



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA (SG)

**Grado en Ingeniería Informática de Servicios y
Aplicaciones**

Bailando en pareja

Alumno: Rubén Burgos Perela

Tutor: José Vicente Álvarez Bravo

Bailando en pareja

Rubén Burgos Perela

Resumen

La afición por los bailes de pareja cuenta con numerosos adeptos, aunque no todos los adeptos a los bailes de pareja bailen todos los bailes que hay en pareja. Algunos bailan un determinado baile como pueda ser el tango, mientras que otros pueden bailar varios como pueda ser la bachata, la salsa y la kizomba. Sin embargo, es muy común en todos sus adeptos, el viajar y descubrir cómo es la escena de los bailes de pareja en otros lugares enfrentándose al desconocimiento, no sabiendo si ciertos eventos o quedadas merecen la pena. Además, los nuevos y los que ya dominan algún baile, pueden estar interesados en conocer referencias de una determinada academia o incluso conocer una nueva pareja con la que poder practicar acrobacias.

La aplicación *Bailando en pareja* pretende ser la red social de la gente aficionada a los bailes de pareja aprovechando las necesidades de intercambio de información que surgen entre los propios aficionados a los bailes de pareja. Dicha aplicación está compuesta por dos proyectos: el proyecto *WebMVC* y el proyecto *ServiciosRest*. *WebMVC* es una aplicación web diseñada bajo el patrón MVC (Modelo-Vista-Controlador). *ServiciosRest* es un servicio que aporta una interfaz web apoyada en HTTP permitiendo a la aplicación web las cuatro operaciones básicas de consulta, creación, edición y eliminación de información de los aficionados.

Palabras claves: red social, baile, aplicación web.

Índice general

Resumen	3
Índice general	5
Índice de figuras	9
Índice de tablas	13
1. Introducción	18
1.1. Motivación	19
1.2. Objetivos	20
1.3. Descripción del proyecto	21
1.4. Estructura del documento	24
2. Estado del arte	25
2.1. Estudio de las herramientas	25
2.2. Conclusiones de las herramientas	29
3. Planificación	31
3.1. Metodología de desarrollo	31
3.2. Detalle de planificación	32
3.3. Estimaciones temporales	34
3.3.1. Estimación por puntos de función	35
3.3.2. Estimación de esfuerzo por COCOMO	39
3.4. Presupuesto	41
3.4.1. Presupuesto software	41
3.4.2. Presupuesto hardware	41
3.4.3. Presupuesto en recursos humanos	42
3.4.4. Presupuesto total del proyecto	42
3.5. Desarrollo y esfuerzo real de la aplicación	43
4. Análisis	44
4.1. Actores	44
4.2. Requisitos de usuario	45
4.2.1. Listado de casos de uso	45
4.2.2. Especificación de casos de uso	48
4.3. Reglas de negocio	51
4.4. Requisitos funcionales	51
4.5. Requisitos no funcionales	53
4.5.1. Atributos de calidad	53

4.5.2. Restricciones de implementación y arquitectura	54
4.5.3. Requisitos de interfaces externas	55
4.6. Esquema conceptual de datos.....	55
4.6.1. Modelo entidad-relación	55
4.6.2. Diccionario de datos.....	57
5. Diseño del sistema.....	69
5.1. Arquitectura lógica	69
5.2. Arquitectura física	74
5.3. Modelo lógico de datos	74
5.4. Diagramas de clases	76
5.5. Diagramas de secuencia	110
5.6. Diseño de interfaces	123
6. Implementación de la aplicación.....	135
6.1. Implementación de WebMVC y ServiciosREST	135
6.2. Implementación de la bases de datos	135
6.3. Principales tecnologías usadas	136
7.4. Herramientas utilizadas	138
7. Pruebas	140
7.1. Pruebas de iteración I	140
7.2. Pruebas de iteración II.....	141
7.3. Pruebas de iteración III	144
7.4. Pruebas de iteración IV	148
8. Manuales	151
8.1. Manual de instalación y despliegue.....	151
8.1.1 Preparación del servidor.....	151
8.1.2 Configuración del puerto.....	152
8.1.3 Despliegue de la aplicación.....	152
8.2. Manual de usuario	152
8.2.1 Manual de usuario no registrado	153
8.2.2 Manual de usuario registrado	153
8.2.3 Manual de usuario general	154
8.2.4 Manual de usuario administrador.....	169
9. Conclusiones	172
9.1. Conocimientos adquiridos.....	172
9.2. Líneas de trabajo para el futuro.....	173
Referencias.....	175

Anexo I: Glosario.....	177
Anexo II: Casos de uso	178
Anexo III: Requisitos funcionales.....	194
Anexo IV: Contenido del repositorio	205

Índice de figuras

Figura 1. Diagrama de Ishikawa de la aplicación de <i>Bailando en pareja</i>	22
Figura 2. Diagrama de incrementos y etapas.	32
Figura 3. Gantt correspondiente a la primera iteración.	33
Figura 4. Gantt correspondiente a la segunda iteración.	33
Figura 5. Gantt correspondiente a la tercera iteración.....	34
Figura 6. Gantt correspondiente a la cuarta iteración.....	34
Figura 7. