqueda e introduce el identificador en la barra.

Entrada: El identificador de un paciente.

Salida esperada: El paciente que se corresponde con el identificador introducido.

Salida real: Salida esperada.

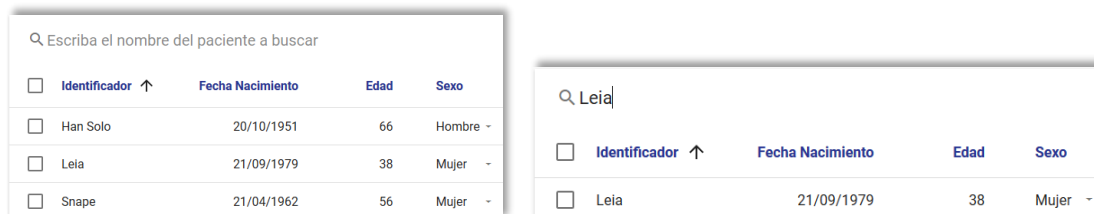



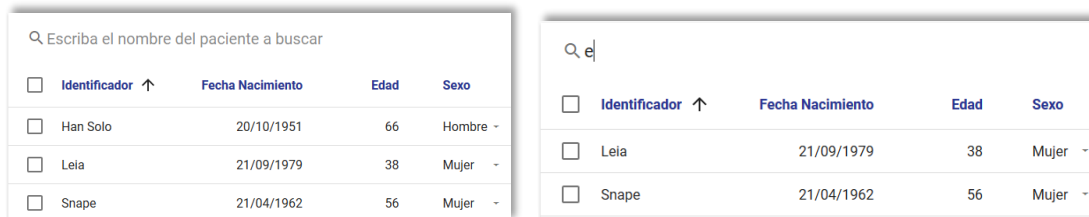
Ilustración 104. Test HU2.9 Prueba 2

Prueba 3: El usuario filtra los pacientes por el identificador introduciendo un texto que forma parte del identificador de varios pacientes. Para ello pulsa sobre el icono  , se muestra una barra de búsqueda e introduce el texto.

Entrada: Texto que coincide con parte del identificador de varios pacientes.

Salida esperada: Los pacientes que tienen en su identificador el texto introducido.

Salida real: Salida esperada.



<input type="checkbox"/>	Identificador ↑	Fecha Nacimiento	Edad	Sexo
<input type="checkbox"/>	Han Solo	20/10/1951	66	Hombre -
<input type="checkbox"/>	Leia	21/09/1979	38	Mujer -
<input type="checkbox"/>	Snape	21/04/1962	56	Mujer -

<input type="checkbox"/>	Identificador ↑	Fecha Nacimiento	Edad	Sexo
<input type="checkbox"/>	Leia	21/09/1979	38	Mujer -
<input type="checkbox"/>	Snape	21/04/1962	56	Mujer -

Ilustración 105. Test HU2.9 Prueba 3

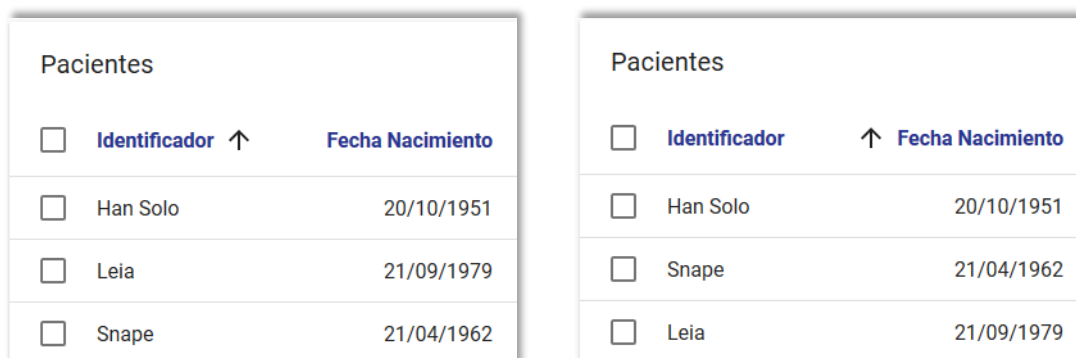
10. Reordenar la tabla por columnas

Prueba 1: El usuario intenta reordenar la tabla en base a la fecha de nacimiento. El sistema reordena la tabla.

Entrada: El usuario pulsa sobre la cabecera de la columna *Fecha de nacimiento*.

Salida esperada: La tabla reordenada en base a la fecha de nacimiento.

Salida real: Salida esperada.



<input type="checkbox"/>	Identificador ↑	Fecha Nacimiento
<input type="checkbox"/>	Han Solo	20/10/1951
<input type="checkbox"/>	Leia	21/09/1979
<input type="checkbox"/>	Snape	21/04/1962

<input type="checkbox"/>	Identificador	↑ Fecha Nacimiento
<input type="checkbox"/>	Han Solo	20/10/1951
<input type="checkbox"/>	Snape	21/04/1962
<input type="checkbox"/>	Leia	21/09/1979

Ilustración 106. Test HU2.10 Prueba 1

Prueba 2: El usuario intenta reordenar la tabla en base a la edad. El sistema reordena la tabla.

Entrada: El usuario pulsa sobre la cabecera de la columna *Edad*.

Salida esperada: La tabla reordenada en base a la edad.

Salida real: Salida esperada.

<input type="checkbox"/> Identificador	↑ Edad
<input type="checkbox"/> Han Solo	66
<input type="checkbox"/> Leia	38
<input type="checkbox"/> Snape	56

<input type="checkbox"/> Identificador	↑ Edad
<input type="checkbox"/> Leia	38
<input type="checkbox"/> Snape	56
<input type="checkbox"/> Han Solo	66

Ilustración 107. Test HU2.10 Prueba 2

Prueba 3: El usuario intenta reordenar la tabla en base al sexo. El sistema reordena la tabla.

Entrada: El usuario pulsa sobre la cabecera de la columna *Sexo*.

Salida esperada: La tabla reordenada en base al sexo.

Salida real: Salida esperada.

<input type="checkbox"/> Identificador	↑ Sexo
<input type="checkbox"/> Han Solo	Hombre ▾
<input type="checkbox"/> Leia	Mujer ▾
<input type="checkbox"/> Snape	Mujer ▾

<input type="checkbox"/> Identificador	Sexo ↓
<input type="checkbox"/> Leia	Mujer ▾
<input type="checkbox"/> Snape	Mujer ▾
<input type="checkbox"/> Han Solo	Hombre ▾

Ilustración 108. Test HU2.10 Prueba 3

Prueba 4: El usuario intenta reordenar la tabla en base a la altura. El sistema reordena la tabla.

Entrada: El usuario pulsa sobre la cabecera de la columna *Altura*.

Salida esperada: La tabla reordenada en base a la altura.

Salida real: Salida esperada.

<input type="checkbox"/> Identificador ↑	Altura(cm)
<input type="checkbox"/> Han Solo	190
<input type="checkbox"/> Leia	Añadir altura
<input type="checkbox"/> Snape	158

<input type="checkbox"/> Identificador	↑ Altura(cm)
<input type="checkbox"/> Snape	158
<input type="checkbox"/> Han Solo	190
<input type="checkbox"/> Leia	Añadir altura

Ilustración 109. Test HU2.10 Prueba 4

Prueba 5: El usuario intenta reordenar la tabla en base al peso. El sistema reordena la tabla.

Entrada: El usuario pulsa sobre la cabecera de la columna *Peso*.

Salida esperada: La tabla reordenada en base al peso.

Salida real: Salida esperada.

<input type="checkbox"/> Identificador ↑	Peso
<input type="checkbox"/> Han Solo	Añadir peso
<input type="checkbox"/> Leia	58
<input type="checkbox"/> Snape	85

<input type="checkbox"/> Identificador	↑ Peso
<input type="checkbox"/> Leia	58
<input type="checkbox"/> Snape	85
<input type="checkbox"/> Han Solo	Añadir peso

Ilustración 110. Test HU2.10 Prueba 5

Prueba 6: El usuario intenta reordenar la tabla en base al año de inicio de la enfermedad. El sistema reordena la tabla.

Entrada: El usuario pulsa sobre la cabecera de la columna *Inicio Enfermedad*.

Salida esperada: La tabla reordenada en base al inicio de la enfermedad.

Salida real: Salida esperada.

<input type="checkbox"/> Identificador	↑ Inicio enfermedad
<input type="checkbox"/> Han Solo	2000
<input type="checkbox"/> Leia	2012
<input type="checkbox"/> Snape	2015

<input type="checkbox"/> Identificador	↑ Inicio enfermedad
<input type="checkbox"/> Han Solo	2000
<input type="checkbox"/> Leia	2012
<input type="checkbox"/> Snape	2015

Ilustración 111. Test HU2.10 Prueba 6

Prueba 7: El usuario intenta reordenar la tabla en base a la clasificación. El sistema reordena la tabla.

Entrada: El usuario pulsa sobre la cabecera de la columna *Clasificación*.

Salida esperada: La tabla reordenada en base a la clasificación.

Salida real: Salida esperada.

<input type="checkbox"/> Identificador	↑ Clasificación
<input type="checkbox"/> Han Solo	2
<input type="checkbox"/> Leia	1
<input type="checkbox"/> Snape	2

<input type="checkbox"/> Identificador	Clasificación	↑
<input type="checkbox"/> Leia	1	▾
<input type="checkbox"/> Han Solo	2	▾
<input type="checkbox"/> Snape	2	▾

Ilustración 112. Test HU2.10 Prueba 7

11. Ocultar columnas

Prueba 1: El usuario intenta ocultar la columna *Fecha de nacimiento* de la tabla

Entrada: El usuario deselecciona *Fecha de nacimiento* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Fecha de nacimiento*.

Salida real: Salida esperada.

The image shows two screenshots of a web application interface. The top screenshot shows the 'Configuración' menu with 'Fecha nacimiento' checked. The table below it has columns for 'Identificador', 'Fecha Nacimiento', 'Edad', 'Sexo', 'Altura(cm)', 'Peso', 'Inicio enfermedad', 'Clasificación', 'Grabaciones', and 'Atributos extra'. The bottom screenshot shows 'Fecha nacimiento' unchecked in the configuration menu, and the 'Fecha Nacimiento' column is hidden in the table.

Identificador	Fecha Nacimiento	Edad	Sexo	Altura(cm)	Peso	Inicio enfermedad	Clasificación	Grabaciones	Atributos extra
Han Solo	20/10/1951	66	Hombre	190	Añadir peso	2000	2	-	VER
Leia	21/09/1979	38	Mujer	Añadir altura	58	2012	1	VER	AÑADIR
Snape	21/04/1962	56	Mujer	158	85	2015	2	-	AÑADIR

Ilustración 113. Test HU2.11 Prueba 1

Prueba 2: El usuario intenta ocultar la columna *Edad* de la tabla

Entrada: El usuario deselecciona *Edad* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Edad*.

Salida real: Salida esperada.

The screenshot shows the 'Configuración' menu with 'Edad' unchecked. The table below it has columns for 'Identificador', 'Fecha Nacimiento', 'Sexo', 'Altura(cm)', 'Peso', 'Inicio enfermedad', 'Clasificación', 'Grabaciones', and 'Atributos extra'. The 'Edad' column is hidden.

Identificador	Fecha Nacimiento	Sexo	Altura(cm)	Peso	Inicio enfermedad	Clasificación	Grabaciones	Atributos extra
Han Solo	20/10/1951	Hombre	190	Añadir peso	2000	2	-	VER
Leia	21/09/1979	Mujer	Añadir altura	58	2012	1	VER	AÑADIR
Snape	21/04/1962	Mujer	158	85	2015	2	-	AÑADIR

Ilustración 114. Test HU2.11 Prueba 2

Prueba 3: El usuario intenta ocultar la columna *Sexo* de la tabla

Entrada: El usuario deselecciona *Sexo* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Sexo*.

Salida real: Salida esperada.



The screenshot shows the 'Configuración' (Configuration) menu at the top with the following options: Seleccionar todas, Fecha nacimiento, Edad, Sexo, Altura, Peso, Inicio enfermedad, Clasificación, Grabaciones, Atributos extra. Below the menu is a table titled 'Pacientes' with the following columns: Identificador, Fecha Nacimiento, Edad, Altura(cm), Peso, Inicio enfermedad, Clasificación, Grabaciones, and Atributos extra. The table contains three rows of patient data.

<input type="checkbox"/>	Identificador ↑	Fecha Nacimiento	Edad	Altura(cm)	Peso	Inicio enfermedad	Clasificación	Grabaciones	Atributos extra
<input type="checkbox"/>	Han Solo	20/10/1951	66	190	Añadir peso	2000	2	-	VER
<input type="checkbox"/>	Leia	21/09/1979	38	Añadir altura	58	2012	1	-	VER
<input type="checkbox"/>	Snape	21/04/1962	56	158	85	2015	2	-	AÑADIR

Ilustración 115. Test HU2.11 Prueba 3

Prueba 4: El usuario intenta ocultar la columna *Altura* de la tabla

Entrada: El usuario deselecciona *Altura* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Altura*.

Salida real: Salida esperada.



The screenshot shows the 'Configuración' (Configuration) menu with the following options: Seleccionar todas, Fecha nacimiento, Edad, Sexo, Altura, Peso, Inicio enfermedad, Clasificación, Grabaciones, Atributos extra. Below the menu is a table titled 'Pacientes' with the following columns: Identificador, Fecha Nacimiento, Edad, Sexo, Peso, Inicio enfermedad, Clasificación, Grabaciones, and Atributos extra. The table contains three rows of patient data.

<input type="checkbox"/>	Identificador ↑	Fecha Nacimiento	Edad	Sexo	Peso	Inicio enfermedad	Clasificación	Grabaciones	Atributos extra
<input type="checkbox"/>	Han Solo	20/10/1951	66	Hombre -	Añadir peso	2000	2	-	VER
<input type="checkbox"/>	Leia	21/09/1979	38	Mujer -	58	2012	1	-	VER
<input type="checkbox"/>	Snape	21/04/1962	56	Mujer -	85	2015	2	-	AÑADIR

Ilustración 116. Test HU2.11 Prueba 4

Prueba 5: El usuario intenta ocultar la columna *Peso* de la tabla

Entrada: El usuario deselecciona *Peso* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Peso*.

Salida real: Salida esperada.

Configuración

Seleccionar todas Fecha nacimiento Edad Sexo Altura Peso Inicio enfermedad Clasificación Grabaciones Atributos extra

Pacientes

<input type="checkbox"/>	Identificador ↑	Fecha Nacimiento	Edad	Sexo	Altura(cm)	Inicio enfermedad	Clasificación	Grabaciones	Atributos extra
<input type="checkbox"/>	Han Solo	20/10/1951	66	Hombre -	190	2000	2	-	VER
<input type="checkbox"/>	Leia	21/09/1979	38	Mujer -	Añadir altura	2012	1	-	VER AÑADIR
<input type="checkbox"/>	Snape	21/04/1962	56	Mujer -	158	2015	2	-	AÑADIR

Página: 1 Filas por página: 10 1-3 de 3

Ilustración 117. Test HU2.11 Prueba 5

Prueba 6: El usuario intenta ocultar la columna *Inicio enfermedad* de la tabla

Entrada: El usuario deselecciona *Inicio enfermedad* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Inicio enfermedad*.

Salida real: Salida esperada.

Configuración

Seleccionar todas Fecha nacimiento Edad Sexo Altura Peso Inicio enfermedad Clasificación Grabaciones Atributos extra

Pacientes

<input type="checkbox"/>	Identificador ↑	Fecha Nacimiento	Edad	Sexo	Altura(cm)	Peso	Clasificación	Grabaciones	Atributos extra
<input type="checkbox"/>	Han Solo	20/10/1951	66	Hombre -	190	Añadir peso	2	-	VER
<input type="checkbox"/>	Leia	21/09/1979	38	Mujer -	Añadir altura	58	1	-	VER AÑADIR
<input type="checkbox"/>	Snape	21/04/1962	56	Mujer -	158	85	2	-	AÑADIR

Página: 1 Filas por página: 10 1-3 de 3

Ilustración 118. Test HU2.11 Prueba 6

Prueba 7: El usuario intenta ocultar la columna *Clasificación* de la tabla

Entrada: El usuario deselecciona *Clasificación* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Clasificación*.

Salida real: Salida esperada.

Configuración

Seleccionar todas Fecha nacimiento Edad Sexo Altura Peso Inicio enfermedad Clasificación Grabaciones Atributos extra

Pacientes

<input type="checkbox"/>	Identificador ↑	Fecha Nacimiento	Edad	Sexo	Altura(cm)	Peso	Inicio enfermedad	Grabaciones	Atributos extra
<input type="checkbox"/>	Han Solo	20/10/1951	66	Hombre -	190	Añadir peso	2000	-	VER
<input type="checkbox"/>	Leia	21/09/1979	38	Mujer -	Añadir altura	58	2012	-	VER AÑADIR
<input type="checkbox"/>	Snape	21/04/1962	56	Mujer -	158	85	2015	-	AÑADIR

Página: 1 Filas por página: 10 1-3 de 3

Ilustración 119. Test HU2.11 Prueba 7

Prueba 8: El usuario intenta ocultar la columna *Grabaciones* de la tabla

Entrada: El usuario deselecciona *Grabaciones* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Grabaciones*.

Salida real: Salida esperada.



The screenshot shows the 'Configuración' (Configuration) menu at the top. The 'Grabaciones' checkbox is checked, while 'Atributos extra' is unchecked. Below the menu is a table of patients with columns for 'Identificador', 'Fecha Nacimiento', 'Edad', 'Sexo', 'Altura(cm)', 'Peso', 'Inicio enfermedad', 'Clasificación', and 'Atributos extra'. The table contains three rows of patient data.

Identificador	Fecha Nacimiento	Edad	Sexo	Altura(cm)	Peso	Inicio enfermedad	Clasificación	Atributos extra
Han Solo	20/10/1951	66	Hombre	190	Añadir peso	2000	2	VER
Leia	21/09/1979	38	Mujer	Añadir altura	58	2012	1	AÑADIR
Snape	21/04/1962	56	Mujer	158	85	2015	2	AÑADIR

Ilustración 120. Test HU2.11 Prueba 8

Prueba 9: El usuario intenta ocultar la columna *Atributos extra* de la tabla

Entrada: El usuario deselecciona *Atributos extra* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Atributos extra*.

Salida real: Salida esperada.



The screenshot shows the 'Configuración' (Configuration) menu at the top. The 'Atributos extra' checkbox is checked, while 'Grabaciones' is unchecked. Below the menu is a table of patients with columns for 'Identificador', 'Fecha Nacimiento', 'Edad', 'Sexo', 'Altura(cm)', 'Peso', 'Inicio enfermedad', 'Clasificación', and 'Grabaciones'. The table contains three rows of patient data.

Identificador	Fecha Nacimiento	Edad	Sexo	Altura(cm)	Peso	Inicio enfermedad	Clasificación	Grabaciones
Han Solo	20/10/1951	66	Hombre	190	Añadir peso	2000	2	
Leia	21/09/1979	38	Mujer	Añadir altura	58	2012	1	VER
Snape	21/04/1962	56	Mujer	158	85	2015	2	

Ilustración 121. Test HU2.11 Prueba 9

HU3. Entorno grabaciones

Como cliente quiero poder ver información sobre las grabaciones, así como poder añadir nuevas grabaciones y gestionar las existentes. Además, quiero poder previsualizar el fichero CSV de la grabación y descargarlo tanto en formato CSV como en JSON.

1. Añadir una nueva grabación y comprobar que se ha añadido correctamente a la base de datos.

Prueba 1: El usuario intenta registrar una nueva grabación sin rellenar ningún campo del formulario. La aplicación no debería permitir al usuario pulsar el botón *Crear grabación*.

Entrada: Vacía.

Salida esperada: El botón *Crear grabación* bloqueado y mensaje de error en los campos pulsados.

Salida real: Salida esperada.

Prueba 2: El usuario intenta registrar una nueva grabación sin rellenar todos los campos obligatorios. La aplicación no debería permitir al usuario pulsar el botón *Crear grabación*.

Entrada: Algún campo del formulario.

Salida esperada: El botón *Crear grabación* bloqueado y mensaje de error en los campos pulsados.

Salida real: Salida esperada.

The screenshot shows a web form titled "Crear grabación" with a close button (X) in the top right corner. The form contains several input fields and buttons:

- Identificador ***: A text input field with a red underline and the error message "Campo requerido." below it.
- Lugar ***: A text input field.
- F. grabación ***: A date picker showing "23/06/2018".
- Paciente**: A dropdown menu.
- CSV no seleccionado**: A button labeled "ELEGIR FICHERO" with a red underline and the error message "Campo requerido." below it.
- Video no seleccionado**: A button labeled "ELEGIR FICHERO".
- Información extra**: A text area with a red plus sign icon.
- CREAR GRABACIÓN**: A greyed-out button at the bottom.

Ilustración 122. Test HU3.1 Prueba 1 y 2

Prueba 3: El usuario intenta registrar una nueva grabación indicando como fecha de nacimiento una fecha posterior a la actual. La aplicación no debería permitir al usuario introducir una fecha posterior.

Entrada: Fecha posterior a la actual.

Salida esperada: El sistema no permite introducir la fecha, en el calendario solo se pueden seleccionar fechas anteriores a la actual y no es posible modificar el campo manualmente.

Salida real: Salida esperada.

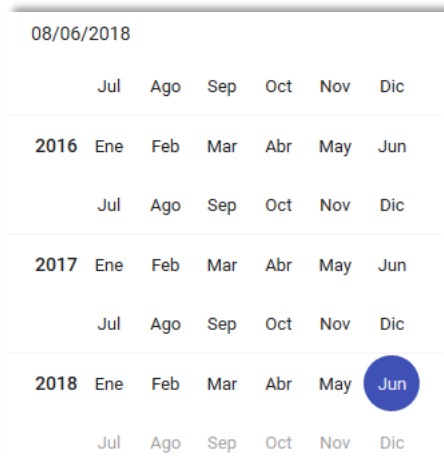


Ilustración 123. Test HU3.1 Prueba 3

Prueba 4: El usuario intenta registrar una nueva grabación introduciendo en el campo de CSV un fichero de otro formato. La aplicación no debería permitir al usuario introducir un fichero con otro formato.

Entrada: Un fichero con formato distinto a CSV.

Salida esperada: El sistema no permite al usuario seleccionar ficheros que no sean CSV.

Salida real: Salida esperada.

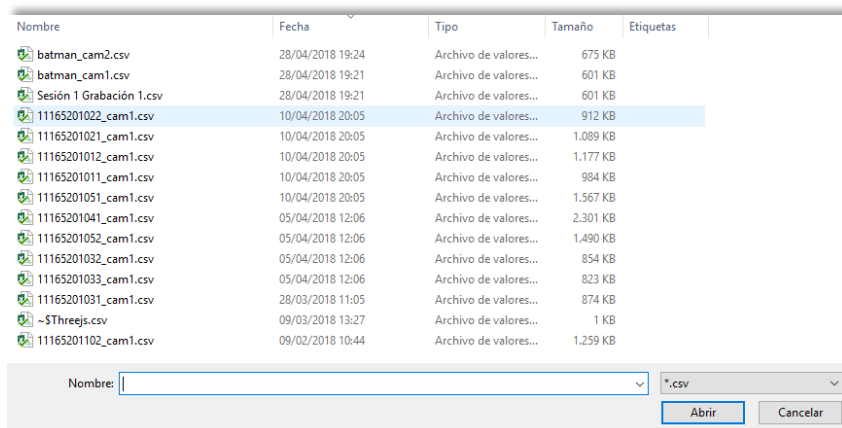


Ilustración 124. Test HU3.1 Prueba 4

Prueba 5: El usuario intenta registrar una nueva grabación introduciendo en el campo de vídeo un fichero de otro formato. La aplicación no debería permitir al usuario introducir un fichero con otro formato.

Entrada: Un fichero con formato distinto al vídeo.

Salida esperada: El sistema no permite al usuario seleccionar ficheros que no sean de vídeo.

Salida real: Salida esperada.

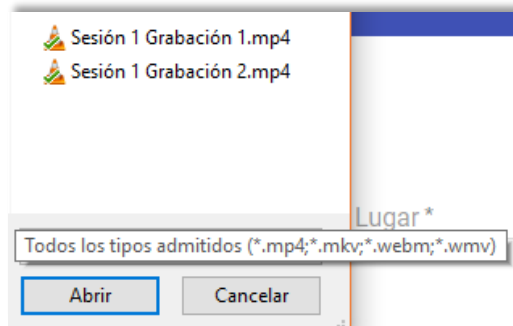


Ilustración 125. Test HU3.1 Prueba 5

Prueba 6: El usuario rellena los campos obligatorios del formulario de creación de grabación de forma correcta.

Entrada: Identificador, lugar, fecha de grabación y fichero CSV.

Salida esperada: Grabación registrada en la base de datos y mensaje indicando que la grabación se ha creado correctamente.

Salida real: Salida esperada.

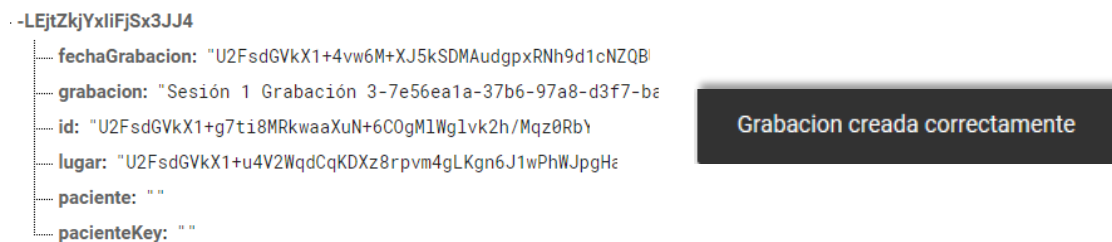


Ilustración 126. Test HU3.1 Prueba 6

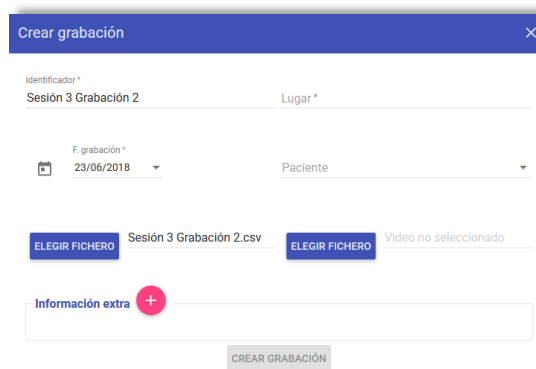
2. Las grabaciones pueden incluir tantos atributos extra como se crea conveniente.

Prueba 1: El usuario añade un nuevo campo extra pulsando el botón + de la parte superior izquierda del formulario. No rellena ningún campo. La aplicación no debería permitir al usuario pulsar el botón *Crear grabación*.

Entrada: Identificador, lugar, fecha de grabación y fichero CSV.

Salida esperada: El botón *Crear grabación* bloqueado.

Salida real: Salida esperada.



The screenshot shows a modal window titled "Crear grabación". It contains several input fields: "Identificador*" with the value "Sesión 3 Grabación 2", "Lugar*", "F. grabación*" with the value "23/06/2018", and "Paciente". Below these are two buttons labeled "ELEGIR FICHERO" next to the text "Sesión 3 Grabación 2.csv" and "Video no seleccionado". At the bottom, there is a section for "Información extra" with a red plus button. A "CREAR GRABACIÓN" button is visible at the bottom center, which is disabled (greyed out).

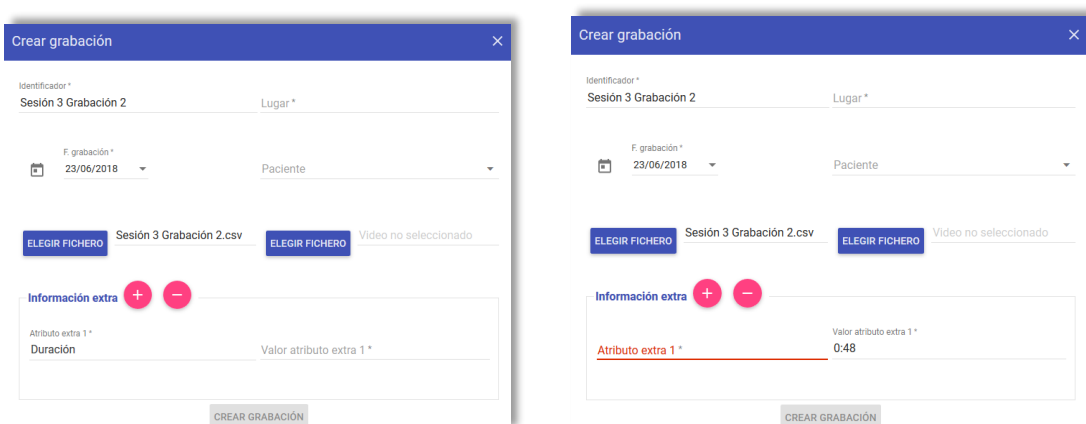
Ilustración 127. Test HU3.2 Prueba 1

Prueba 2: El usuario intenta añadir un nuevo campo extra rellenando únicamente el identificador o el valor. La aplicación no debería permitir al usuario pulsar el botón *Crear grabación*.

Entrada: Identificador, lugar, fecha de grabación y fichero CSV. Además, el Identificador o valor del nuevo atributo extra.

Salida esperada: El botón *Crear grabación* bloqueado.

Salida real: Salida esperada.



The two screenshots show the "Crear grabación" form in two states. The left screenshot is identical to the one in Illustration 127, showing the "Información extra" section with a red plus button. The right screenshot shows the "Información extra" section with both a red plus and a red minus button. Below the plus button, the "Atributo extra 1*" field is filled with "Duración" and the "Valor atributo extra 1*" field is filled with "0:48". The "CREAR GRABACIÓN" button remains disabled (greyed out) in both screenshots.

Ilustración 128. Test HU3.2 Prueba 2

Prueba 3: El usuario rellena los campos obligatorios del formulario de creación de grabación de forma correcta y completa tanto el identificador como el valor del atributo extra.

Entrada: Identificador, lugar, fecha de grabación y fichero CSV, identificador y valor del nuevo atributo extra.

Salida esperada: Paciente registrado en la base de datos con el atributo extra definido y mensaje indicando que el usuario se ha creado correctamente.

Salida real: Salida esperada.



Ilustración 129. Test HU3.2 Prueba 3

Prueba 4: El usuario elimina el último atributo extra creado pulsando sobre el botón con el símbolo - de la parte superior izquierda del formulario.

Entrada: Al menos debe haber un atributo extra creado, el usuario pulsa el botón -.

Salida esperada: El último atributo extra creado es eliminado.

Salida real: Salida esperada.

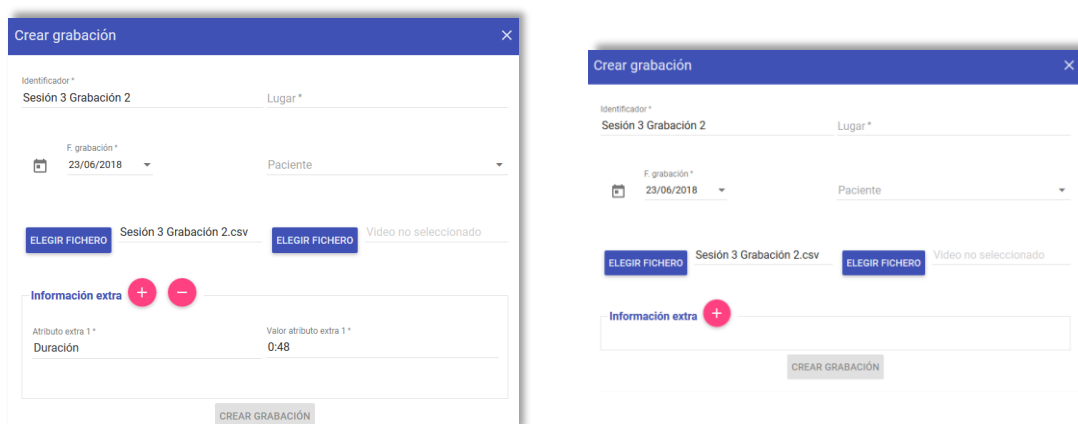


Ilustración 130. Test HU3.2 Prueba 4

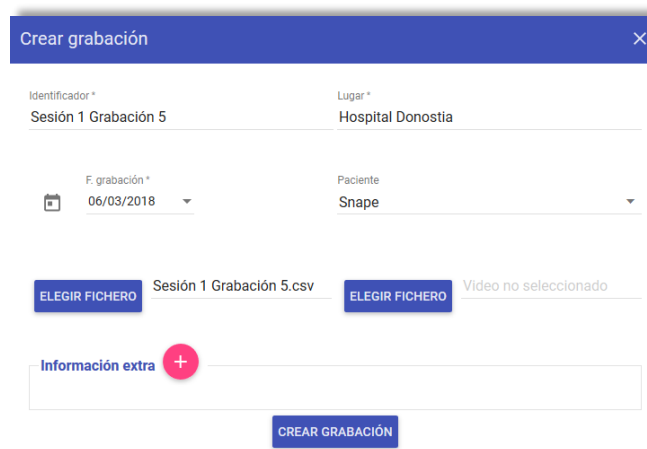
3. Asignar un paciente a una grabación al paciente a la hora de crear una grabación.

Prueba 1: El usuario registra una nueva grabación introduciendo todos los campos obligatorios de forma correcta. Existe al menos un paciente perteneciente al mismo grupo. El usuario asigna una paciente a la grabación durante el registro. La grabación queda asignada al paciente.

Entrada: El usuario introduce el identificador, lugar, fecha de grabación, fichero CSV y selección un paciente al que asignarle la grabación.

Salida esperada: La nueva grabación se crea correctamente y la grabación queda asignada al paciente seleccionado.

Salida real: Salida esperada.



Id	Fecha Grabación	Lugar	Paciente	Edad	Edad grabación	Grabación	CSV	Video	Atributos extra	Notas
Sesión 1 Grabación 5	06/03/2018	Hospital Donostia	Snape	-	56	55	Sesión 1 Grabación 5.csv	Añadir video	AÑADIR	
Sesión 1 Grabación 1	06/04/2018	Hospital Donostia	Leia	-	38	38	Sesión 1 Grabación 1.csv	Sesión 1 Grabación 1.mp4	AÑADIR	

Ilustración 131. Test HU3.3 Prueba 1

4. Comprobar que las nuevas grabaciones solamente son visibles para los usuarios que pertenecen al mismo grupo que el usuario que la ha añadido.

Prueba 1: El usuario1 perteneciente al grupo1 introduce una nueva grabación. La nueva grabación es visible para usuario1. El usuario2, perteneciente a un grupo distinto al grupo1, inicia sesión y accede a la sección *Grabaciones*. La grabación creada por usuario1 no es visible para el usuario2.

Entrada: Usuario1 introduce el identificador, lugar, fecha de grabación, fichero CSV y paciente.

Salida esperada: El usuario1 puede ver la grabación creado por él mismo, mientras que el usuario2 no tiene acceso a la nueva grabación.

Salida real: Salida esperada.

Crear grabación
✕

Identificador *
Sesión 1 Grabación 5

Lugar *
Hospital Donostia

F. grabación *
06/03/2018

Paciente
Snape

ELEGIR FICHERO Sesión 1 Grabación 5.csv

ELEGIR FICHERO Video no seleccionado

Información extra +

CREAR GRABACIÓN

Id	Fecha Grabación	Lugar	Paciente ↓	Edad	Edad grabación	Grabación	CSV	Video	Atributos extra	Notas	
<input type="checkbox"/>	Sesión 1 Grabación 5	06/03/2018	Hospital Donostia	Snape	-	56	55	📁 📄	Sesión 1 Grabación 5.csv	Añadir video	AÑADIR
<input type="checkbox"/>	Sesión 1 Grabación 1	06/04/2018	Hospital Donostia	Leta	-	38	38	📁 📄	Sesión 1 Grabación 1.csv	Sesión 1 Grabación 1.mp4	AÑADIR

Ilustración 132. Test HU3.4 Prueba 1. Grabación creada por usuario1

Id ↑	Fecha Grabación	Lugar	Paciente	Edad	Edad grabación	Grabación	CSV	Video	Atributos extra	Notas	
<input type="checkbox"/>	Prueba	06/06/2018	asdf	Link	-	37	37	📁 📄	batman_cam1.csv	Añadir video	AÑADIR
<input type="checkbox"/>	Sesión 2 Grabación 1	13/04/2018	Aulario	Hulk	-	41	41	📁 📄	batman_cam2.csv	My Video 1.mp4	AÑADIR VER
<input type="checkbox"/>	Sesión 2 Grabación 2	09/03/2018	Aulario	Mario	-	57	57	📁 📄	batman_cam2.csv	movie.mp4	AÑADIR VER
<input type="checkbox"/>	Sesión 2 Grabación 2	31/05/2018	Amurrio	Sin Asignar	-			📁 📄	batman_cam1.csv	Añadir video	AÑADIR VER

Ilustración 133. Test HU3.4 Prueba 1. Sección grabaciones del usuario2

Prueba 2: El usuario1 perteneciente al grupo1 introduce una nueva grabación. El usuario2, perteneciente al mismo grupo, inicia sesión y accede a la sección *Grabaciones*. La grabación creada por usuario1 es visible para el usuario2, además del resto de grabaciones registradas en ese grupo.

Entrada: Usuario1 introduce el identificador, lugar, fecha de grabación, fichero CSV y paciente.

Salida esperada: Tanto el usuario1 como el usuario2 tienen acceso a la nueva grabación y al resto de grabaciones de ese grupo.

Salida real: Salida esperada.

Id	Fecha Grabación	Lugar	Paciente ↓	Edad	Edad grabación	Grabación	CSV	Video	Atributos extra	Notas
<input type="checkbox"/> Sesión 1 Grabación 5	06/03/2018	Hospital Donostia	Snape	-	56	55	Sesión 1 Grabación 5.csv	Añadir video	añadir	
<input type="checkbox"/> Sesión 1 Grabación 1	06/04/2018	Hospital Donostia	Leta	-	38	38	Sesión 1 Grabación 1.csv	Sesión 1 Grabación 1.mp4	añadir	

Ilustración 134. Test HU3.4 Prueba 2. Grabación creada por usuario1

Id	Fecha Grabación	Lugar	Paciente ↓	Edad	Edad grabación	Grabación	CSV	Video	Atributos extra	Notas
<input type="checkbox"/> Sesión 1 Grabación 5	06/03/2018	Hospital Donostia	Snape	-	56	55	Sesión 1 Grabación 5.csv	Añadir video	añadir	
<input type="checkbox"/> Sesión 1 Grabación 1	06/04/2018	Hospital Donostia	Leta	-	38	38	Sesión 1 Grabación 1.csv	Sesión 1 Grabación 1.mp4	añadir	

Ilustración 135. Test HU3.4 Prueba 2. Sección grabaciones del usuario2

5. Eliminar una grabación y comprobar que se ha eliminado en la base de datos y no se muestra más en la tabla.

Prueba 1: El usuario selecciona una grabación para ser eliminada, pero cancela su borrado. Para ello selecciona la grabación pulsando en el cuadrado que se encuentra en el lateral izquierdo de la grabación y seguidamente pulsa sobre el icono con forma de papelera. El usuario cancela el borrado de la grabación pulsando en el botón *Cancelar*.

Entrada: El usuario selecciona una grabación para ser eliminada, pero cuando el sistema solicita su confirmación el usuario cancela el borrado.

Salida esperada: Las grabaciones se mantienen intactas.

Salida real: Salida esperada.

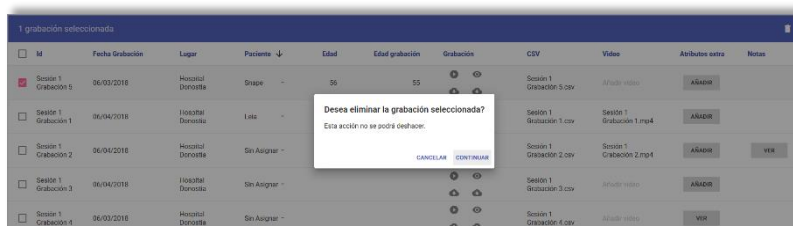


Ilustración 136. Test HU3.5 Prueba 1

Prueba 2: El usuario selecciona una grabación y la elimina. Para ello selecciona la grabación pulsando en el cuadrado que se encuentra en el lateral izquierdo de la grabación y seguidamente pulsa sobre el icono con forma de papelera. El usuario pulsa el botón *Continuar*.

Entrada: El usuario selecciona una grabación para ser eliminada, y confirma el borrado.

Salida esperada: La grabación seleccionada se borra de la base de datos.

Salida real: Salida esperada.

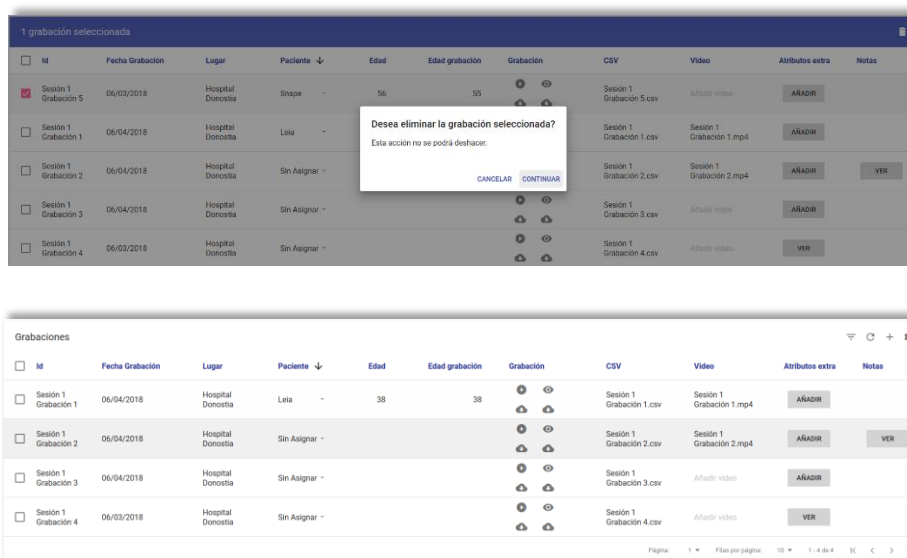


Ilustración 137. Test HU3.5 Prueba 2

6. Eliminar varias grabaciones y comprobar que se han eliminado en la base de datos y no se muestran más en la tabla.

Prueba 1: El usuario selecciona varias grabaciones y las elimina. Para ello selecciona las grabaciones pulsando en los cuadrados que se encuentra en el lateral izquierdo de cada grabación y seguidamente pulsa sobre el icono con forma de papelera. El usuario pulsa el botón *Continuar*.

Entrada: El usuario selecciona dos grabaciones para ser eliminadas, y confirma que desea eliminarlas.

Salida esperada: Las grabaciones seleccionados son borradas de la base de datos.

Salida real: Salida esperada.

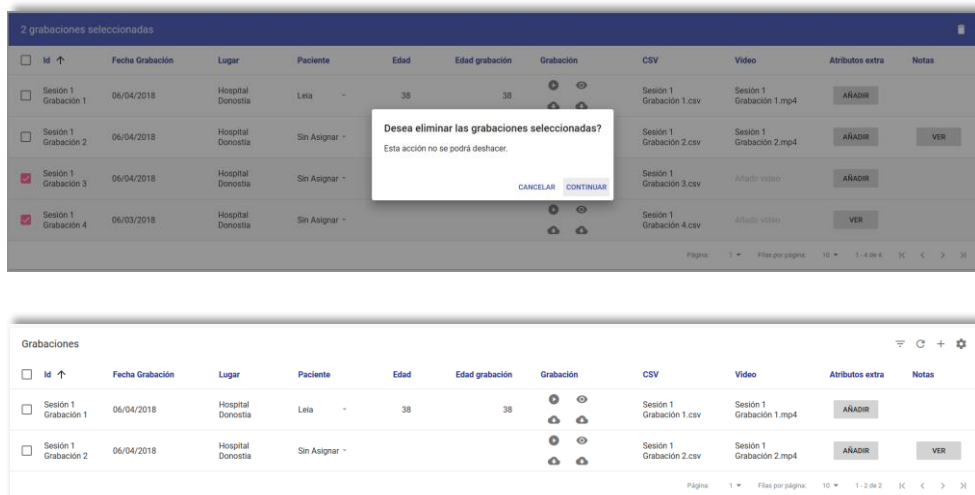


Ilustración 138. Test HU3.6 Prueba 1

7. Editar todos los campos y comprobar que los cambios se ven reflejados en la base de datos.

Prueba 1: El usuario edita el identificador de una grabación. Para ello pulsa sobre el identificador, se muestra una ventana con el identificador actual y el usuario lo edita. El usuario pulsa el botón *Guardar*.

Entrada: El nuevo identificador de la grabación.

Salida esperada: Se actualiza el identificador de la grabación.

Salida real: Salida esperada.

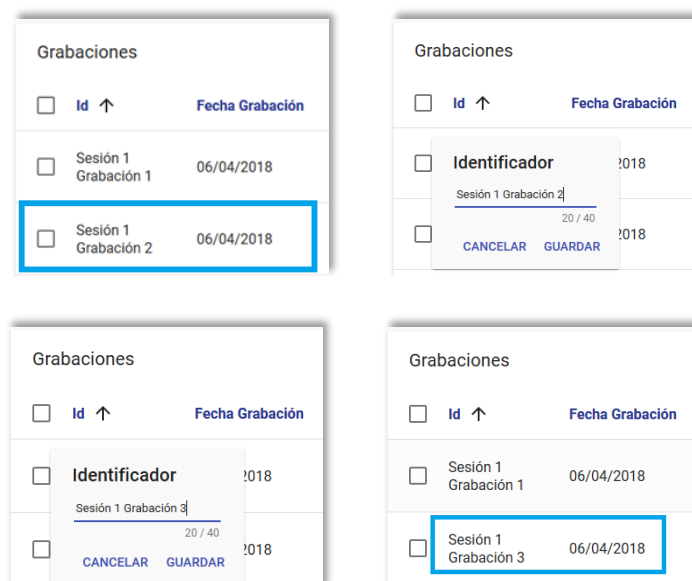


Ilustración 139. Test HU3.7 Prueba 1

Prueba 2: El usuario edita la fecha de grabación de una grabación. Para ello pulsa sobre la fecha, se muestra una ventana con la fecha de grabación y el usuario la edita. El usuario pulsa el botón *Guardar*.

Entrada: La nueva fecha de grabación de la grabación.

Salida esperada: Se actualiza la fecha de nacimiento de la grabación.

Salida real: Salida esperada.

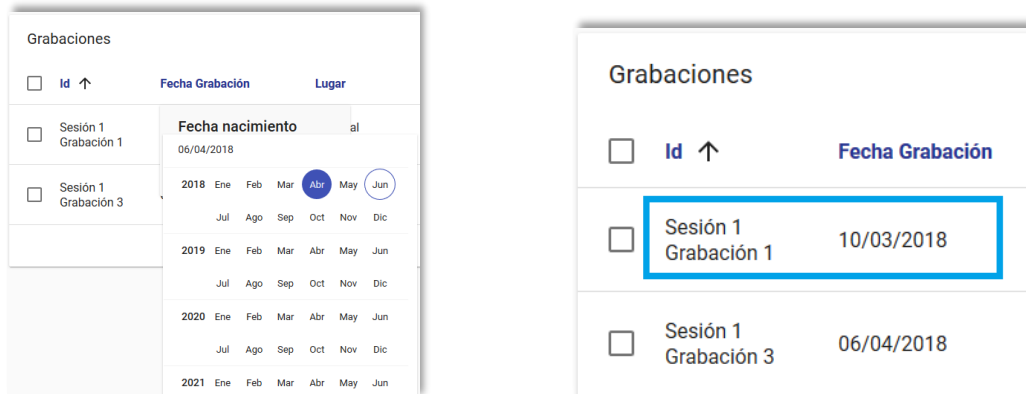


Ilustración 140. Test HU3.7 Prueba 2

Prueba 3: El usuario edita el lugar de una grabación. Para ello pulsa sobre el lugar, se muestra una ventana con el lugar y el usuario lo edita. El usuario pulsa el botón *Guardar*.

Entrada: El nuevo lugar de la grabación.

Salida esperada: Se actualiza el lugar de la grabación.

Salida real: Salida esperada.

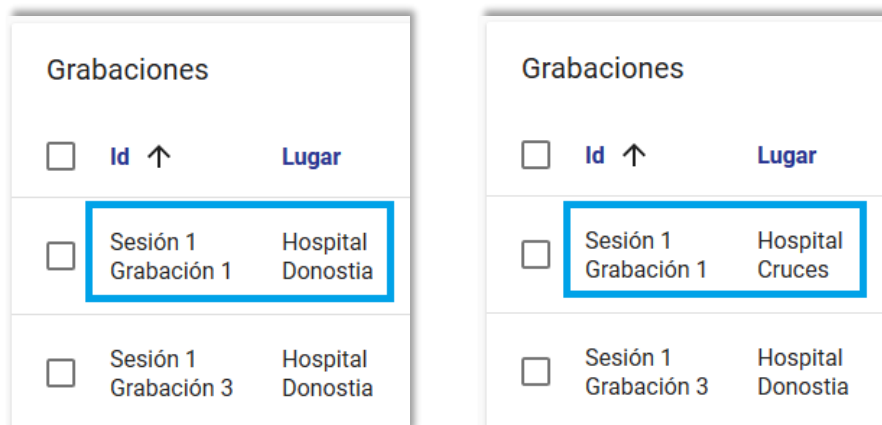


Ilustración 141. Test HU3.7 Prueba 3

Prueba 4: El usuario edita el paciente asignado a una grabación. Para ello pulsa sobre el paciente, se muestra una lista desplegable con los pacientes disponibles y el usuario elige uno. El usuario pulsa el botón *Guardar*.

Entrada: El nuevo paciente de la grabación.

Salida esperada: Se actualiza el paciente de la grabación.

Salida real: Salida esperada.

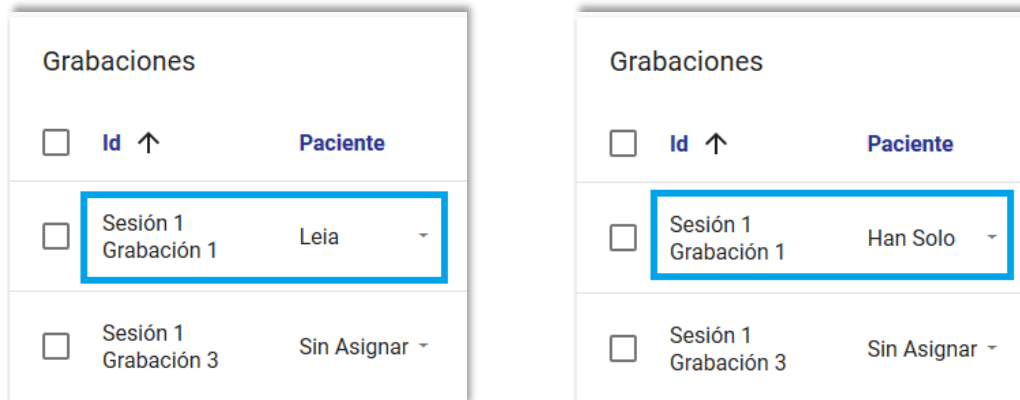


Ilustración 142. Test HU3.7 Prueba 4

Prueba 5: El usuario edita el fichero CSV de una grabación. Para ello pulsa sobre el nombre del fichero, se muestra un selector de ficheros y el usuario elige uno. El usuario pulsa el botón *Guardar*.

Entrada: El nuevo fichero CSV de la grabación.

Salida esperada: Se actualiza el fichero CSV de la grabación.

Salida real: Salida esperada.

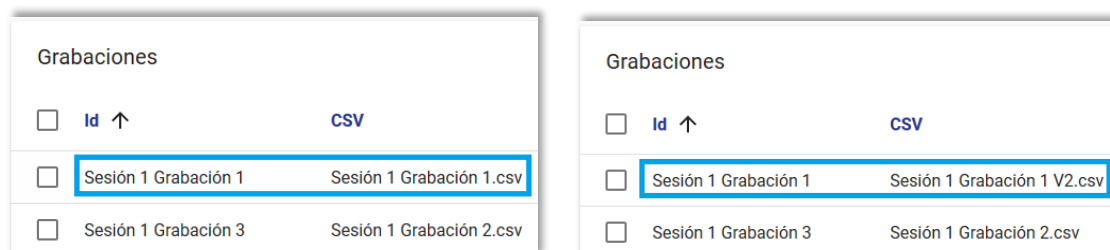


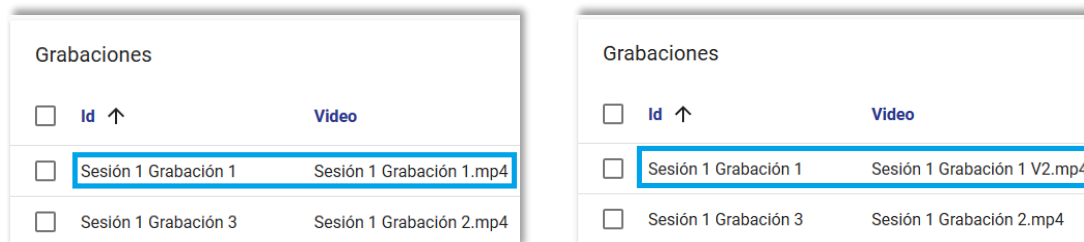
Ilustración 143. Test HU3.7 Prueba 5

Prueba 6: El usuario edita el fichero de vídeo de una grabación. Para ello pulsa sobre el nombre del fichero, se muestra un selector de ficheros y el usuario elige uno. El usuario pulsa el botón *Guardar*.

Entrada: El nuevo fichero de vídeo de la grabación.

Salida esperada: Se actualiza el fichero vídeo de la grabación.

Salida real: Salida esperada.




Grabaciones	
<input type="checkbox"/> Id ↑	Video
<input type="checkbox"/> Sesión 1 Grabación 1	Sesión 1 Grabación 1.mp4
<input type="checkbox"/> Sesión 1 Grabación 3	Sesión 1 Grabación 2.mp4

Grabaciones	
<input type="checkbox"/> Id ↑	Video
<input type="checkbox"/> Sesión 1 Grabación 1	Sesión 1 Grabación 1 V2.mp4
<input type="checkbox"/> Sesión 1 Grabación 3	Sesión 1 Grabación 2.mp4

Ilustración 144. Test HU3.7 Prueba 6

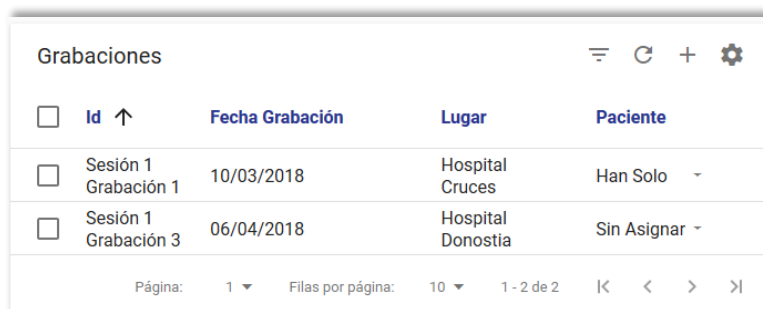
8. Filtrar las grabaciones por el identificador.

Prueba 1: El usuario filtra las grabaciones por el identificador introduciendo un identificador que no tiene ninguna grabación. Para ello pulsa sobre el icono  y se muestra una barra de búsqueda e introduce el identificador en la barra.

Entrada: Un identificador que no coincida con el de ninguna grabación.

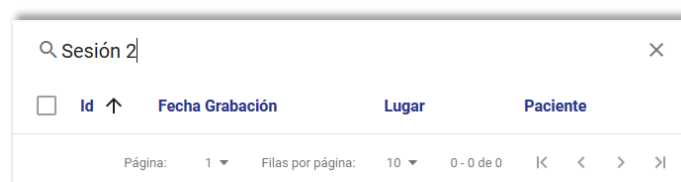
Salida esperada: La lista de grabaciones vacía.

Salida real: Salida esperada.



<input type="checkbox"/> Id ↑	Fecha Grabación	Lugar	Paciente
<input type="checkbox"/> Sesión 1 Grabación 1	10/03/2018	Hospital Cruces	Han Solo ▾
<input type="checkbox"/> Sesión 1 Grabación 3	06/04/2018	Hospital Donostia	Sin Asignar ▾

Página: 1 ▾ Filas por página: 10 ▾ 1 - 2 de 2 |< < > >|




🔍 Sesión 2 ✕

<input type="checkbox"/> Id ↑	Fecha Grabación	Lugar	Paciente
-------------------------------	-----------------	-------	----------

Página: 1 ▾ Filas por página: 10 ▾ 0 - 0 de 0 |< < > >|

Ilustración 145. Test HU3.8 Prueba 1

Prueba 2: El usuario filtra las grabaciones por el identificador introduciendo el identificador de una grabación. Para ello pulsa sobre el icono  , se muestra una barra de búsqueda e introduce el identificador en la barra.

Entrada: El identificador de una grabación.

Salida esperada: La grabación que contienen el texto introducido como identificador.

Salida real: Salida esperada.

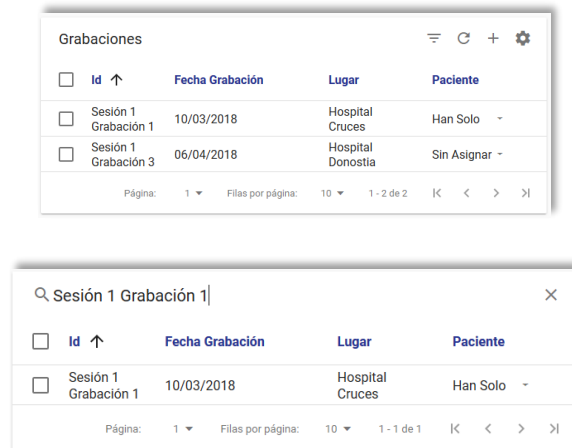



Ilustración 146. Test HU3.8 Prueba 2

Prueba 3: El usuario filtra las grabaciones por el identificador introduciendo parte del identificador de varias grabaciones. Para ello pulsa sobre el icono  , se muestra una barra de búsqueda e introduce el texto en la barra.

Entrada: Un texto que forma parte del identificador de varias grabaciones.

Salida esperada: La lista de grabaciones que contienen el texto introducido.

Salida real: Salida esperada.

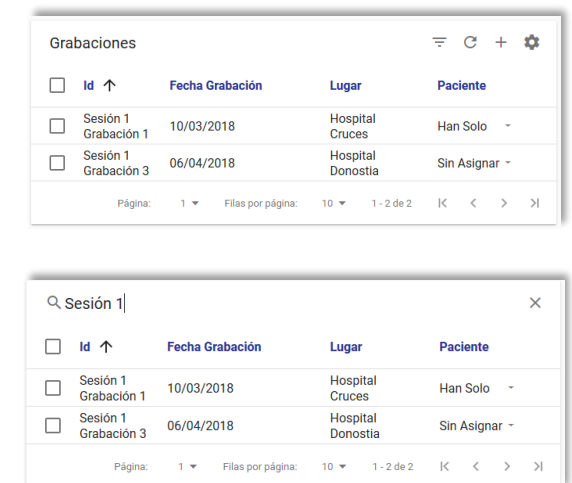


Ilustración 147. Test HU3.8 Prueba 3

9. Reordenar la tabla por columnas.

Prueba 1: El usuario intenta reordenar la tabla en base a la fecha de grabación. El sistema reordena la tabla.

Entrada: El usuario pulsa sobre la cabecera de la columna *Fecha de grabación*.

Salida esperada: La tabla reordenada en base a la fecha de grabación.

Salida real: Salida esperada.

<input type="checkbox"/>	Id ↑	Fecha Grabación
<input type="checkbox"/>	Sesión 1 Grabación 1	10/03/2018
<input type="checkbox"/>	Sesión 1 Grabación 3	06/04/2018
<input type="checkbox"/>	Sesión 2 Grabación 1	02/02/2018

<input type="checkbox"/>	Id	Fecha Grabación ↑
<input type="checkbox"/>	Sesión 2 Grabación 1	02/02/2018
<input type="checkbox"/>	Sesión 1 Grabación 1	10/03/2018
<input type="checkbox"/>	Sesión 1 Grabación 3	06/04/2018

Ilustración 148. Test HU3.9 Prueba 1

Prueba 2: El usuario intenta reordenar la tabla en base al lugar de grabación. El sistema reordena la tabla.

Entrada: El usuario pulsa sobre la cabecera de la columna *Lugar*.

Salida esperada: La tabla reordenada en base al lugar de grabación.

Salida real: Salida esperada.

<input type="checkbox"/>	Id ↑	Lugar
<input type="checkbox"/>	Sesión 1 Grabación 1	Hospital Cruces
<input type="checkbox"/>	Sesión 1 Grabación 3	Hospital Donostia
<input type="checkbox"/>	Sesión 2 Grabación 1	Hospital Donostia

<input type="checkbox"/>	Id	Lugar ↓
<input type="checkbox"/>	Sesión 1 Grabación 3	Hospital Donostia
<input type="checkbox"/>	Sesión 2 Grabación 1	Hospital Donostia
<input type="checkbox"/>	Sesión 1 Grabación 1	Hospital Cruces

Ilustración 149. Test HU3.9 Prueba 2

Prueba 3: El usuario intenta reordenar la tabla en base al paciente de la grabación. El sistema reordena la tabla.

Entrada: El usuario pulsa sobre la cabecera de la columna *Paciente*.

Salida esperada: La tabla reordenada en base al paciente de la grabación.

Salida real: Salida esperada.

The image shows two side-by-side screenshots of a table titled "Grabaciones".

The left screenshot shows the table with the "Id" column selected for sorting (indicated by an upward arrow). The rows are:

<input type="checkbox"/>	Id ↑	Paciente
<input type="checkbox"/>	Sesión 1 Grabación 1	Han Solo ▾
<input type="checkbox"/>	Sesión 1 Grabación 3	Leia ▾
<input type="checkbox"/>	Sesión 2 Grabación 1	Sin Asignar ▾

The right screenshot shows the table after sorting by "Paciente" (indicated by an upward arrow). The rows are:

<input type="checkbox"/>	Id	Paciente ↑
<input type="checkbox"/>	Sesión 2 Grabación 1	Sin Asignar ▾
<input type="checkbox"/>	Sesión 1 Grabación 1	Han Solo ▾
<input type="checkbox"/>	Sesión 1 Grabación 3	Leia ▾

Ilustración 150. Test HU3.9 Prueba 3

Prueba 4: El usuario intenta reordenar la tabla en base a la edad del paciente de la grabación. El sistema reordena la tabla.

Entrada: El usuario pulsa sobre la cabecera de la columna *Edad*.

Salida esperada: La tabla reordenada en base a la edad del paciente de la grabación.

Salida real: Salida esperada.

The image shows two side-by-side screenshots of a table titled "Grabaciones".

The left screenshot shows the table with the "Edad" column selected for sorting (indicated by an upward arrow). The rows are:

<input type="checkbox"/>	Id ↑	Edad
<input type="checkbox"/>	Sesión 1 Grabación 1	66
<input type="checkbox"/>	Sesión 1 Grabación 3	38
<input type="checkbox"/>	Sesión 2 Grabación 1	

The right screenshot shows the table after sorting by "Edad" (indicated by an upward arrow). The rows are:

<input type="checkbox"/>	Id	↑ Edad
<input type="checkbox"/>	Sesión 1 Grabación 3	38
<input type="checkbox"/>	Sesión 1 Grabación 1	66
<input type="checkbox"/>	Sesión 2 Grabación 1	

Ilustración 151. Test HU3.9 Prueba 4

Prueba 5: El usuario intenta reordenar la tabla en base a la edad del paciente en el momento de realizar la grabación. El sistema reordena la tabla.

Entrada: El usuario pulsa sobre la cabecera de la columna *Edad grabación*.

Salida esperada: La tabla reordenada en base a la edad del paciente en el momento de realizar la grabación.

Salida real: Salida esperada.

<input type="checkbox"/> Id ↑	Edad grabación
<input type="checkbox"/> Sesión 1 Grabación 1	66
<input type="checkbox"/> Sesión 1 Grabación 3	38
<input type="checkbox"/> Sesión 2 Grabación 1	

<input type="checkbox"/> Id	↑ Edad grabación
<input type="checkbox"/> Sesión 1 Grabación 3	38
<input type="checkbox"/> Sesión 1 Grabación 1	66
<input type="checkbox"/> Sesión 2 Grabación 1	

Ilustración 152. Test HU3.9 Prueba 5

Prueba 6: El usuario intenta reordenar la tabla en base al nombre del fichero CSV la grabación. El sistema reordena la tabla.

Entrada: El usuario pulsa sobre la cabecera de la columna CSV.

Salida esperada: La tabla reordenada en base al nombre del fichero CSV de la grabación.

Salida real: Salida esperada.

<input type="checkbox"/> Id ↑	CSV
<input type="checkbox"/> Sesión 1 Grabación 1	Sesión 1 Grabación 1 V2.csv
<input type="checkbox"/> Sesión 1 Grabación 2	Sesión 1 Grabación 2.csv
<input type="checkbox"/> Sesión 2 Grabación 1	Sesión 2 Grabación 1.csv

<input type="checkbox"/> Id	CSV ↓
<input type="checkbox"/> Sesión 2 Grabación 1	Sesión 2 Grabación 1.csv
<input type="checkbox"/> Sesión 1 Grabación 2	Sesión 1 Grabación 2.csv
<input type="checkbox"/> Sesión 1 Grabación 1	Sesión 1 Grabación 1 V2.csv

Ilustración 153. Test HU3.9 Prueba 6

Prueba 7: El usuario intenta reordenar la tabla en base al nombre del fichero de vídeo de la grabación. El sistema reordena la tabla.

Entrada: El usuario pulsa sobre la cabecera de la columna *Vídeo*.

Salida esperada: La tabla reordenada en base al nombre del fichero de vídeo de la grabación.

Salida real: Salida esperada.

<input type="checkbox"/>	Id ↑	Vídeo
<input type="checkbox"/>	Sesión 1 Grabación 1	Sesión 1 Grabación 1 V2.mp4
<input type="checkbox"/>	Sesión 1 Grabación 2	Sesión 1 Grabación 2.mp4
<input type="checkbox"/>	Sesión 2 Grabación 1	Añadir vídeo

<input type="checkbox"/>	Id	Vídeo ↓
<input type="checkbox"/>	Sesión 2 Grabación 1	Añadir vídeo
<input type="checkbox"/>	Sesión 1 Grabación 2	Sesión 1 Grabación 2.mp4
<input type="checkbox"/>	Sesión 1 Grabación 1	Sesión 1 Grabación 1 V2.mp4

Ilustración 154. Test HU3.9 Prueba 7

10. Ocultar columnas

Prueba 1: El usuario intenta ocultar la columna *Fecha Grabación* de la tabla.

Entrada: El usuario deselecciona *Fecha Grabación* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Fecha Grabación*.

Salida real: Salida esperada.

<input checked="" type="checkbox"/>	De-Seleccionar todas	<input checked="" type="checkbox"/>	Fecha grabación	<input checked="" type="checkbox"/>	Lugar	<input checked="" type="checkbox"/>	Paciente	<input checked="" type="checkbox"/>	Edad	<input checked="" type="checkbox"/>	Edad grabación	<input checked="" type="checkbox"/>	Grabación	<input checked="" type="checkbox"/>	CSV	<input checked="" type="checkbox"/>	Vídeo	<input checked="" type="checkbox"/>	Atributos extra	<input checked="" type="checkbox"/>	Notas
-------------------------------------	----------------------	-------------------------------------	-----------------	-------------------------------------	-------	-------------------------------------	----------	-------------------------------------	------	-------------------------------------	----------------	-------------------------------------	-----------	-------------------------------------	-----	-------------------------------------	-------	-------------------------------------	-----------------	-------------------------------------	-------

<input type="checkbox"/>	Id	Fecha Grabación	Lugar	Paciente	Edad	Edad grabación	Grabación	CSV	Vídeo ↓	Atributos extra	Notas
<input type="checkbox"/>	Sesión 2 Grabación 1	02/02/2018	Hospital Donostia	Sin Asignar -				Sesión 2 Grabación 1.csv	Añadir vídeo	AÑADIR	
<input type="checkbox"/>	Sesión 1 Grabación 2	06/04/2018	Hospital Donostia	Leia -	38	38		Sesión 1 Grabación 2.csv	Sesión 1 Grabación 2.mp4	AÑADIR	VER
<input type="checkbox"/>	Sesión 1 Grabación 1	10/03/2018	Hospital Cruces	Han Solo -	66	66		Sesión 1 Grabación 1 V2.csv	Sesión 1 Grabación 1 V2.mp4	AÑADIR	VER

<input type="checkbox"/>	De-Seleccionar todas	<input type="checkbox"/>	Fecha grabación	<input checked="" type="checkbox"/>	Lugar	<input checked="" type="checkbox"/>	Paciente	<input checked="" type="checkbox"/>	Edad	<input checked="" type="checkbox"/>	Edad grabación	<input checked="" type="checkbox"/>	Grabación	<input checked="" type="checkbox"/>	CSV	<input checked="" type="checkbox"/>	Vídeo	<input checked="" type="checkbox"/>	Atributos extra	<input checked="" type="checkbox"/>	Notas
--------------------------	----------------------	--------------------------	-----------------	-------------------------------------	-------	-------------------------------------	----------	-------------------------------------	------	-------------------------------------	----------------	-------------------------------------	-----------	-------------------------------------	-----	-------------------------------------	-------	-------------------------------------	-----------------	-------------------------------------	-------

<input type="checkbox"/>	Id	Lugar	Paciente	Edad	Edad grabación	Grabación	CSV	Vídeo ↓	Atributos extra	Notas
<input type="checkbox"/>	Sesión 2 Grabación 1	Hospital Donostia	Sin Asignar -				Sesión 2 Grabación 1.csv	Añadir vídeo	AÑADIR	
<input type="checkbox"/>	Sesión 1 Grabación 2	Hospital Donostia	Leia -	38	38		Sesión 1 Grabación 2.csv	Sesión 1 Grabación 2.mp4	AÑADIR	VER
<input type="checkbox"/>	Sesión 1 Grabación 1	Hospital Cruces	Han Solo -	66	66		Sesión 1 Grabación 1 V2.csv	Sesión 1 Grabación 1 V2.mp4	AÑADIR	VER

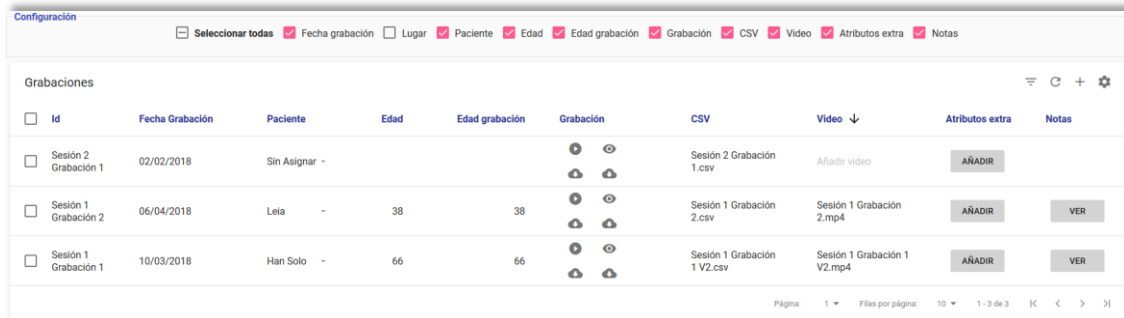
Ilustración 155. Test HU3.10 Prueba 1

Prueba 2: El usuario intenta ocultar la columna *Lugar* de la tabla.

Entrada: El usuario deselecciona *Lugar* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Lugar*.

Salida real: Salida esperada.



The screenshot shows the 'Configuración' (Configuration) menu at the top. The 'Lugar' checkbox is unchecked, indicating it is hidden. Below the menu is the 'Grabaciones' table with the following data:







<input type="checkbox"/>	Id	Fecha Grabación	Paciente	Edad	Edad grabación	Grabación	CSV	Video ↓	Atributos extra	Notas
<input type="checkbox"/>	Sesión 2 Grabación 1	02/02/2018	Sin Asignar -			 	Sesión 2 Grabación 1.csv	Añadir video	AÑADIR	
<input type="checkbox"/>	Sesión 1 Grabación 2	06/04/2018	Lola -	38	38	 	Sesión 1 Grabación 2.csv	Sesión 1 Grabación 2.mp4	AÑADIR	VER
<input type="checkbox"/>	Sesión 1 Grabación 1	10/03/2018	Han Solo -	66	66	 	Sesión 1 Grabación 1 V2.csv	Sesión 1 Grabación 1 V2.mp4	AÑADIR	VER

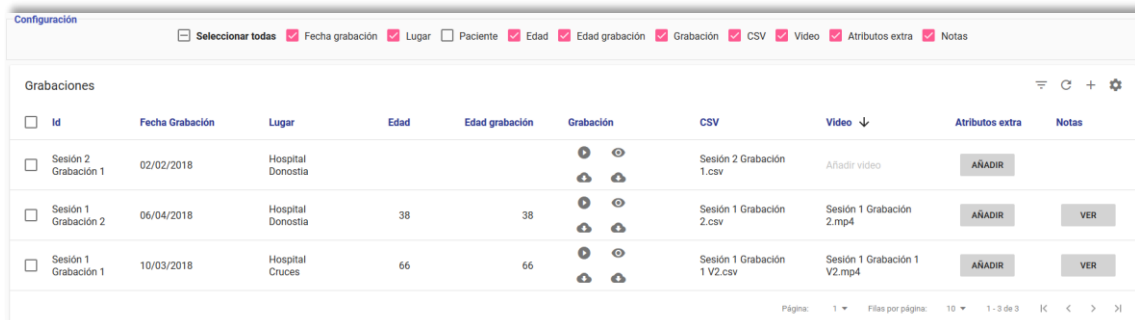
Ilustración 156. Test HU3.10 Prueba 2

Prueba 3: El usuario intenta ocultar la columna *Paciente* de la tabla.

Entrada: El usuario deselecciona *Paciente* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Paciente*.

Salida real: Salida esperada.



The screenshot shows the 'Configuración' (Configuration) menu at the top. The 'Paciente' checkbox is unchecked, indicating it is hidden. Below the menu is the 'Grabaciones' table with the following data:







<input type="checkbox"/>	Id	Fecha Grabación	Lugar	Edad	Edad grabación	Grabación	CSV	Video ↓	Atributos extra	Notas
<input type="checkbox"/>	Sesión 2 Grabación 1	02/02/2018	Hospital Donostia			 	Sesión 2 Grabación 1.csv	Añadir video	AÑADIR	
<input type="checkbox"/>	Sesión 1 Grabación 2	06/04/2018	Hospital Donostia	38	38	 	Sesión 1 Grabación 2.csv	Sesión 1 Grabación 2.mp4	AÑADIR	VER
<input type="checkbox"/>	Sesión 1 Grabación 1	10/03/2018	Hospital Cruces	66	66	 	Sesión 1 Grabación 1 V2.csv	Sesión 1 Grabación 1 V2.mp4	AÑADIR	VER

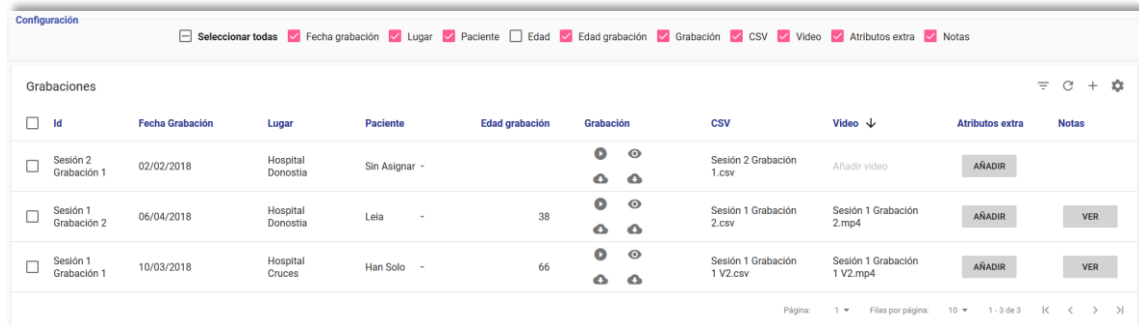
Ilustración 157. Test HU3.10 Prueba 3

Prueba 4: El usuario intenta ocultar la columna *Edad* de la tabla.

Entrada: El usuario deselecciona *Edad* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Edad*.

Salida real: Salida esperada.



The screenshot shows a configuration window titled 'Configuración'. At the top, there is a list of checkboxes for table columns: 'Seleccionar todas', 'Fecha grabación', 'Lugar', 'Paciente', 'Edad', 'Edad grabación', 'Grabación', 'CSV', 'Video', 'Atributos extra', and 'Notas'. The 'Edad' checkbox is checked. Below this is a table with the following columns: 'Id', 'Fecha Grabación', 'Lugar', 'Paciente', 'Edad grabación', 'Grabación', 'CSV', 'Video', 'Atributos extra', and 'Notas'. The table contains three rows of recording sessions.

<input type="checkbox"/>	Id	Fecha Grabación	Lugar	Paciente	Edad grabación	Grabación	CSV	Video	Atributos extra	Notas
<input type="checkbox"/>	Sesión 2 Grabación 1	02/02/2018	Hospital Donostia	Sin Asignar -			Sesión 2 Grabación 1.csv	Añadir video	AÑADIR	
<input type="checkbox"/>	Sesión 1 Grabación 2	06/04/2018	Hospital Donostia	Leia -	38		Sesión 1 Grabación 2.csv	Sesión 1 Grabación 2.mp4	AÑADIR	VER
<input type="checkbox"/>	Sesión 1 Grabación 1	10/03/2018	Hospital Cruces	Han Solo -	66		Sesión 1 Grabación 1 V2.csv	Sesión 1 Grabación 1 V2.mp4	AÑADIR	VER

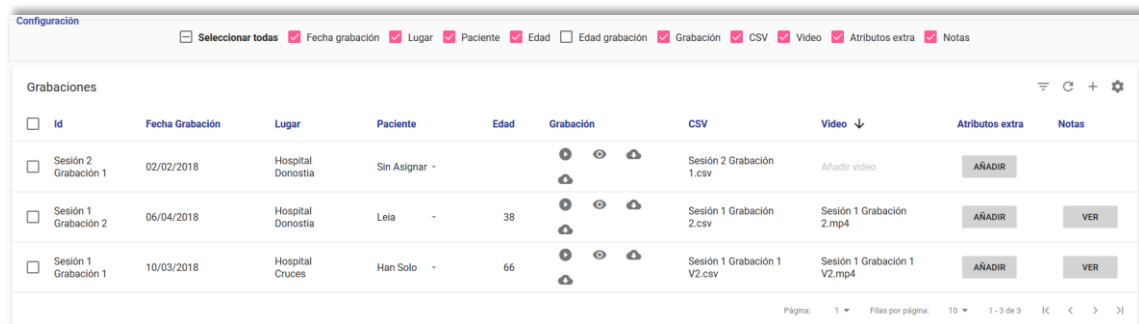
Ilustración 158. Test HU3.10 Prueba 4

Prueba 5: El usuario intenta ocultar la columna *Edad Grabación* de la tabla.

Entrada: El usuario deselecciona *Edad Grabación* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Edad Grabación*.

Salida real: Salida esperada.



The screenshot shows the same configuration window as in the previous image, but now the 'Edad grabación' checkbox is unchecked. The table below shows that the 'Edad grabación' column is no longer visible in the table headers.

<input type="checkbox"/>	Id	Fecha Grabación	Lugar	Paciente	Edad	Grabación	CSV	Video	Atributos extra	Notas
<input type="checkbox"/>	Sesión 2 Grabación 1	02/02/2018	Hospital Donostia	Sin Asignar -			Sesión 2 Grabación 1.csv	Añadir video	AÑADIR	
<input type="checkbox"/>	Sesión 1 Grabación 2	06/04/2018	Hospital Donostia	Leia -	38		Sesión 1 Grabación 2.csv	Sesión 1 Grabación 2.mp4	AÑADIR	VER
<input type="checkbox"/>	Sesión 1 Grabación 1	10/03/2018	Hospital Cruces	Han Solo -	66		Sesión 1 Grabación 1 V2.csv	Sesión 1 Grabación 1 V2.mp4	AÑADIR	VER

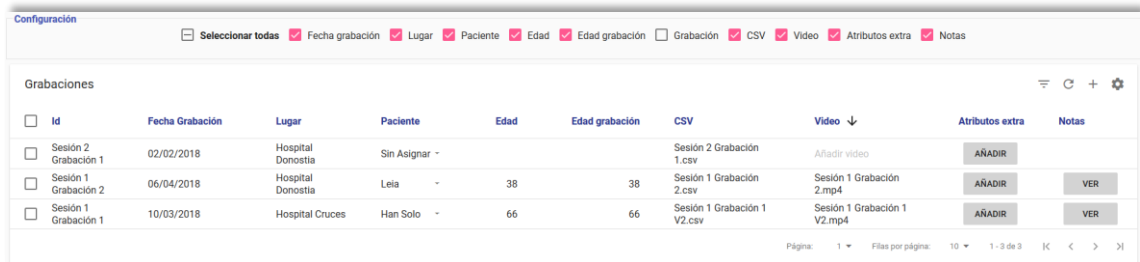
Ilustración 159. Test HU3.10 Prueba 5

Prueba 6: El usuario intenta ocultar la columna *Grabación* de la tabla.

Entrada: El usuario deselecciona *Grabación* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Grabación*.

Salida real: Salida esperada.



The screenshot shows the 'Configuración' (Configuration) interface. At the top, there is a navigation bar with a menu icon and a list of checkboxes for various columns: 'Seleccionar todas', 'Fecha grabación', 'Lugar', 'Paciente', 'Edad', 'Edad grabación', 'Grabación', 'CSV', 'Video', 'Atributos extra', and 'Notas'. The 'Grabación' checkbox is unchecked. Below this is a table titled 'Grabaciones'. The table has columns: 'Id', 'Fecha Grabación', 'Lugar', 'Paciente', 'Edad', 'Edad grabación', 'Video', 'Atributos extra', and 'Notas'. The 'Grabación' column is hidden. The table contains three rows of recording data.

<input type="checkbox"/>	Id	Fecha Grabación	Lugar	Paciente	Edad	Edad grabación	Video	Atributos extra	Notas
<input type="checkbox"/>	Sesión 2 Grabación 1	02/02/2018	Hospital Donostia	Sin Asignar -			Sesión 2 Grabación 1.csv	Añadir video	añadir
<input type="checkbox"/>	Sesión 1 Grabación 2	06/04/2018	Hospital Donostia	Leia -	38	38	Sesión 1 Grabación 2.csv	Sesión 1 Grabación 2.mp4	añadir VER
<input type="checkbox"/>	Sesión 1 Grabación 1	10/03/2018	Hospital Cruces	Han Solo -	66	66	Sesión 1 Grabación 1 V2.csv	Sesión 1 Grabación 1 V2.mp4	añadir VER

Page: 1 | Rows per page: 10 | 1-3 de 3

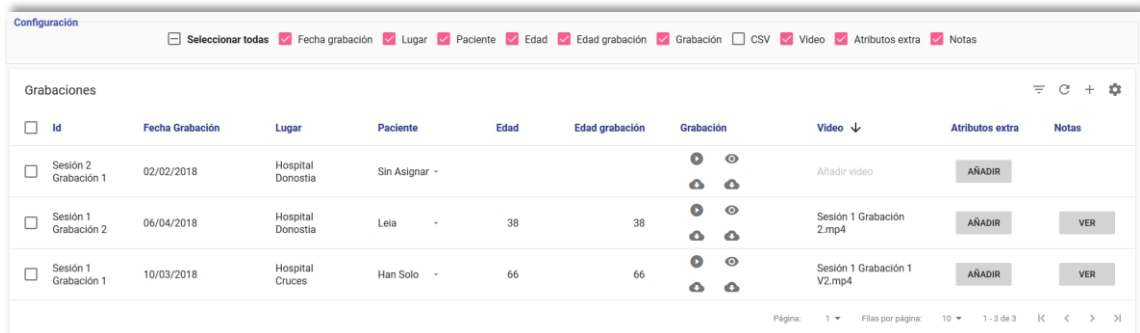
Ilustración 160. Test HU3.10 Prueba 6

Prueba 7: El usuario intenta ocultar la columna CSV de la tabla.

Entrada: El usuario deselecciona CSV del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna CSV.

Salida real: Salida esperada.



The screenshot shows the 'Configuración' (Configuration) interface. At the top, there is a navigation bar with a menu icon and a list of checkboxes for various columns: 'Seleccionar todas', 'Fecha grabación', 'Lugar', 'Paciente', 'Edad', 'Edad grabación', 'Grabación', 'CSV', 'Video', 'Atributos extra', and 'Notas'. The 'CSV' checkbox is unchecked. Below this is a table titled 'Grabaciones'. The table has columns: 'Id', 'Fecha Grabación', 'Lugar', 'Paciente', 'Edad', 'Edad grabación', 'Grabación', 'Video', 'Atributos extra', and 'Notas'. The 'CSV' column is hidden. The table contains three rows of recording data.

<input type="checkbox"/>	Id	Fecha Grabación	Lugar	Paciente	Edad	Edad grabación	Grabación	Video	Atributos extra	Notas
<input type="checkbox"/>	Sesión 2 Grabación 1	02/02/2018	Hospital Donostia	Sin Asignar -			👁️ 👁️	Añadir video	añadir	
<input type="checkbox"/>	Sesión 1 Grabación 2	06/04/2018	Hospital Donostia	Leia -	38	38	👁️ 👁️	Sesión 1 Grabación 2.mp4	añadir	VER
<input type="checkbox"/>	Sesión 1 Grabación 1	10/03/2018	Hospital Cruces	Han Solo -	66	66	👁️ 👁️	Sesión 1 Grabación 1 V2.mp4	añadir	VER

Page: 1 | Rows per page: 10 | 1-3 de 3

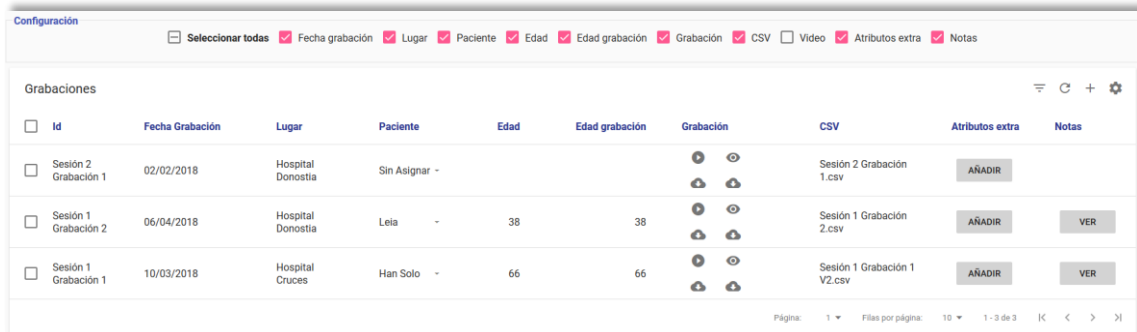
Ilustración 161. Test HU3.10 Prueba 7

Prueba 8: El usuario intenta ocultar la columna *Vídeo* de la tabla.

Entrada: El usuario deselecciona *Vídeo* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Vídeo*.

Salida real: Salida esperada.



The screenshot shows the 'Configuración' (Configuration) menu at the top. A row of checkboxes allows toggling various columns: 'Seleccionar todas', 'Fecha grabación', 'Lugar', 'Paciente', 'Edad', 'Edad grabación', 'Grabación', 'CSV', 'Video', 'Atributos extra', and 'Notas'. The 'Video' checkbox is currently unchecked. Below this is the 'Grabaciones' table with the following columns: Id, Fecha Grabación, Lugar, Paciente, Edad, Edad grabación, Grabación, CSV, Atributos extra, and Notas. The 'Video' column is hidden. The table contains three rows of recording sessions.







<input type="checkbox"/>	Id	Fecha Grabación	Lugar	Paciente	Edad	Edad grabación	Grabación	CSV	Atributos extra	Notas
<input type="checkbox"/>	Sesión 2 Grabación 1	02/02/2018	Hospital Donostia	Sin Asignar -			 	Sesión 2 Grabación 1.csv	ANADIR	
<input type="checkbox"/>	Sesión 1 Grabación 2	06/04/2018	Hospital Donostia	Leia -	38	38	 	Sesión 1 Grabación 2.csv	ANADIR	VER
<input type="checkbox"/>	Sesión 1 Grabación 1	10/03/2018	Hospital Cruces	Han Solo -	66	66	 	Sesión 1 Grabación 1 V2.csv	ANADIR	VER

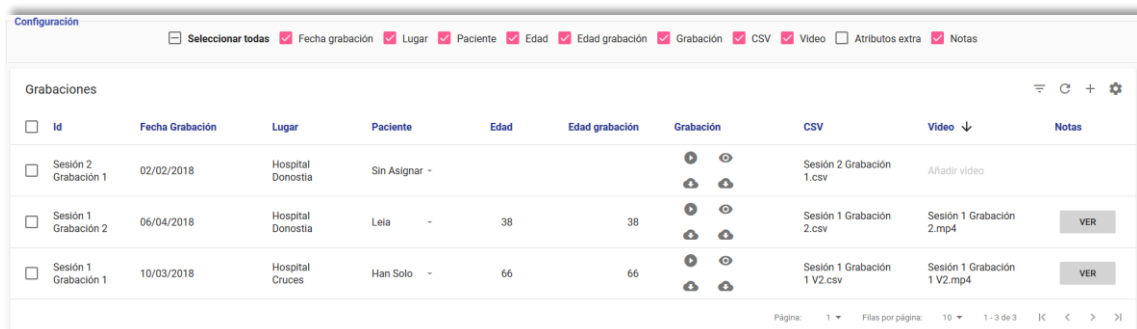
Ilustración 162. Test HU3.10 Prueba 8

Prueba 9: El usuario intenta ocultar la columna *Atributos extra* de la tabla.

Entrada: El usuario deselecciona *Atributos extra* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Atributos extra*.

Salida real: Salida esperada.



The screenshot shows the 'Configuración' (Configuration) menu at the top. A row of checkboxes allows toggling various columns: 'Seleccionar todas', 'Fecha grabación', 'Lugar', 'Paciente', 'Edad', 'Edad grabación', 'Grabación', 'CSV', 'Video', 'Atributos extra', and 'Notas'. The 'Atributos extra' checkbox is currently unchecked. Below this is the 'Grabaciones' table with the following columns: Id, Fecha Grabación, Lugar, Paciente, Edad, Edad grabación, Grabación, CSV, Video, and Notas. The 'Atributos extra' column is hidden. The table contains three rows of recording sessions.







<input type="checkbox"/>	Id	Fecha Grabación	Lugar	Paciente	Edad	Edad grabación	Grabación	CSV	Video	Notas
<input type="checkbox"/>	Sesión 2 Grabación 1	02/02/2018	Hospital Donostia	Sin Asignar -			 	Sesión 2 Grabación 1.csv	Añadir vídeo	
<input type="checkbox"/>	Sesión 1 Grabación 2	06/04/2018	Hospital Donostia	Leia -	38	38	 	Sesión 1 Grabación 2.csv	Sesión 1 Grabación 2.mp4	VER
<input type="checkbox"/>	Sesión 1 Grabación 1	10/03/2018	Hospital Cruces	Han Solo -	66	66	 	Sesión 1 Grabación 1 V2.csv	Sesión 1 Grabación 1 V2.mp4	VER

Ilustración 163. Test HU3.10 Prueba 9

Prueba 10: El usuario intenta ocultar la columna *Notas* de la tabla.

Entrada: El usuario deselecciona *Notas* del menú Configuración de la parte superior de la pantalla.

Salida esperada: Se oculta la columna *Notas*.

Salida real: Salida esperada.

Configuración										
<input type="checkbox"/> Seleccionar todas <input checked="" type="checkbox"/> Fecha grabación <input checked="" type="checkbox"/> Lugar <input checked="" type="checkbox"/> Paciente <input checked="" type="checkbox"/> Edad <input checked="" type="checkbox"/> Edad grabación <input checked="" type="checkbox"/> Grabación <input checked="" type="checkbox"/> CSV <input checked="" type="checkbox"/> Video <input checked="" type="checkbox"/> Atributos extra <input type="checkbox"/> Notas										
Grabaciones										
<input type="checkbox"/> Id	Fecha Grabación	Lugar	Paciente	Edad	Edad grabación	Grabación	CSV	Video ↓	Atributos extra	
<input type="checkbox"/> Sesión 2 Grabación 1	02/02/2018	Hospital Donostia	Sin Asignar -				Sesión 2 Grabación 1.csv	Añadir video	AÑADIR	
<input type="checkbox"/> Sesión 1 Grabación 2	06/04/2018	Hospital Donostia	Leia -	38	38		Sesión 1 Grabación 2.csv	Sesión 1 Grabación 2.mp4	AÑADIR	
<input type="checkbox"/> Sesión 1 Grabación 1	10/03/2018	Hospital Cruces	Han Solo -	66	66		Sesión 1 Grabación 1 V2.csv	Sesión 1 Grabación 1 V2.mp4	AÑADIR	

Ilustración 164. Test HU3.10 Prueba 10

11. Previsualizar el fichero csv.

Prueba 1: El usuario intenta previsualizar el CSV de una grabación.

Entrada: El usuario pulsa sobre el icono de la columna *Grabación*.

Salida esperada: Se previsualizan las primeras 200 líneas del fichero CSV en una ventana emergente.


Salida real: Salida esperada.

Previsualizar CSV (primeras 200 líneas)																
00:00:02.7330575	0	0	0	0	0	0,003986061	0,2590617	-0,05879498	0	0,01034391	0,5148278	-0,1280084	0	0,04025632	0,6540575	-0,16
00:00:02.7669870	0	0	0	0	0	0,006572962	0,2557962	-0,05996513	0	0,01596075	0,5079291	-0,1311679	0	0,03927171	0,6491221	-0,16
00:00:02.8331128	0	0	0	0	0	0,008556247	0,2550792	-0,06052113	0	0,01942039	0,5068847	-0,1313848	0	0,03580016	0,6482009	-0,15
00:00:02.8669748	0	0	0	0	0	0,009211361	0,2577412	-0,05303717	0	0,02053529	0,5115408	-0,1176229	0	0,03872687	0,6547628	-0,15
00:00:02.9000080	0	0	0	0	0	0,008554757	0,2606372	-0,04383659	0	0,01925808	0,5169447	-0,09814978	0	0,03949505	0,6574172	-0,14
00:00:02.9330424	0	0	0	0	0	0,009468198	0,2717975	-0,04722881	0	0,02149302	0,5413916	-0,1105862	0	0,04312199	0,6771777	-0,13
00:00:02.9669989	0	0	0	0	0	0,01470584	0,2758149	-0,04422426	0	0,03067738	0,5471836	-0,09302616	0	0,05094332	0,6884049	-0,13
00:00:03.0000175	0	0	0	0	0	0,01393664	0,280026	-0,04073	0	0,02996242	0,5561286	-0,09389448	0	0,04999936	0,6941211	-0,12
00:00:03.0330396	0	0	0	0	0	0,01968735	0,2627768	-0,04731274	0	0,04165733	0,5218394	-0,1052604	0	0,05784333	0,6647643	-0,13
00:00:03.0669734	0	0	0	0	0	0,01337069	0,2711679	-0,03667545	0	0,03132623	0,5355382	-0,08786249	0	0,042018	0,6770582	-0,12
00:00:03.1000109	0	0	0	0	0	0,01251054	0,2736791	-0,04020452	0	0,02738583	0,5426177	-0,08956003	0	0,03322464	0,685883	-0,11
00:00:03.1330475	0	0	0	0	0	0,01523411	0,2759488	-0,03692675	0	0,03250265	0,5475471	-0,0858233	0	0,03245127	0,6897444	-0,11
00:00:03.1669740	0	0	0	0	0	0,008019984	0,2791451	-0,03538179	0	0,01861978	0,5542871	-0,07989073	0	0,02428812	0,7011126	-0,10
00:00:03.1999927	0	0	0	0	0	0,008270144	0,2791837	-0,03805757	0	0,01903516	0,5542538	-0,08793235	0	0,02242839	0,6985454	-0,11
00:00:03.2331178	0	0	0	0	0	0,009511888	0,2799407	-0,04208088	0	0,02167374	0,5559496	-0,09443951	0	0,0195232	0,6975855	-0,12
00:00:03.2999945	0	0	0	0	0	0,01154989	0,2832565	-0,04047036	0	0,02565873	0,5622752	-0,09182835	0	0,01052868	0,7024144	-0,12
00:00:03.3330401	0	0	0	0	0	0,01201385	0,2838714	-0,03656006	0	0,02656358	0,5634359	-0,08432531	0	0,01331329	0,7031119	-0,11
00:00:03.3670128	0	0	0	0	0	0,008796871	0,2855452	-0,03734994	0	0,02028328	0,5667589	-0,08565426	0	0,01474881	0,7086622	-0,11
00:00:03.4330384	0	0	0	0	0	0,004307253	0,2863187	-0,03451705	0	0,01123601	0,5681854	-0,08000469	0	0,001154184	0,7090763	-0,09

Ilustración 165. Test HU3.11 Prueba 1

12. Descargar el fichero csv.

Prueba 1: El usuario intenta descargar el CSV de una grabación.

Entrada: El usuario pulsa sobre el icono  correspondiente al fichero CSV de la columna *Grabación*.

Salida esperada: Se descarga el fichero CSV.

Salida real: Salida esperada.

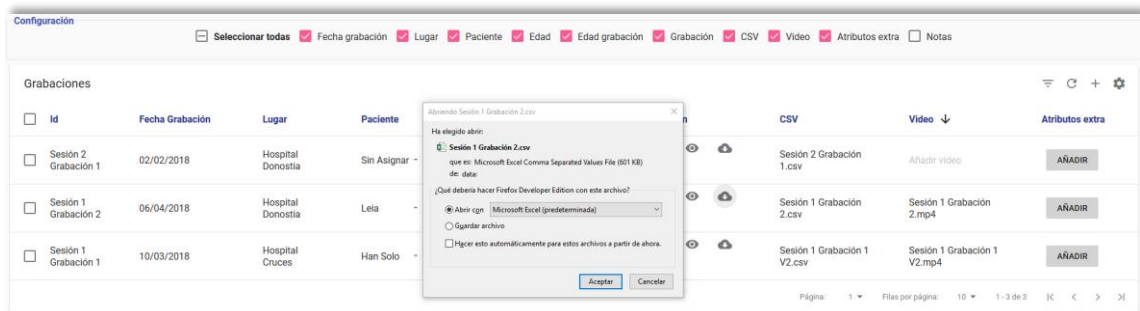


Ilustración 166. Test HU3.12 Prueba 1

13. Descargar el fichero JSON.

Prueba 1: El usuario intenta descargar el JSON de una grabación.

Entrada: El usuario pulsa sobre el icono  correspondiente al JSON de la columna *Grabación*.

Salida esperada: Se descarga el fichero JSON.

Salida real: Salida esperada.

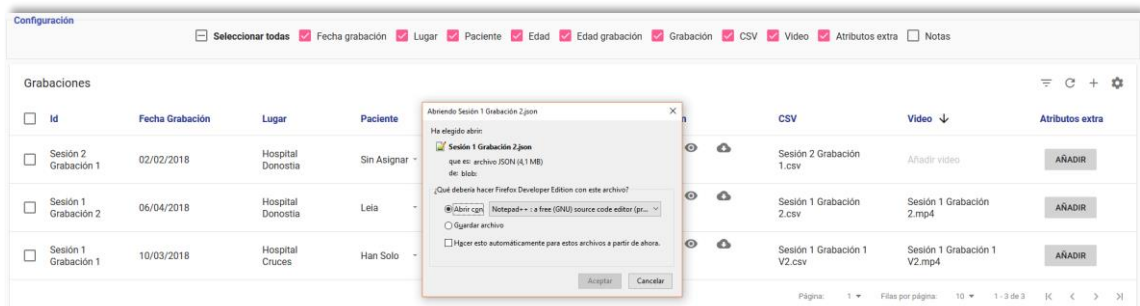



Ilustración 167. Test HU3.12 Prueba 1

Prueba 2: El usuario intenta descargar el JSON de una grabación y comprueba que en el fichero se incluye información sobre las notas añadidas.

Entrada: El usuario pulsa sobre el icono  correspondiente al JSON de la columna *Grabación*.

Salida esperada: Se descarga el fichero JSON y contiene las notas asociadas a la grabación.

Salida real: Salida esperada.

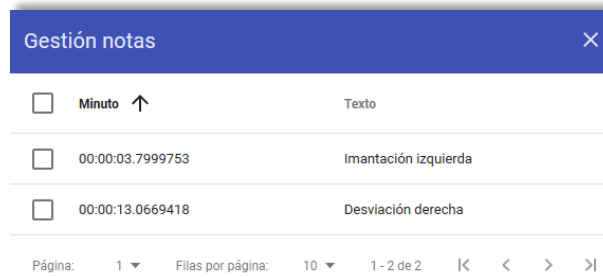


Ilustración 168. Test HU3.13 Prueba 2 Notas

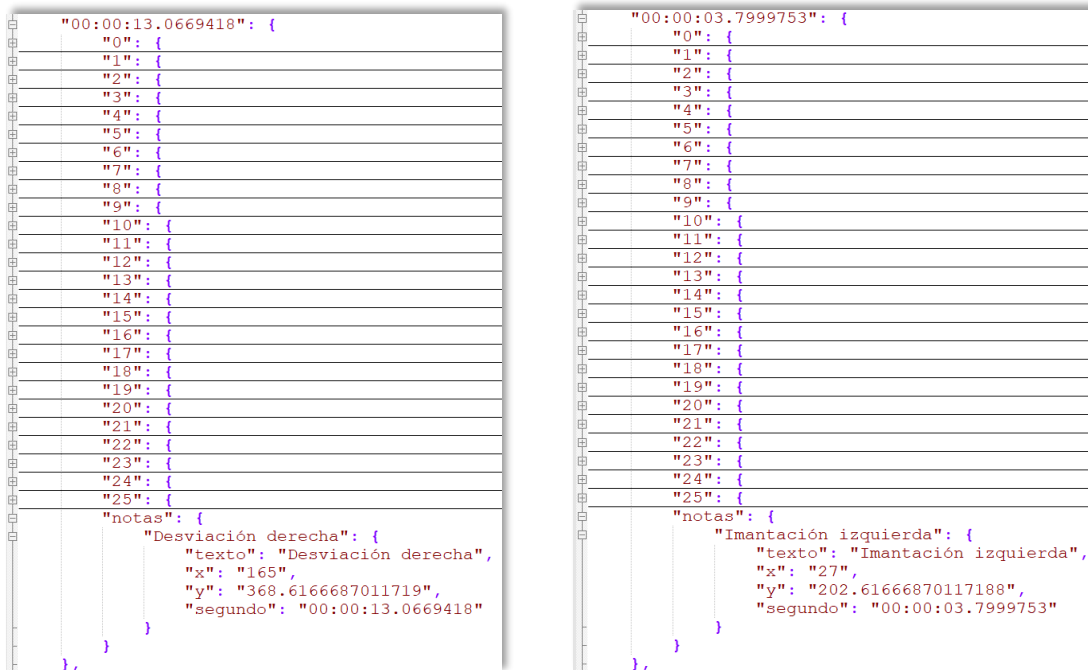



Ilustración 169. Test HU3.13 Prueba 2. JSON

HU4. Visualización gráfica de los esqueletos

Como cliente quiero visualizar gráficamente los esqueletos a partir de un fichero para poder observar los movimientos del paciente junto con información relativa a los mismos.










1. Comprobar que el cliente puede visualizar el esqueleto para un fichero concreto.

Prueba 1: El usuario intenta visualizar el esqueleto de una grabación.

Entrada: El usuario pulsa sobre el icono  de la columna *Grabación*.

Salida esperada: Se visualiza el esqueleto del fichero CSV asociado a la grabación.

Salida real: Salida esperada.

<input type="checkbox"/>	Id	Fecha Grabación	Lugar	Paciente	Edad	Edad grabación	Grabación	CSV	Video ↓	Atributos extra
<input type="checkbox"/>	Sesión 2 Grabación 1	02/02/2018	Hospital Donostia	Sin Asignar -			  	Sesión 2 Grabación 1.csv	Añadir video	<input type="button" value="AÑADIR"/>
<input type="checkbox"/>	Sesión 1 Grabación 2	06/04/2018	Hospital Donostia	Leia -	38	38	  	Sesión 1 Grabación 2.csv	Sesión 1 Grabación 2.mp4	<input type="button" value="AÑADIR"/>
<input type="checkbox"/>	Sesión 1 Grabación 1	10/03/2018	Hospital Cruces	Han Solo -	66	66	  	Sesión 1 Grabación 1 V2.csv	Sesión 1 Grabación 1 V2.mp4	<input type="button" value="AÑADIR"/>

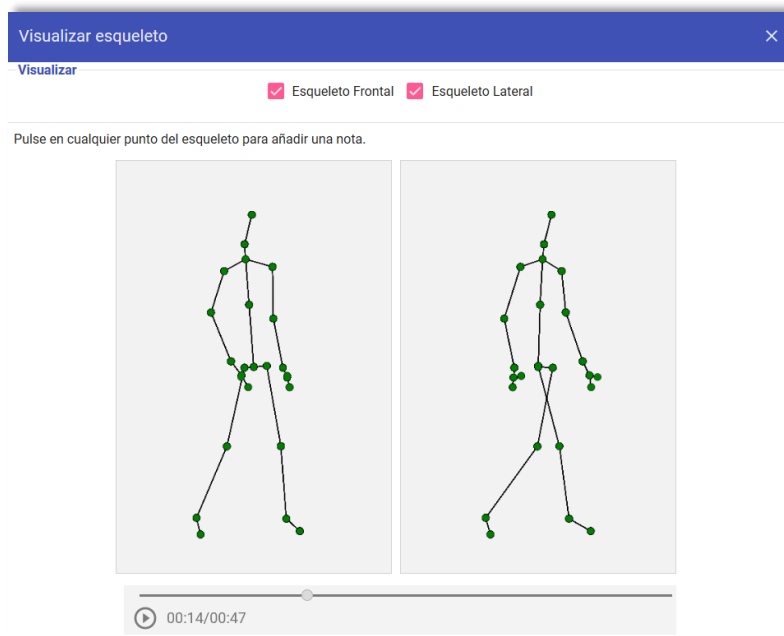


Ilustración 170. Test HU4.1 Prueba 1

2. Comprobar que si la articulación es inferida se muestra de un color distinto.

Prueba 1: El usuario intenta visualizar el esqueleto de una grabación.

Entrada: El usuario pulsa sobre el icono  de la columna *Grabación*.

Salida esperada: Se visualiza el esqueleto del fichero CSV asociado a la grabación, pintando en verde las articulaciones no inferidas y en rojo las inferidas.

Salida real: Salida esperada.



Ilustración 171. Test HU4.2 Prueba 1

3. Se permite tomar anotaciones durante la reproducción pulsando en cualquier punto de la zona de visualización del esqueleto.

Prueba 1: El usuario intenta tomar una nota mientras visualiza el esqueleto de una grabación.

Entrada: El usuario pulsa sobre cualquier punto del esqueleto.

Salida esperada: Se muestra una ventana emergente en la que añadir la anotación.

Salida real: Salida esperada.

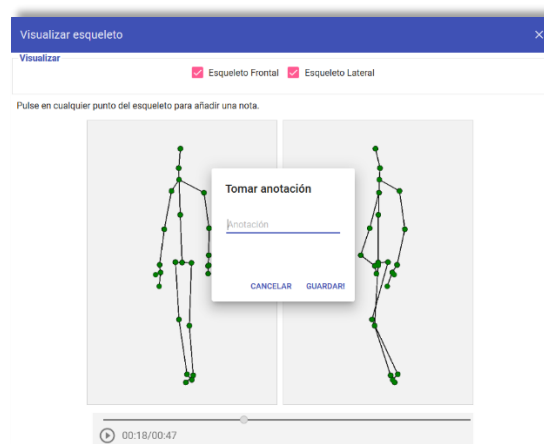


Ilustración 172. Test HU4.3 Prueba 1

4. Las anotaciones tomadas se visualizan durante la reproducción del esqueleto.

Prueba 1: El usuario intenta visualizar una nota mientras visualiza el esqueleto de una grabación.

Entrada: El usuario visualiza la grabación del esqueleto.

Salida esperada: Se muestran las notas durante la reproducción del esqueleto.

Salida real: Salida esperada.

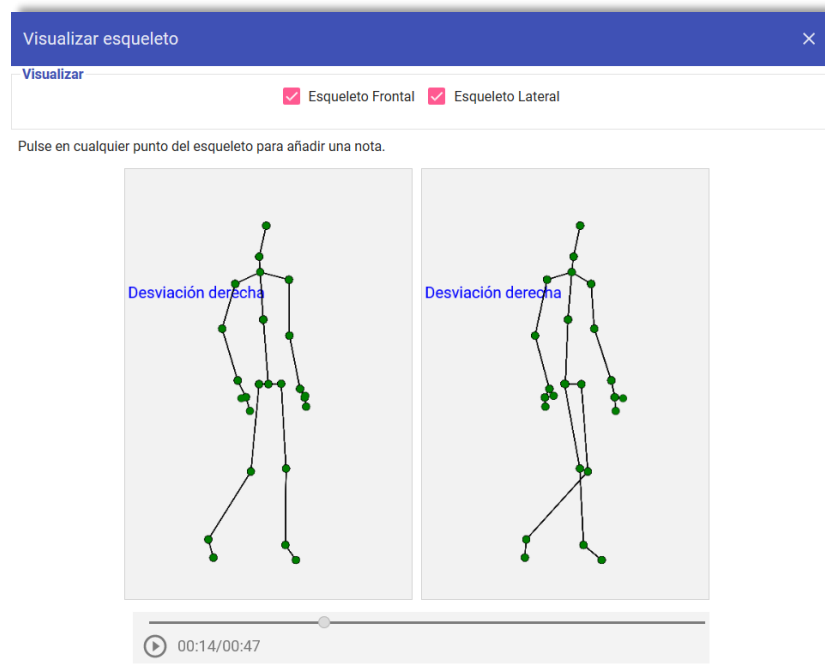


Ilustración 173. Test HU4.4 Prueba 1

HU5. Visualización de vídeos

Como cliente quiero visualizar los vídeos de las grabaciones desde la aplicación web.

1. Comprobar que el vídeo se ve correctamente.

Prueba 1: El usuario intenta visualizar el vídeo de una grabación.

Entrada: El usuario pulsa sobre el icono de  la columna *Grabación*.

Salida esperada: Se visualiza el fichero de vídeo de la grabación, en caso de que la grabación tenga un vídeo asociado.

Salida real: Salida esperada.

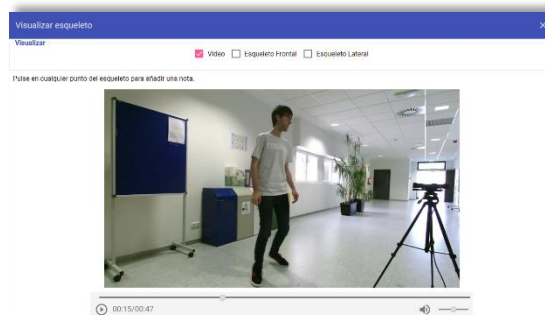


Ilustración 174. Test HU5.1 Prueba 1

2. Pausar y retomar el vídeo.

Prueba 1: El usuario intenta pausar el vídeo de una grabación.

Entrada: El usuario pulsa sobre el icono de  la columna *Grabación* y sobre el botón pausa del reproductor.

Salida esperada: La reproducción del vídeo se pausa.

Salida real: Salida esperada.

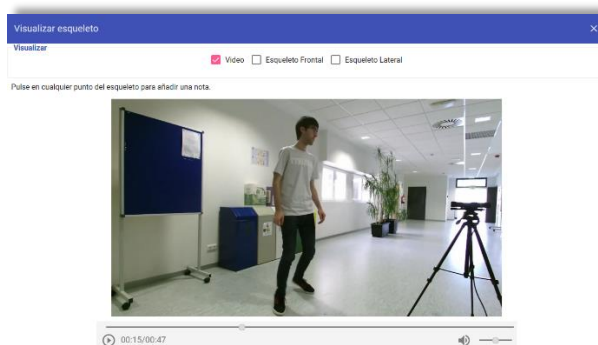


Ilustración 175. Test HU5.2 Prueba 1

Prueba 2: El usuario intenta reproducir el vídeo de una grabación.

Entrada: El usuario pulsa sobre el icono de ▶ la columna *Grabación* y sobre el botón reproducir del reproductor.

Salida esperada: La reproducción del vídeo se retoma.

Salida real: Salida esperada.

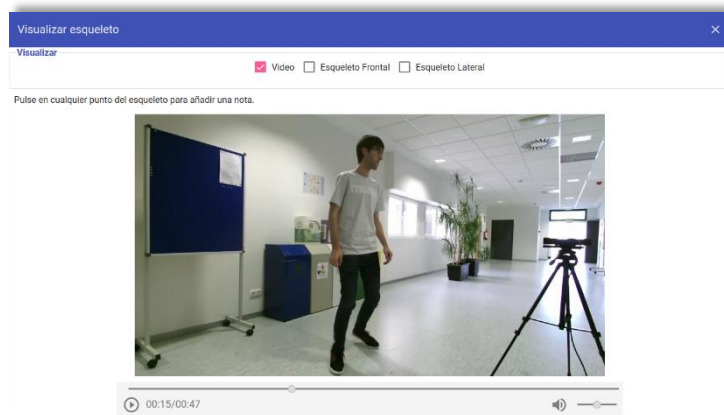


Ilustración 176. Test HU5.2 Prueba 2

3. Avanzar y retroceder en la reproducción del vídeo.

Prueba 1: El usuario intenta avanzar el vídeo de una grabación.

Entrada: El usuario pulsa sobre el icono de ▶ la columna *Grabación* y sobre un punto posterior al actual en la barra de reproducción.

Salida esperada: La reproducción del vídeo avanza hasta ese punto.

Salida real: Salida esperada.

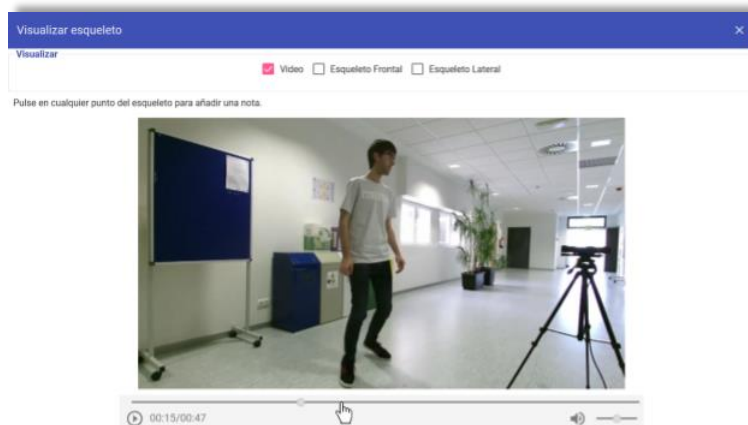



Ilustración 177. Test HU5.3 Prueba 1

Prueba 2: El usuario intenta retroceder el vídeo de una grabación.

Entrada: El usuario pulsa sobre el icono de  la columna *Grabación* y sobre un punto anterior al actual en la barra de reproducción.

Salida esperada: La reproducción del vídeo avanza hasta ese punto.

Salida real: Salida esperada.

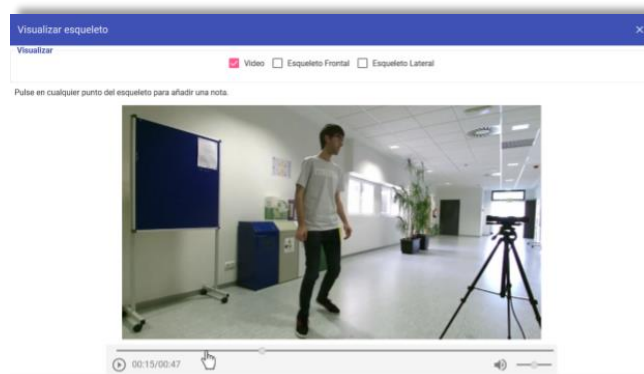



Ilustración 178. Test HU5.3 Prueba 2

HU7. Herramientas reproducción vídeo/esqueleto

Como cliente quiero poder controlar la reproducción del vídeo/esqueleto para poder pausar, reproducir, retroceder y avanzar y controlar el volumen del vídeo.

1. Pausar y retomar la reproducción, comprobando que continúan sincronizados

Prueba 1: El usuario pausa y retoma la reproducción.

Entrada: El usuario pulsa sobre el icono de  la columna *Grabaciones* y pulsa sobre el botón de pausa. A continuación, pulsa sobre el botón de reproducción.

Salida esperada: La reproducción continúa sincronizada.

Salida real: Salida esperada.

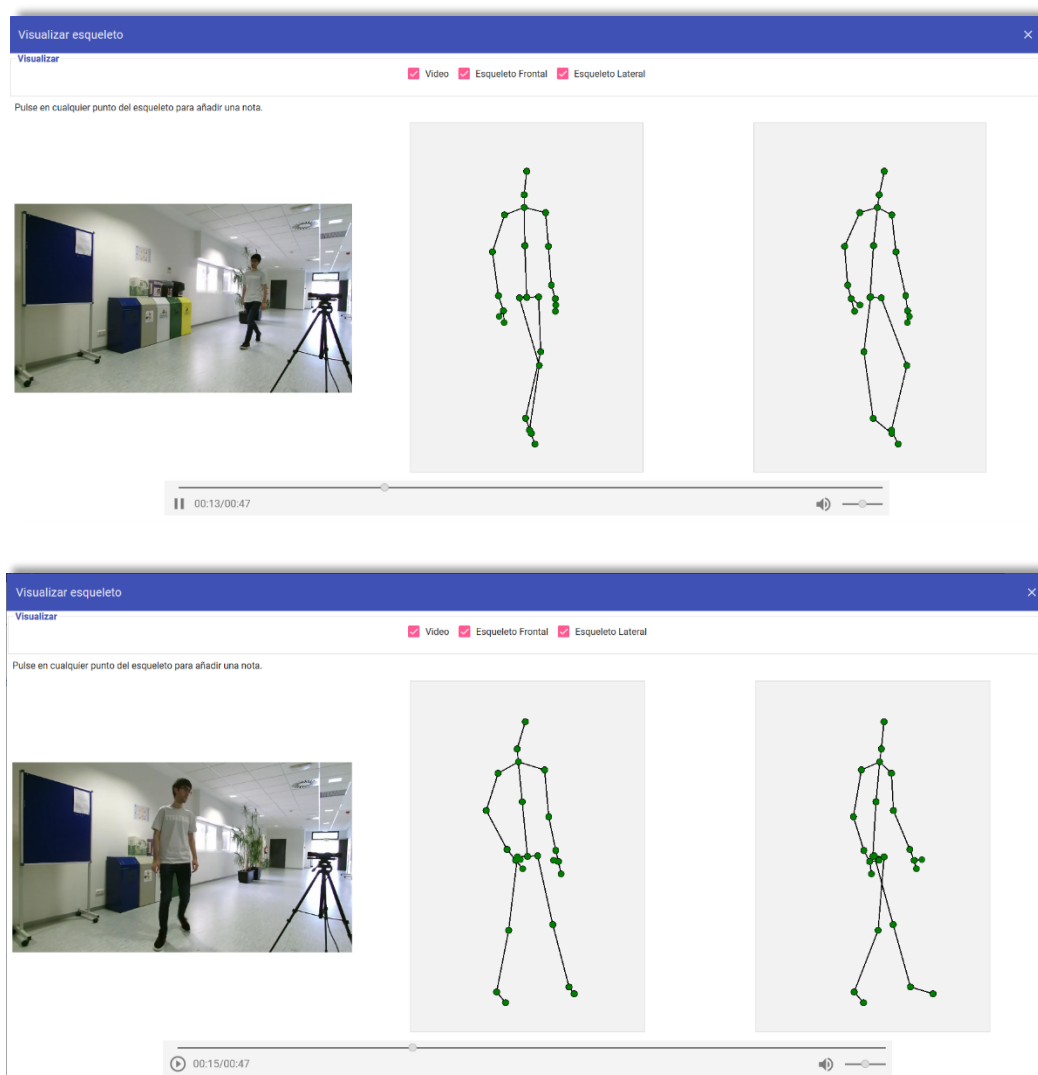


Ilustración 179. Test HU 7.1 Prueba 1

2. Avanzar en la reproducción, comprobando que continúan sincronizados.

Prueba 1: El usuario avanza en la reproducción.

Entrada: El usuario pulsa sobre el icono de ▶ la columna *Grabaciones* y pulsa sobre un punto posterior de la barra de progreso.

Salida esperada: La reproducción continúa sincronizada tras avanzar.


Salida real: Salida esperada.



Ilustración 180. Test HU7.2 Prueba 1

3. Retroceder en la reproducción, comprobando que continúan sincronizados.

Prueba 1: El usuario avanza en la reproducción.

Entrada: El usuario pulsa sobre el icono de  la columna *Grabaciones* y pulsa sobre un punto anterior de la barra de progreso. A continuación, pulsa sobre el botón de reproducción.

Salida esperada: La reproducción continúa sincronizada tras retroceder.

Salida real: Salida esperada.

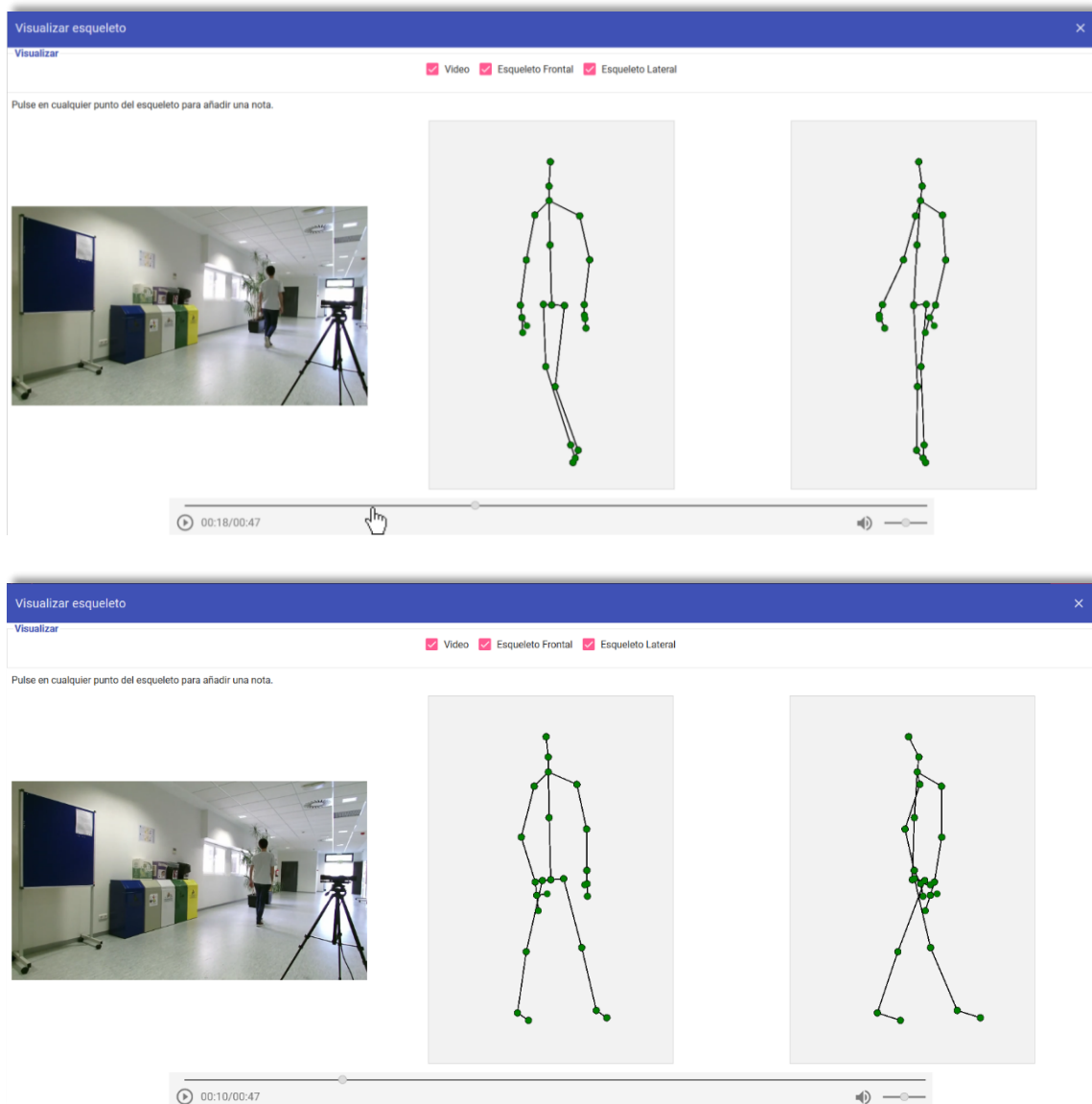




Ilustración 181. Test HU7.3 Prueba 1

4. Quitar el sonido al vídeo.

Prueba 1: El usuario quita el sonido del vídeo.

Entrada: El usuario pulsa sobre el icono de  la columna *Grabaciones* y pulsa sobre el icono para  silenciar el vídeo.

Salida esperada: La reproducción es silenciada.

Salida real: Salida esperada.

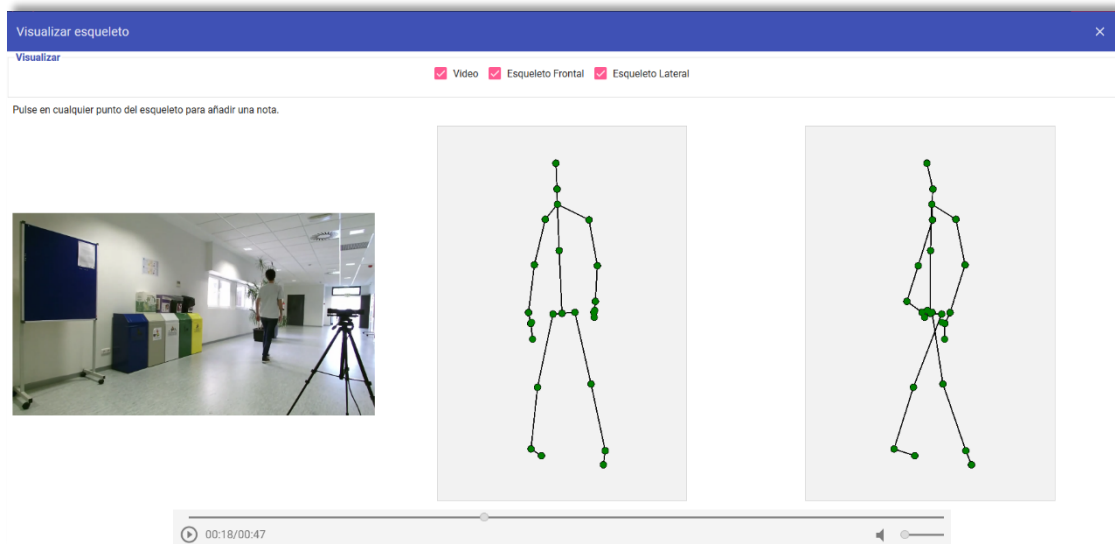
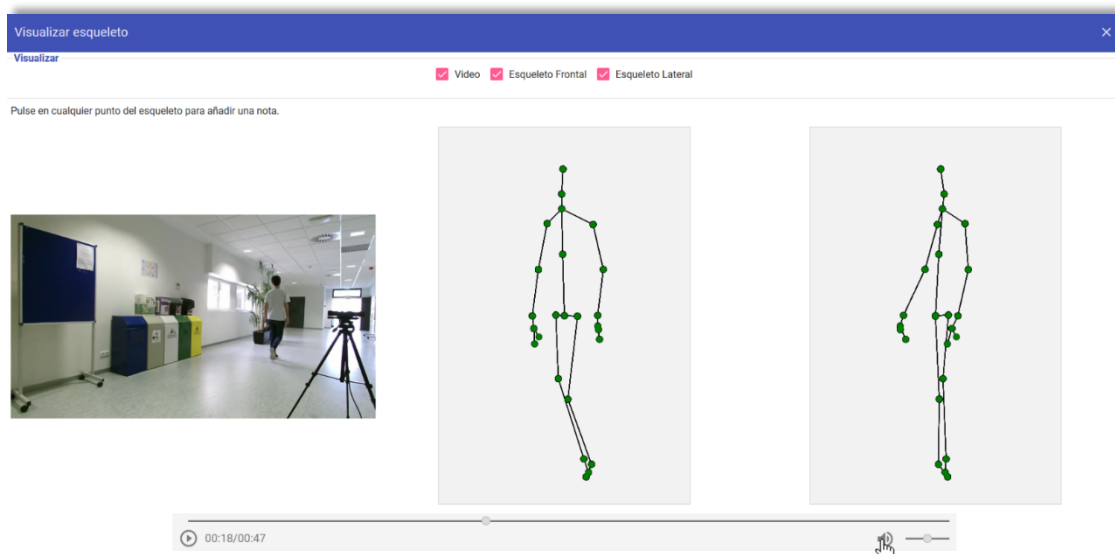


Ilustración 182. Test HU7.4 Prueba 1

5. Regular el volumen del vídeo.

Prueba 1: El usuario regula el volumen del vídeo.

Entrada: El usuario pulsa sobre el icono  de la columna *Grabaciones* y pulsa sobre en la barra de volumen para regular el volumen del vídeo.

Salida esperada: El volumen del vídeo cambia.

Salida real: Salida esperada.

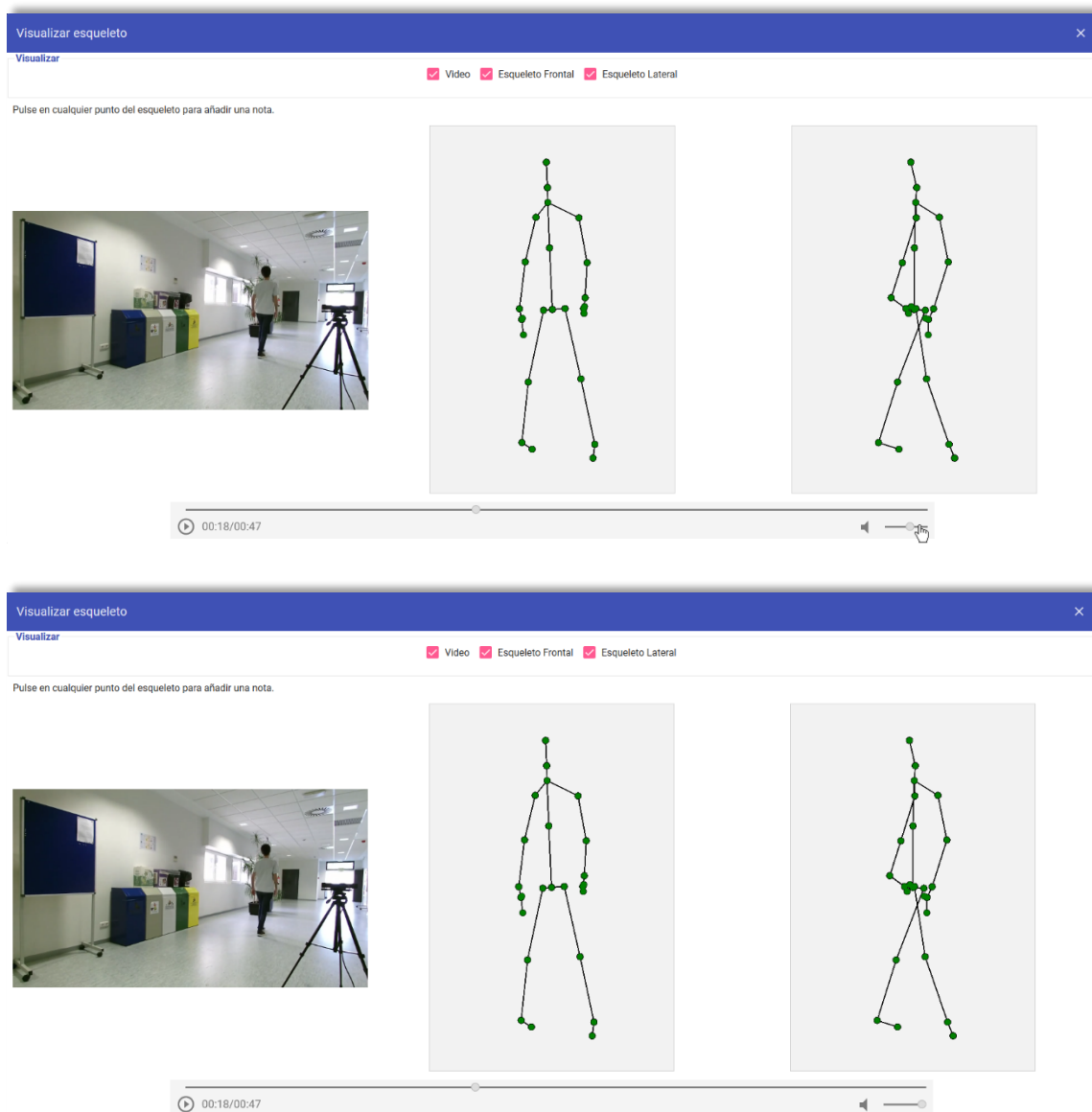


Ilustración 183. Test HU7.5 Prueba 1

HU9. Despliegue en Firebase

Como cliente quiero poder acceder a la aplicación web a través de Firebase para poder usar la aplicación y ver su evolución durante el desarrollo.

1. **Comprobar que el cliente puede acceder a la aplicación web desde la URL facilitada por Firebase.**

Prueba 1: El usuario accede a la aplicación web desde la URL facilitada por Firebase.

Entrada: URL a la aplicación: <https://gestion-pacientes-parkinson.firebaseio.com/>.

Salida esperada: Acceso satisfactorio a la aplicación.

Salida real: Salida esperada.

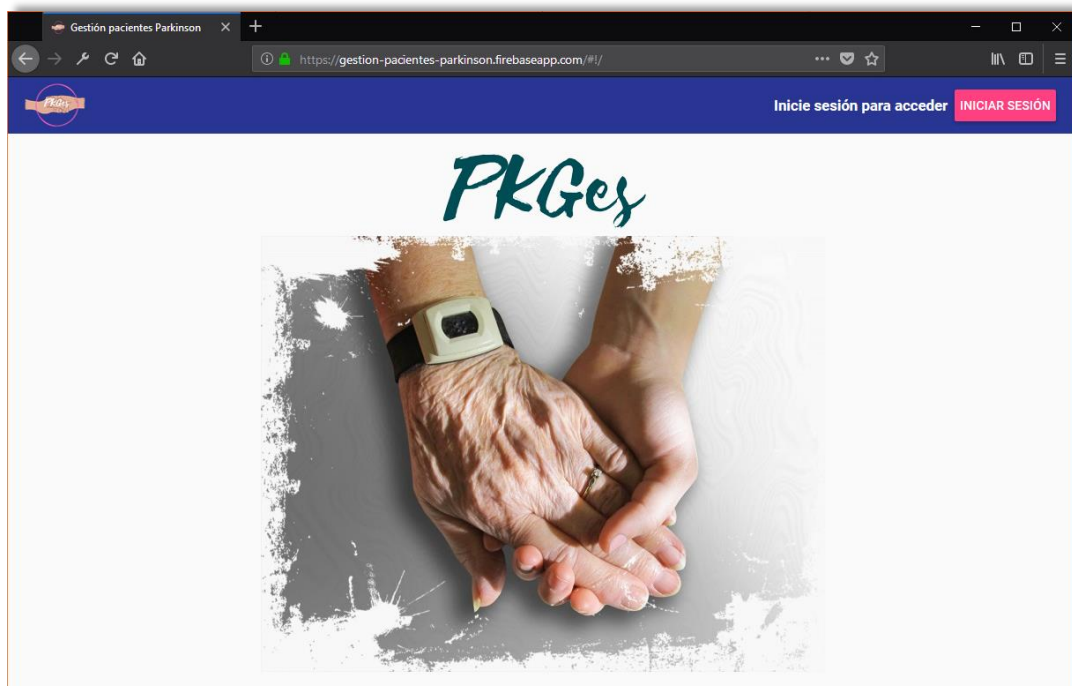


Ilustración 184. Test HU9.1 Prueba 1

HU10. Sincronización GitHub

Como desarrollador quiero que se sincronice el proyecto en un repositorio de GitHub mediante Git para poder realizar un control de versiones.

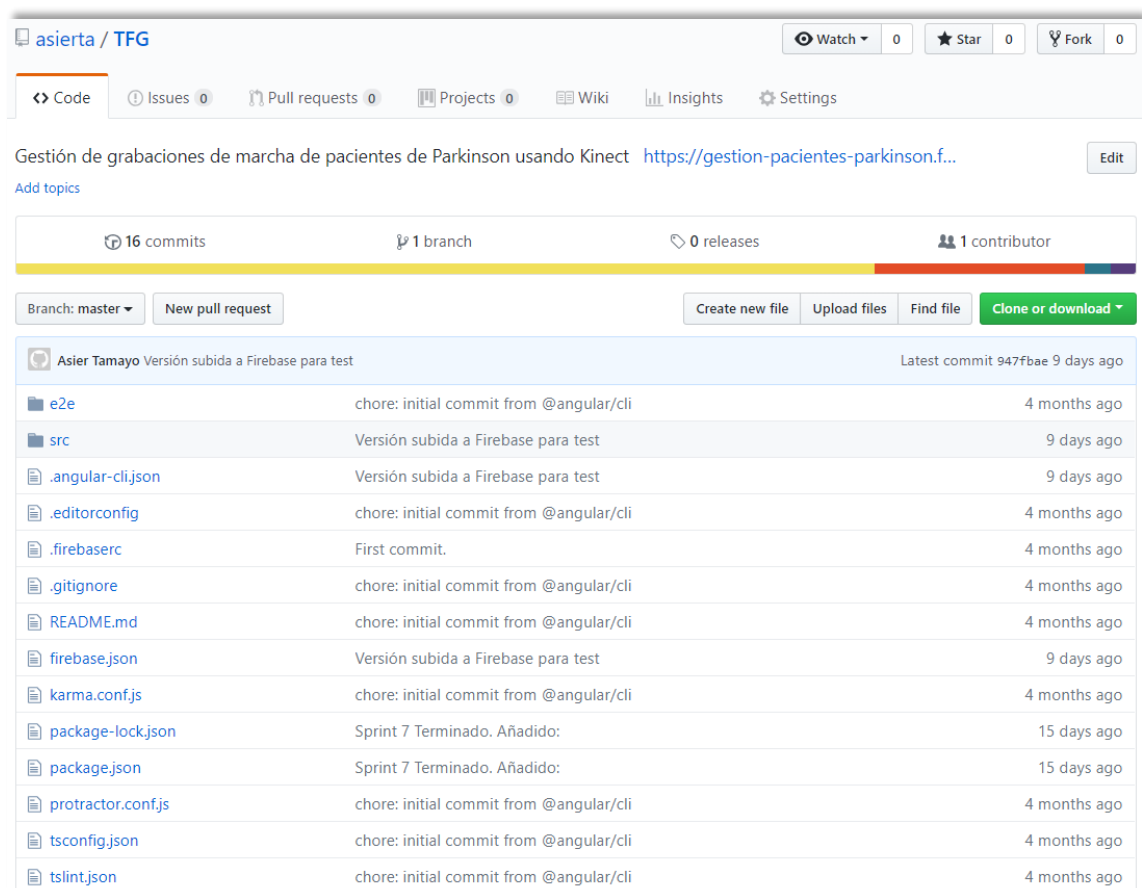
1. **Comprobar que el repositorio es accesible desde GitHub y que el control de versiones se realiza correctamente.**

Prueba 1: El usuario accede al repositorio de GitHub y comprueba que el código de la aplicación se encuentra accesible.

Entrada: URL al repositorio: <https://github.com/asierta/TFG>.

Salida esperada: Acceso satisfactorio al repositorio.

Salida real: Salida esperada.



The screenshot shows the GitHub interface for the repository 'asierta / TFG'. At the top, there are navigation tabs for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings. Below the repository name, there are statistics: 16 commits, 1 branch, 0 releases, and 1 contributor. A list of files and folders is displayed, including 'e2e', 'src', '.angular-cli.json', '.editorconfig', '.firebase', '.gitignore', 'README.md', 'firebase.json', 'karma.conf.js', 'package-lock.json', 'package.json', 'protractor.conf.js', 'tsconfig.json', and 'tslint.json'. Each item shows its commit message and the time since the last commit.

File/Folder	Commit Message	Time
e2e	chore: initial commit from @angular/cli	4 months ago
src	Versión subida a Firebase para test	9 days ago
.angular-cli.json	Versión subida a Firebase para test	9 days ago
.editorconfig	chore: initial commit from @angular/cli	4 months ago
.firebase	First commit.	4 months ago
.gitignore	chore: initial commit from @angular/cli	4 months ago
README.md	chore: initial commit from @angular/cli	4 months ago
firebase.json	Versión subida a Firebase para test	9 days ago
karma.conf.js	chore: initial commit from @angular/cli	4 months ago
package-lock.json	Sprint 7 Terminado. Añadido:	15 days ago
package.json	Sprint 7 Terminado. Añadido:	15 days ago
protractor.conf.js	chore: initial commit from @angular/cli	4 months ago
tsconfig.json	chore: initial commit from @angular/cli	4 months ago
tslint.json	chore: initial commit from @angular/cli	4 months ago

Ilustración 185. Test HU10.1 Prueba 1

2. Se ha actualizado el proyecto mínimo una vez por iteración.

Prueba 1: El usuario accede al repositorio de GitHub y comprueba que se ha realizado mínimo un commit por iteración.

Entrada: URL al repositorio: <https://github.com/asierta/TFG>.

Salida esperada: Acceso al repositorio y un mínimo de 8 commits.

Salida real: Salida esperada.

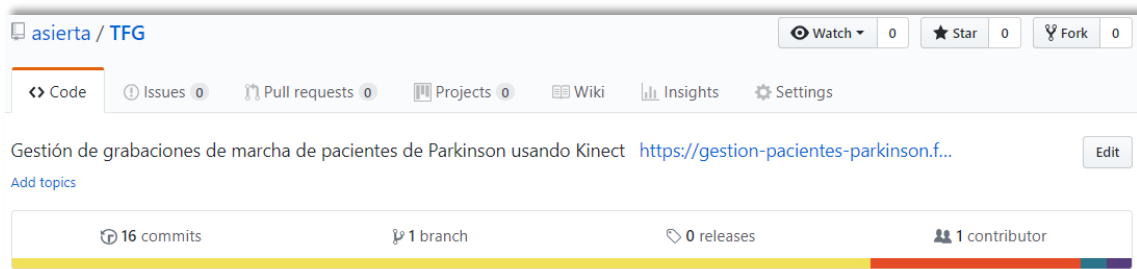


Ilustración 186. Test HU10.2 Prueba 1

HU11. Estructura fichero de datos

Como cliente quiero que se diseñe una nueva estructura del fichero de datos para que, en un futuro, si es necesario añadir más información en el fichero, la aplicación pueda seguir funcionando sin realizar demasiados cambios.

1. Comprobar que la estructura propuesta cumple con los requisitos y que es adecuada para el propósito definido.

Prueba 1: El fichero CSV se reproduce correctamente independientemente del número de columnas extra que aparezcan después de las relativas a las articulaciones.

Entrada: Fichero CSV con columnas extra.

Salida esperada: Reproducción correcta.

Salida real: Salida esperada.

```
esq24Inf;esq24X;esq24Y;esq24Z;esqRefInf;esqRefX;esqRefY;esqRefZ;esqRefCamInf;esqRefCamX;esqRefCamY;esqRefCamZ;Imantación;Altura;Peso
```

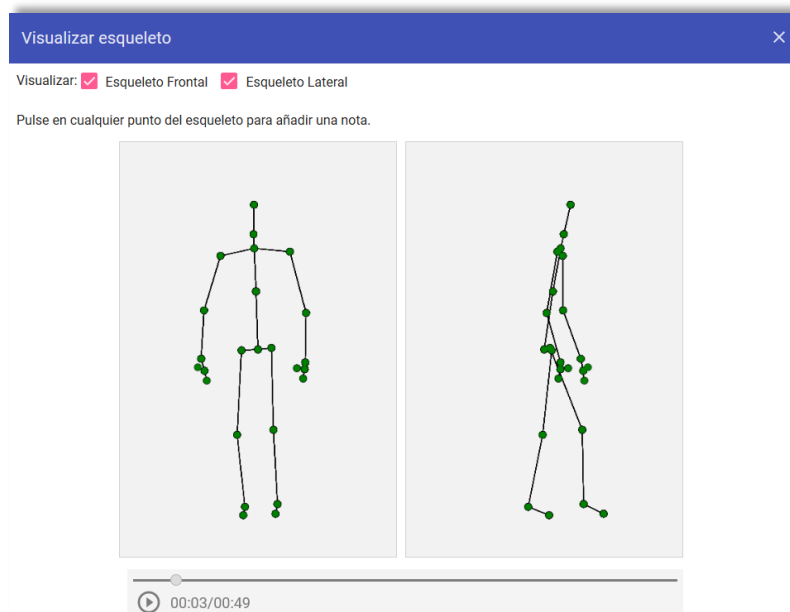


Ilustración 187. Test HU11.1 Prueba 1

Anexo D.

Actas reuniones

En este anexo se incluyen las actas de las reuniones de cierre de iteración. En ellas se presentan las conclusiones de cada una de las iteraciones del proyecto. Estas conclusiones están formadas por la explicación de qué se ha hecho en esa iteración, si se da el visto bueno o no a lo desarrollado y qué se va a hacer en la próxima iteración. Además, se incluye la dedicación a cada una de las tareas durante cada iteración.

Sprint 1.

Como resultado de la reunión retrospectiva del primer Sprint se ha dado el visto bueno a las historias de usuario trabajadas durante el mismo. Durante este primer Sprint se ha tenido el primer contacto con las principales tecnologías con las que se desarrollará el proyecto, AngularJS y Firebase. Es por ello que la mayor parte del primer Sprint se ha dedicado al estudio de estas tecnologías. Por otro lado, se ha visto la necesidad de que en la estructura de datos relativa a los esqueletos aparezcan datos sobre las características (nombre, valor y posición de visualización). También se ha decidido que la herramienta de visualización de los esqueletos sea la librería ThreeJS.

Historia de Usuario	Horas dedicadas	Notas
Despliegue en Firebase	4.5	Estudio Firebase y despliegue inicial con el boceto de la aplicación
Sincronización GitHub	1	Crear repositorio, asociarlo al proyecto y realizar commit inicial
Inicio de sesión	5	Versión inicial
Diseño Material Design	9	Estudio AngularJS y boceto inicial del diseño de la aplicación con AngularJS Material
Estructura de datos	8	Versión inicial del fichero de datos
Visualización gráfica de esqueletos	3	Estudio de las herramientas disponibles
Memoria	2.5	Planificación proyecto

Sprint 2.

Como resultado de la reunión retrospectiva del segundo Sprint se ha aprobado lo desarrollado en el mismo. En este Sprint se ha comenzado a desarrollar la visualización de esqueletos con la librería ThreeJS, siguiendo para ello ejemplos disponibles en su página web. De cara al siguiente Sprint se decidió continuar desarrollando la visualización de esqueletos con la librería ThreeJS pese a las dificultades surgidas por la falta de documentación. Además en el siguiente Sprint se mejorará la visualización mostrando cada articulación en el lugar correcto, actualmente no todas las articulaciones se muestran en la posición que les corresponde (los brazos aparecen elevados, la cabeza sobredimensionada y los pies estirados), también se acompañará cada articulación con un texto donde se indique información sobre la misma, asimismo se intentará encontrar el modo de colorear de otra forma las articulaciones inferidas para facilitar su identificación.

Historia de Usuario	Horas dedicadas	Notas
Visualización gráfica de esqueletos	15.5	Implementación de la visualización usando la librería ThreeJS. Problemas durante su desarrollo por falta de documentación

Sprint 3.

Como resultado de la reunión retrospectiva del tercer Sprint se ha decidido extender una semana más el plazo para intentar visualizar los esqueletos con ThreeJS, ya que los problemas con la localización de las articulaciones continúan. Sin embargo, se ha conseguido acompañar las articulaciones con texto informativo, así como colorear las articulaciones de rojo en caso de que sean inferidas.

Historia de Usuario	Horas dedicadas	Notas
Visualización gráfica de esqueletos	12	Incluir texto y color en las articulaciones. Continúan problemas con posición articulaciones
Inicio de sesión	1.5	Correcciones en la interfaz

Sprint 4.

Como resultado de la reunión retrospectiva del cuarto Sprint se ha decidido descartar la visualización de esqueletos con ThreeJS, ante la imposibilidad de posicionar las articulaciones correctamente, e implementarla mediante un canvas de HTML. En esta iteración se ha desarrollado una prueba realizada con el canvas donde la animación del esqueleto se visualiza correctamente coloreando las articulaciones en verde si no son inferidas y en rojo en caso de serlo.

Por otro lado, se ha decidido dejar de lado la visualización por el momento, para centrarse en la aplicación web. En el siguiente Sprint habrá que permitir la creación de pacientes y grabaciones con la posibilidad de relacionarlos entre ellos. Los pacientes tendrán varios campos comunes (edad, altura), pero siempre habrá opción de añadir campos personalizados, por lo que la base de datos tendrá que tener flexibilidad para poder añadir diferente número de atributos. Las grabaciones tendrán ciertos campos comunes (fecha, grabación) y también será posible añadir nuevos campos. La grabación se almacenará en la base de datos y será posible exportarla tanto en formato JSON como en CSV.

La creación de pacientes y grabaciones deben ser modulares de cara a la futura encriptación y desencriptación de los datos.

Tanto los pacientes como las grabaciones se tienen que poder eliminar.

Historia de Usuario	Horas dedicadas	Notas
Visualización gráfica de esqueletos	7	Intentar visualizar correctamente las articulaciones con ThreeJS. Prototipo de la visualización usando canvas de HTML

Sprint 5.

Como resultado de la reunión retrospectiva del cuarto Sprint se ha dado el visto bueno a lo desarrollado durante el Sprint. Esto es, la creación y borrado de usuarios y grabaciones y la posibilidad de relacionarlos entre ellos. Para cada funcionalidad se ha creado una nueva vista en la aplicación. De cara al Sprint 6 se ha decidido agrupar por un lado las funcionalidades del cliente y por otro lado las de las grabaciones. En el caso de los pacientes se mostrará una tabla con la información de todos los pacientes, con un botón para borrar el paciente de la fila. En esa misma pantalla se debe poder añadir también nuevos pacientes pulsando en un botón y mostrando el formulario en un *Dialog*. El formulario se modificará para que a la hora de añadir un atributo extra aparezcan como sugerencia alguno de los atributos extra añadidos anteriormente, dando la opción de seleccionar uno de ellos. Pulsando en un paciente será posible ver sus grabaciones, añadir una nueva y borrar una existente.

En el caso de las grabaciones se mostrará una tabla con la información de las grabaciones con la opción de borrado. También será posible añadir nuevas grabaciones a través de un *Dialog* y previsualizar el archivo csv. En cuanto a su almacenamiento, hay que sustituir el id actual por los identificadores generados por Firebase para evitar la sobrescritura de los datos.

Por otro lado, también se ha decidido que en el siguiente Sprint se va a hacer un análisis comparativo (algoritmos, rendimiento) de distintos métodos de encriptación de la información eligiendo el más apropiado. Probar a encriptar vídeos y pdfs además de la información.

Hay que probar a almacenar un vídeo en Firebase y visualizarlo.

Analizar también distintas plataformas de almacenamiento en la nube.

Historia de Usuario	Horas dedicadas	Notas
Entorno paciente	9.5	Creación y borrado de pacientes
Entorno grabación	8.5	Creación y borrado de grabaciones

Sprint 6.

Como resultado de la reunión retrospectiva del sexto Sprint se ha dado el visto bueno a lo desarrollado durante el Sprint. Esto es, la agrupación por un lado de todas las funcionalidades relativas a los pacientes y por otro lado las de las grabaciones, incluyendo la visualización en forma de tabla. De cara al Sprint 7 se ha decidido incluir la posibilidad de editar los datos de los pacientes y de las grabaciones. En cuanto a la encriptación, en el nuevo Sprint se requiere encriptar los datos para que el administrador del sistema no sea capaz de visualizarlos. Actualmente, cualquier usuario con acceso al proyecto en Firebase es capaz de visualizar la información almacenada. Para ello al iniciar sesión se solicitará al usuario una clave además de la contraseña, esa clave será la que se use para encriptar y desencriptar los datos de Firebase.

También se intentará encriptar los vídeos de los pacientes para almacenarlos de la misma forma. Además, también se implementará la visualización simultanea del vídeo de la grabación junto con el esqueleto.

Historia de Usuario	Horas dedicadas	Notas
Entorno paciente	18	Visualización en tabla
Entorno grabación	10.5	Visualización en tabla
Encriptación de la información	3.5	Estudio seguridad en Firebase
Visualización de vídeos	0.5	Prueba a reproducir vídeo almacenado en Firebase desde la aplicación web
Memoria	5.5	Herramientas usadas, seguridad Firebase

Sprint 7.

Como resultado de la reunión retrospectiva del séptimo Sprint se ha dado el visto bueno a lo desarrollado durante el mismo. Esto es, la encriptación de los datos con los que trabaja la aplicación, por un lado, la información sobre pacientes y grabaciones y por otro los ficheros asociados a las grabaciones. Para ello también se ha implementado la gestión de claves de cifrado sobre grupos de usuarios, que consiste en pedir al usuario una clave de cifrado en el momento de iniciar sesión, para hacer uso de esa clave para el cifrado y descifrado de los datos. También se ha desarrollado la reproducción simultanea del vídeo de la grabación junto con el esqueleto. Además, se ha reestructurado la BD teniendo en cuenta la existencia de distintos grupos de usuarios, también se ha mejorado la visualización de los datos en las tablas, añadiendo la posibilidad de ocultar columnas y editar todos los campos.

Es por ello que se da la aplicación por finalizada, a la espera de realizar alguna pequeña mejora y posterior revisión de bugs que encuentre el tester ajeno al desarrollo.

Se han propuesto como mejoras de la aplicación: añadir la visión lateral del esqueleto, añadir un mensaje durante la carga del vídeo, y comprobar que la clave de cifrado introducida por el usuario es correcta.

El resto del tiempo se dedicará a la redacción de la memoria del proyecto.

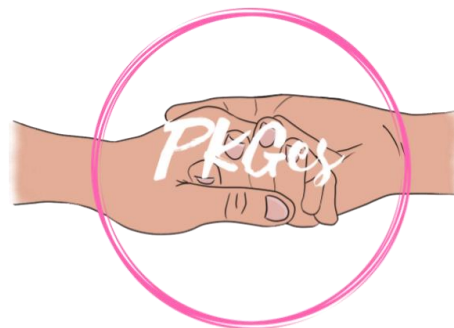
Historia de Usuario	Horas dedicadas	Notas
Entorno paciente	15	Edición de pacientes. Ordenación por columnas.
Entorno grabación	13	Edición grabaciones. Ordenación por columnas.
Encriptación de la información	16.5	Encriptación de la información de pacientes, grabaciones, CSV y vídeos de las grabaciones.
Visualización gráfica de esqueletos	3	Visualizar esqueleto desde el CSV.
Visualización de vídeos	5	Reproducción de vídeos cifrados almacenados en Firebase.

Visualizar los esqueletos en sincronía con el vídeo	7	Sincronizar el vídeo con los esqueletos.
Herramientas reproducción vídeo/esqueleto	4	Pausar/reproducir y avanzar y retroceder la reproducción hasta el punto deseado. Añadir notas durante la reproducción.
Inicio de sesión	1.5	Incluir clave de cifrado, restaurar contraseña.
Estructura de datos	1.5	Reestructuración de la BD.
Diseño Material Design	1.5	Mejoras en el diseño.
Memoria	8.5	Metodología, gestión seguridad,

Anexo E.

Manual de usuario

El siguiente manual tiene como objetivo facilitar la experiencia de cualquier usuario de la aplicación. En él se describen cada una de las funcionalidades disponibles y se explica cómo hacer uso de ellas. El manual se divide en tres secciones, la primera dedicada a los usuarios, donde se explica todo lo relacionado con el acceso a la aplicación. La segunda se centra en todas las funcionalidades en torno a los pacientes y la tercera en las funcionalidades en torno a las grabaciones. La aplicación desarrollada está disponible a través del siguiente enlace: <https://gestion-pacientes-parkinson.firebaseio.com/>.



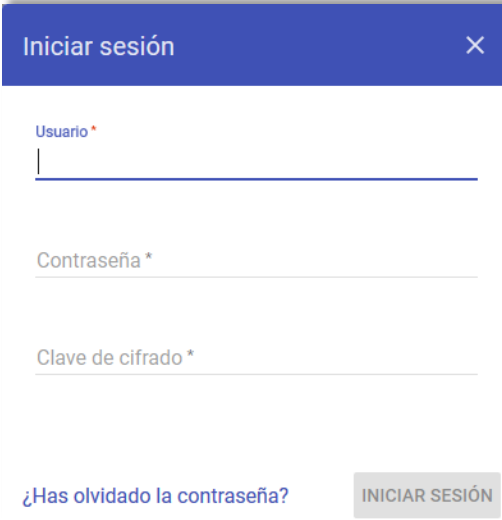
Índice

1. Usuarios.....	206
1.1 Inicio de sesión	206
1.2 Cambio de contraseña.....	207
1.3 Clave de cifrado	208
2. Pacientes	208
2.1 Registrar paciente.....	208
2.2 Visualizar pacientes	210
2.3 Editar pacientes	210
2.4 Eliminar pacientes	211
2.5 Asignar grabación a paciente.....	211
3. Grabaciones.....	212
3.1 Registrar grabación.....	212
3.2 Visualizar grabaciones	213
3.3 Editar grabaciones	214
3.4 Reproducir grabaciones.....	215
3.5 Descargar grabación	215
3.6 Previsualizar archivo CSV	216

1. Usuarios

1.1. Inicio de sesión

Para iniciar sesión en la aplicación acceda a la página principal de la misma, <https://gestion-pacientes-parkinson.firebaseio.com>, y pulse sobre el botón “Iniciar sesión”, situado en la esquina superior derecha de la página. A continuación, se mostrará una ventana emergente donde deberá introducir su correo electrónico, contraseña y clave de cifrado.

Una ventana emergente con un encabezado azul que dice "Iniciar sesión" y un botón de cerrar "X". El formulario contiene tres campos de texto: "Usuario *" con un cursor de texto, "Contraseña *" y "Clave de cifrado *". Debajo de los campos hay un enlace "¿Has olvidado la contraseña?" y un botón gris "INICIAR SESIÓN".

Iniciar sesión

Usuario *

Contraseña *

Clave de cifrado *

¿Has olvidado la contraseña?

INICIAR SESIÓN

Ilustración 188. Inicio de sesión

Si es la primera vez que inicia sesión, deberá introducir el usuario y la contraseña de la que le haya provisto el administrador del sistema. Es recomendable que realice un cambio de contraseña en el primer inicio de sesión.

Si quiere conocer más detalles de la clave de cifrado diríjase a la sección [1.3 Clave de cifrado](#).

Si no recuerda su contraseña o desea cambiarla, puede hacerlo haciendo clic en el enlace ¿Has olvidado la contraseña? ubicado a la izquierda del botón “Iniciar sesión”. Para conocer el proceso de recuperación de contraseña diríjase a la sección [1.2 Cambio de contraseña](#).

1.2. Cambio de contraseña

Para cambiar de contraseña diríjase a la página principal de la aplicación, <https://gestion-pacientes-parkinson.firebaseio.com>, y pulse sobre el botón “Iniciar sesión”, situado en la esquina superior derecha de la página. A continuación, se mostrará una ventana emergente donde deberá hacer clic en el enlace ¿Has olvidado la contraseña? ubicado a la izquierda del botón “Iniciar sesión”.

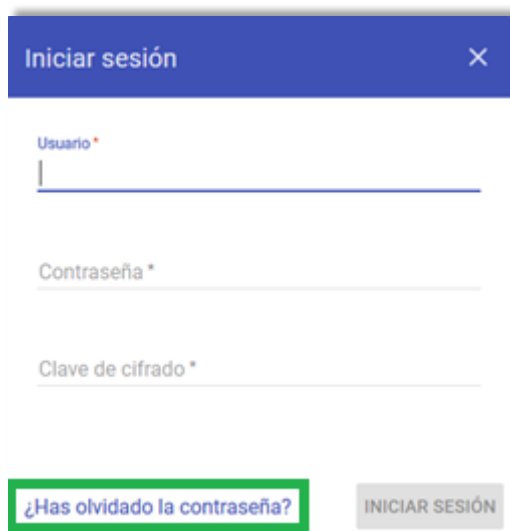


Ilustración 189. Inicio de sesión. Cambio de contraseña

A continuación, se mostrará una nueva ventana emergente donde deberá introducir su correo electrónico y pulsar sobre el botón “Restaurar”. Si existe una cuenta para el correo electrónico introducido, recibirá, con la mayor brevedad posible, un correo con las instrucciones para restablecer su contraseña.

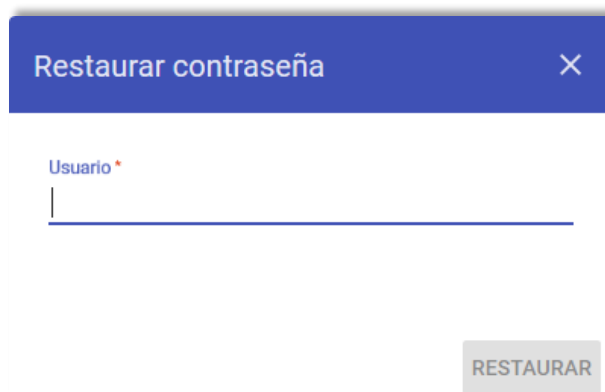


Ilustración 190. Cambio de contraseña

1.3. Clave de cifrado

La clave de cifrado es la clave con la que se encripta toda la información de un grupo de usuarios, por lo tanto, todos los miembros de un grupo comparten clave de cifrado.


Si no conoce la clave de cifrado de su grupo, pregúntesela al encargado de su grupo.

Si es el primer usuario del grupo que accede a la aplicación, deberá establecer la clave de cifrado para su grupo. Para establecer la clave de cifrado, simplemente debe introducirla en el primer inicio de sesión. La clave que haya introducido será la que se use para encriptar toda la información de su grupo de usuarios.

2. Pacientes

Para acceder a cualquiera de las funcionalidades descritas en este apartado debe haber iniciado sesión previamente. Para iniciar sesión siga las indicaciones de la sección [1.1 Inicio de sesión](#).

2.1. Registrar paciente

Para registrar un nuevo paciente diríjase a la sección Pacientes de la aplicación. Para ello acceda al menú, accesible desde el icono situado  en la esquina superior izquierda de la página y pulse sobre “Pacientes”.

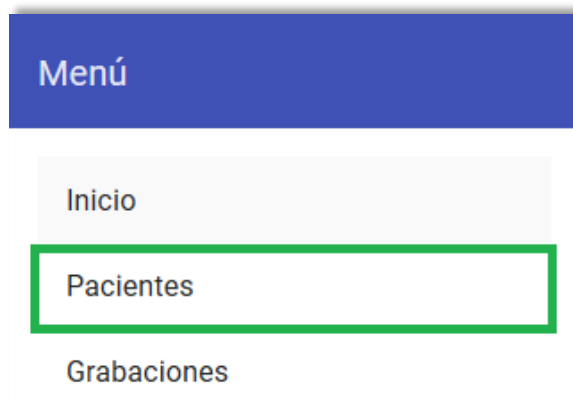



Ilustración 191. Menú Pacientes

A continuación, se mostrará una tabla con todos los pacientes asociados a su grupo de usuarios. Para añadir un nuevo paciente pulse sobre el icono  situado en la esquina superior derecha de la tabla. Deberá rellenar todos los campos marcados con un *, el resto de campos los podrá cumplimentar en caso de que lo desee. Para **asignar** una **grabación** al **paciente** siga las indicaciones de la sección [2.5 Asignar grabaciones](#).

También puede **añadir** tantos **campos extra** como desee. Para ello basta con pulsar sobre el botón + de la esquina superior izquierda.

Por cada campo extra deberá indicar un identificador y un valor. Al pulsar sobre el campo del identificador se le propondrán los identificadores usados con anterioridad. En caso de no aparecer el identificador que está buscando, bastará con escribirlo y pulsar sobre el botón “crear” que aparecerá debajo del identificador.

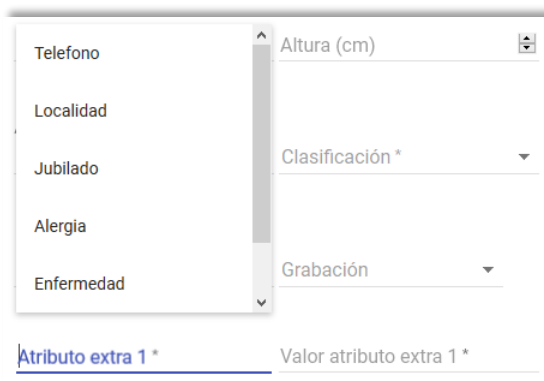


Ilustración 192. Campo extra Paciente

Si desea **eliminar** el último **campo extra** añadido, deberá pulsar sobre el botón – de la esquina superior izquierda del formulario.

Una vez completado el formulario pulse sobre el botón “Crear paciente” para completar el registro del nuevo paciente.

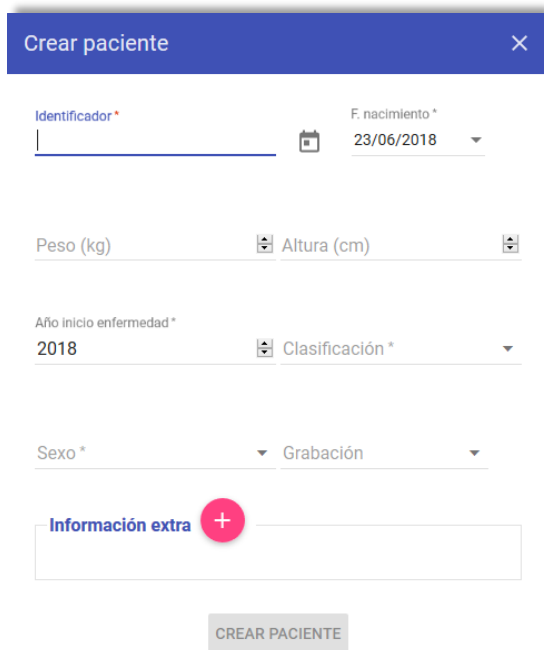




Ilustración 193. Registro paciente

2.2. Visualizar pacientes

Para visualizar los pacientes diríjase a la sección Pacientes de la aplicación. Para ello acceda al menú, accesible desde el icono  situado en la esquina superior izquierda de la página y pulse sobre “Pacientes”. Se mostrará una tabla con todos los pacientes disponibles junto con su información. Puede seleccionar qué información quiere ver acerca de los pacientes desde el menú configuración, situado inmediatamente encima de la tabla. Los datos que aparezcan seleccionados en este menú, serán los que se estén visualizando en la tabla. Para **ocultar una columna**, simplemente desmarque el nombre de la columna en el menú *configuración*. Si quiere que se muestre de nuevo, márkuelo.

También puede **reordenar la tabla** en función de una columna. Para ello pulse sobre el nombre de la columna en base a la que quiere ordenar los pacientes.

Es posible **filtrar los pacientes** por su identificador. Pulse sobre el icono  , situado en la esquina superior derecha de la tabla. Se mostrará una barra de búsqueda donde deberá introducir el texto por el que desea filtrar los pacientes.

Si el paciente tiene asignada alguna grabación, aparecerá el botón Ver en la columna Grabaciones, si desea **ver las grabaciones de un paciente** pulse el botón.



Configuración

De-Seleccionar todas Fecha nacimiento Edad Sexo Altura Peso Inicio enfermedad Clasificación Grabaciones Atributos extra


Pacientes 🔍 🔄 + ⚙️

<input type="checkbox"/>	Identificador ↑	Fecha Nacimiento	Edad	Sexo	Altura(cm)	Peso	Inicio enfermedad	Clasificación	Grabaciones	Atributos extra
<input type="checkbox"/>	Han Solo	20/10/1951	66	Hombre -	190	Añadir peso	2000	2	-	VER
<input type="checkbox"/>	Lela	21/09/1979	38	Mujer -	Añadir altura	58	2012	1	-	VER
<input type="checkbox"/>	Snapé	21/04/1962	56	Mujer -	158	85	2015	2	-	AÑADIR

Página: 1 Filas por página: 10 1-3 de 3 < >

Ilustración 194. Visualizar pacientes


2.3. Editar pacientes

Para editar un paciente diríjase a la sección Pacientes de la aplicación. Para ello acceda al menú, accesible desde el icono  situado en la esquina superior izquierda de la página y pulse sobre “Pacientes”. Se mostrará una tabla con todos los pacientes disponibles junto con su información. Pulse sobre el dato que quiera modificar, se mostrará una ventana emergente en la que aparecerá el valor actual para ese dato y desde ahí podrá editar el valor actual por el que desee. También puede editar los atributos extra del paciente, para ello pulse sobre el botón situado en la columna correspondiente a Atributos extra. Aparecerá una ventana emergente donde se mostrarán los atributos extra que tiene el paciente y desde ahí podrá añadir o eliminar atributos extra. Para añadir uno, pulse sobre el botón + de la esquina superior izquierda. Si desea eliminar el último campo extra, deberá pulsar sobre el botón – de la esquina superior izquierda del formulario.

Identificador	Fecha Nacimiento	Edad	Sexo	Altura(cm)	Peso	Inicio enfermedad	Clasificación	Grabaciones	Atributos extra
<input type="checkbox"/> Han Solo	20/10/1951	66	Hombre	190	Añadir peso	2000	2	-	VER
<input type="checkbox"/> Leia	21/09/1979	38	Mujer	Añadir altura	58	2000	1	-	VER
<input type="checkbox"/> Snape	21/04/1962	56	Mujer	158	85	2000	2	-	AÑADIR

Ilustración 195. Editar paciente

2.4. Eliminar pacientes

Para eliminar un paciente diríjase a la sección Pacientes de la aplicación. Para ello acceda al menú, accesible desde el icono  situado en la esquina superior izquierda de la página y pulse sobre “Pacientes”. Se mostrará una tabla con todos los pacientes disponibles junto con su información. Seleccione los pacientes que desee borrar pulsando sobre el cuadrado que aparece en la parte izquierda del identificador de los pacientes. Una vez seleccionados los pacientes, pulse sobre el icono con forma de papelera que aparece en la parte superior izquierda de la tabla. Se le solicitará que confirme que desea borrar los pacientes seleccionados, puesto que esta acción es irreversible. Cuando elimine un paciente las grabaciones asociadas a este paciente pasarán a estar sin asignar.

Identificador	Fecha Nacimiento	Edad	Sexo	Altura(cm)	Peso	Inicio enfermedad	Clasificación	Grabaciones	Atributos extra
<input type="checkbox"/> Han Solo	24/05/1959	59	Hombre	180	Añadir peso	2000	2	-	VER
<input type="checkbox"/> Leia	21/09/1979	38	Mujer			2012	3	-	AÑADIR
<input type="checkbox"/> Luke	12/10/1944	73	Hombre			2015	2	-	AÑADIR
<input checked="" type="checkbox"/> Paquita Salas	25/03/1989	29	Mujer			2016	3	-	VER
<input type="checkbox"/> Snape	12/11/1960	57	Hombre			2015	2	-	AÑADIR

Ilustración 196. Eliminar pacientes

2.5. Asignar grabación a paciente

Puede asignar una grabación a un paciente de varias formas. La primera de ellas sería durante el registro del usuario, en él aparece el campo no obligatorio *Grabación*, pulsando sobre este campo se mostrarán las grabaciones disponibles (grabaciones pertenecientes a su mismo grupo de usuarios que todavía no tienen un paciente asignado). Para registrar un nuevo paciente siga las indicaciones de la sección [2.1 Registrar paciente](#).


El segundo método para asignar una grabación a un paciente sería durante la creación de una grabación, en el formulario aparece el campo no obligatorio *Paciente*, pulsando sobre este campo se mostrarán los pacientes disponibles (pacientes pertenecientes a su mismo grupo). Para registrar una nueva grabación siga las indicaciones de la sección [3.1 Registrar grabación](#).

El último método consiste en editar la grabación, en ese caso pulse sobre la columna relativa al paciente, se mostrarán los pacientes disponibles (pacientes pertenecientes a su mismo grupo) y seleccione el paciente al que quiere asignarle la grabación. Para saber cómo editar una grabación diríjase a la sección [3.3 Editar grabaciones](#).

3. Grabaciones

Para acceder a cualquiera de las funcionalidades descritas en este apartado debe haber iniciado sesión previamente. Para iniciar sesión siga las indicaciones de la sección [1.1 Inicio de sesión](#).

3.1. Registrar grabación

Para registrar una nueva grabación diríjase a la sección Grabaciones de la aplicación. Para ello acceda al menú, accesible desde el icono  situado en la esquina superior izquierda de la página y pulse sobre “Grabaciones”.

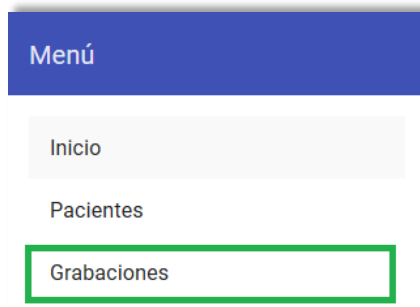



Ilustración 197 Menú grabaciones

A continuación, se mostrará una tabla con todas las grabaciones asociadas a su grupo de usuarios. Para añadir una nueva grabación pulse sobre el icono situado  en la esquina superior derecha de la tabla. Deberá rellenar todos los campos marcados con un *, el resto de campos los podrá complementar en caso de que lo desee. También puede **añadir** tantos **campos extra** como desee. Para ello basta con pulsar sobre el botón + de la esquina superior izquierda. Por cada campo extra deberá indicar un identificador y un valor. Al pulsar sobre el campo del identificador se le propondrán los identificadores usados con anterioridad. En caso de no aparecer el identificador que está buscando, bastará con escribirlo y pulsar sobre el botón “crear” que aparecerá debajo del identificador. Si desea **eliminar** el último **campo extra** añadido, deberá pulsar sobre el botón – de la esquina superior izquierda del formulario.


Una captura de pantalla de un formulario de registro de grabación. El formulario tiene un encabezado con el texto "Atributo extra 1 *". Debajo del encabezado, hay un campo de texto con el texto "Duración" y un campo de texto con el texto "Valor atributo extra 1 *". Debajo de estos campos, hay un botón con el texto "GRABACIÓN".

Ilustración 198. Campo extra Grabación


Una vez completado el formulario pulse sobre el botón “Crear grabación” para completar el registro de la nueva grabación.

Ilustración 199. Registro grabación

3.2. Visualizar grabaciones

Para visualizar las grabaciones diríjase a la sección Grabaciones de la aplicación. Para ello acceda al menú, accesible desde el icono  situado en la esquina superior izquierda de la página y pulse sobre “Grabaciones”. Se mostrará una tabla con todas las grabaciones disponibles junto con su información. Puede seleccionar qué información quiere ver acerca de las grabaciones desde el menú configuración, situado inmediatamente encima de la tabla. Los datos que aparezcan seleccionados en este menú, serán los que se estén visualizando en la tabla. Para **ocultar una columna**, simplemente desmarque el nombre de la columna en el menú *configuración*. Si quiere que se muestre de nuevo, márkuelo.


También puede **reordenar la tabla** en función de una columna. Para ello pulse sobre el nombre de la columna en base a la que quiere ordenar las grabaciones.

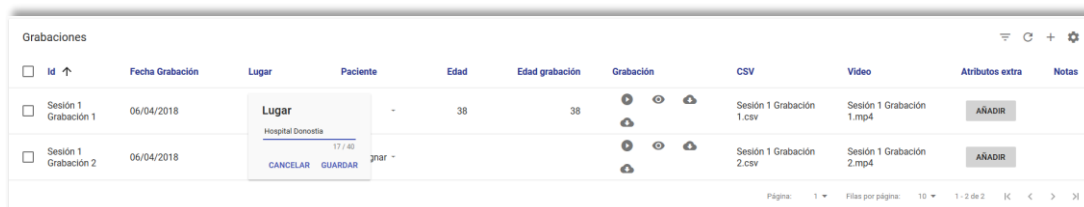
Es posible **filtrar las grabaciones** por su identificador. Pulse sobre el icono , situado en la esquina superior derecha de la tabla. Se mostrará una barra de búsqueda donde deberá introducir el texto por el que desea filtrar las grabaciones.

<input type="checkbox"/>	Id ↑	Fecha Grabación	Lugar	Paciente	Edad	Edad grabación	Grabación	CSV	Video	Atributos extra	Notas
<input type="checkbox"/>	Sesión 1 Grabación 1	06/04/2018	Hospital Donostia	Lela -	38	38		Sesión 1 Grabación 1.csv	Sesión 1 Grabación 1.mp4	<input type="checkbox"/>	añadir
<input type="checkbox"/>	Sesión 1 Grabación 2	06/04/2018	Hospital Donostia	Sin Asignar -				Sesión 1 Grabación 2.csv	Sesión 1 Grabación 2.mp4	<input type="checkbox"/>	añadir

Ilustración 200. Visualizar grabaciones

3.3. Editar grabaciones

Para editar las grabaciones diríjase a la sección Grabaciones de la aplicación. Para ello acceda al menú, accesible desde el icono  situado en la esquina superior izquierda de la página y pulse sobre “Grabaciones”. Se mostrará una tabla con todas las grabaciones disponibles junto con su información. Pulse sobre el dato que quiera modificar, se mostrará una ventana emergente en la que aparecerá el valor actual para ese dato y desde ahí podrá editar el valor actual por el que desee. También puede editar los atributos extra de la grabación, para ello pulse sobre el botón situado en la columna correspondiente a Atributos extra. Aparecerá una ventana emergente donde se mostrarán los atributos extra que tiene la grabación y desde ahí podrá añadir o eliminar atributos extra. Para añadir uno, pulse sobre el botón + de la esquina superior izquierda. Si desea eliminar el último campo extra añadido, deberá pulsar sobre el botón – de la esquina superior izquierda del formulario.



Id	Fecha Grabación	Lugar	Paciente	Edad	Edad grabación	Grabación	CSV	Video	Atributos extra	Notas
Sesión 1 Grabación 1	06/04/2018	Hospital Donostia	17 / 40	38	38		Sesión 1 Grabación 1.csv	Sesión 1 Grabación 1.mp4	AÑADIR	
Sesión 1 Grabación 2	06/04/2018						Sesión 1 Grabación 2.csv	Sesión 1 Grabación 2.mp4	AÑADIR	

Ilustración 201. Editar grabación

De esta misma forma también puede **asignar un paciente** a la grabación o editar el paciente asignado actualmente. Si quiere conocer con más detalle como asignar grabaciones a un paciente diríjase a la sección [2.5 Asignar grabación a paciente](#).

También puede **añadir un fichero de vídeo o** editar el fichero **CSV** o de Vídeo actual pulsando en la columna correspondiente.

3.4. Eliminar grabaciones




Para eliminar una grabación diríjase a la sección Grabaciones de la aplicación. Para ello acceda al menú, accesible desde el icono situado en  la esquina superior izquierda de la página y pulse sobre “Grabaciones”. Se mostrará una tabla con todas las grabaciones disponibles junto con su información. Seleccione las grabaciones que desee borrar pulsando sobre el cuadrado que aparece en la parte izquierda del identificador de las grabaciones. Una vez seleccionadas, pulse sobre el icono con forma de papelera que aparece en la parte superior izquierda de la tabla. Se le solicitará que confirme que desea borrar las grabaciones seleccionadas, puesto que esta acción es irreversible. Cuando elimine una grabación, el paciente dejará de estar asociado a la grabación.



Ilustración 202. Eliminar grabaciones

3.5. Reproducir grabaciones

Para reproducir las grabaciones diríjase a la sección Grabaciones de la aplicación. Para ello acceda al menú, accesible desde el icono situado en  la esquina superior izquierda de la página y pulse sobre “Grabaciones”. Se mostrará una tabla con todas las grabaciones disponibles junto con su información. Para reproducir una grabación pulse sobre el icono  situado en la columna Grabación. Se abrirá una ventana emergente en la que se mostrará la visión frontal y lateral del esqueleto extraído por Kinect, y el vídeo en caso de haberlo añadido.

Es posible pausar/reproducir, avanzar y retroceder a lo largo de la reproducción a su antojo. Además, puede **tomar anotaciones** durante la reproducción. Para ello debe pulsar en el punto del esqueleto sobre el que desee tomar la anotación. Aparecerá una ventana emergente donde puede insertar la nota, que posteriormente se visualizará sobre el esqueleto.

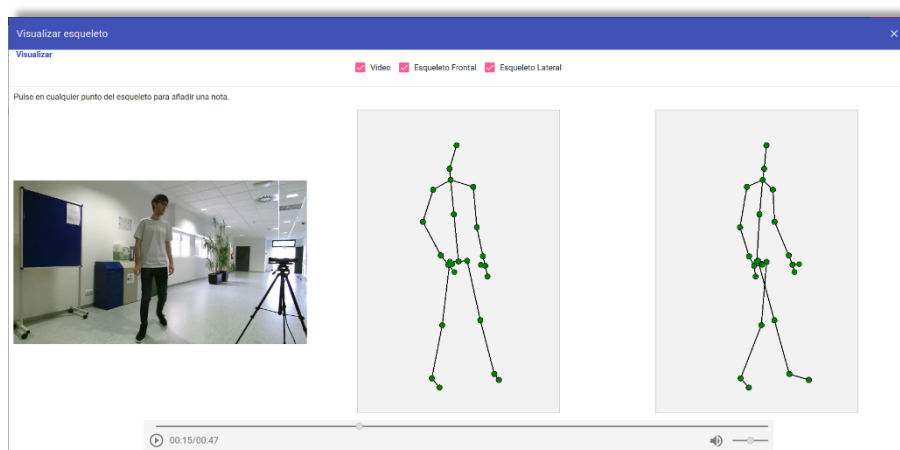






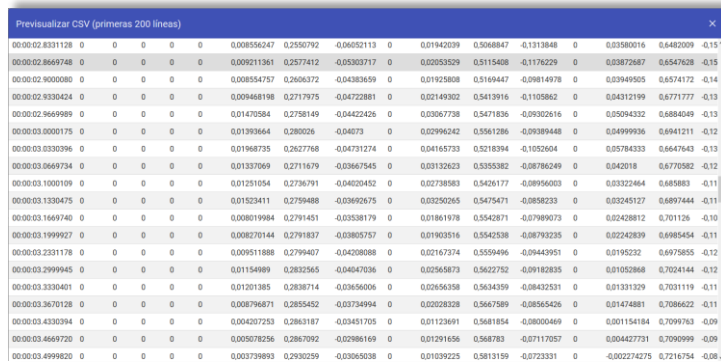
Ilustración 203. Reproducción grabación

3.6. Descargar grabación

Para descargar las grabaciones diríjase a la sección Grabaciones de la aplicación. Para ello acceda al menú, accesible desde el icono situado en  en la esquina superior izquierda de la página y pulse sobre “Grabaciones”. Se mostrará una tabla con todas las grabaciones disponibles junto con su información. Para descargar una grabación pulse sobre uno de los iconos situados  en la columna Grabación. Puede descargar la grabación tanto en formato CSV como en formato JSON, para ello pulse sobre el icono correspondiente al tipo de fichero. El CSV que se descargará será el que se haya incluido a la hora de crear la grabación. El JSON incluirá toda la información relativa a las articulaciones contenida en el CSV y además las notas añadidas a la grabación por usted o por otros miembros del grupo.

3.7. Previsualizar archivo CSV

Para previsualizar el CSV diríjase a la sección Grabaciones de la aplicación. Para ello acceda al menú, accesible desde el icono situado  en la esquina superior izquierda de la página y pulse sobre “Grabaciones”. Se mostrará una tabla con todas las grabaciones disponibles junto con su información. Pulse sobre el icono situado en la  columna Grabación. Se abrirá una ventana emergente en la que se mostrarán las primeras 200 líneas del fichero CSV de la grabación.



Previsualizar CSV (primeras 200 líneas)																															
00:00:02.833128	0	0	0	0	0.00855247	0.2550792	-0.06052113	0	0.01942039	0.5088847	-0.1313848	0	0.03580016	0.4482009	-0.15	0.000028669748	0	0	0	0.009211361	0.2577412	-0.05303717	0	0.02053529	0.5115408	-0.1176229	0	0.03872687	0.6547628	-0.15	
00:00:02.9000080	0	0	0	0	0.008554757	0.2606372	-0.04383659	0	0.01925808	0.5169447	-0.08814978	0	0.02949505	0.6574172	-0.14	00:00:02.9330424	0	0	0	0	0.009468198	0.2717975	-0.04722881	0	0.02149302	0.5413916	-0.1105562	0	0.04312199	0.6771777	-0.13
00:00:02.9669989	0	0	0	0	0.01470584	0.2758149	-0.04422426	0	0.03067738	0.5471836	-0.09302616	0	0.05094332	0.6884049	-0.13	00:00:03.0000175	0	0	0	0	0.01393664	0.280026	-0.04073	0	0.02996242	0.5561286	-0.09389448	0	0.04999936	0.6941211	-0.12
00:00:03.0330396	0	0	0	0	0.01968735	0.2627768	-0.04731274	0	0.04165733	0.5218394	-0.1052604	0	0.05784333	0.6647643	-0.13	00:00:03.0669734	0	0	0	0	0.01337069	0.2711679	-0.03667545	0	0.03132623	0.5355382	-0.08786249	0	0.042018	0.6770582	-0.12
00:00:03.1000109	0	0	0	0	0.01251054	0.2736791	-0.04020452	0	0.02738583	0.5426177	-0.08956003	0	0.03222464	0.685883	-0.11	00:00:03.1330475	0	0	0	0	0.01523411	0.2759488	-0.03692675	0	0.03250265	0.5475471	-0.0858233	0	0.03245127	0.6897444	-0.11
00:00:03.1669740	0	0	0	0	0.008019984	0.2791451	-0.03538179	0	0.01861978	0.5542871	-0.07989073	0	0.02428812	0.7011126	-0.10	00:00:03.1999927	0	0	0	0	0.008270144	0.2791837	-0.03805757	0	0.01903516	0.5542538	-0.08793235	0	0.02242839	0.6985454	-0.11
00:00:03.2331178	0	0	0	0	0.009511888	0.2799407	-0.04208088	0	0.02167374	0.5594996	-0.09443951	0	0.0195232	0.6975855	-0.12	00:00:03.299945	0	0	0	0	0.01154889	0.2832565	-0.04047036	0	0.02565873	0.5622752	-0.09182835	0	0.01052868	0.7024144	-0.12
00:00:03.3330401	0	0	0	0	0.01201385	0.2838714	-0.03656006	0	0.02656358	0.5634359	-0.08432531	0	0.01331329	0.7031119	-0.11	00:00:03.3670128	0	0	0	0	0.008796871	0.2855452	-0.03734994	0	0.02028328	0.5667589	-0.08545426	0	0.01474881	0.7086622	-0.11
00:00:03.4330394	0	0	0	0	0.004207253	0.2863187	-0.03451705	0	0.01123691	0.5681854	-0.08000469	0	0.001154184	0.7099783	-0.09	00:00:03.4669720	0	0	0	0	0.003078256	0.2867092	-0.02986169	0	0.01291656	0.568783	-0.07117057	0	0.004427731	0.7090999	-0.09
00:00:03.4999620	0	0	0	0	0.003739893	0.2930259	-0.03066038	0	0.01039225	0.5813159	-0.0723331	0	-0.002274275	0.7216754	-0.08																

Ilustración 204. Previsualizar CSV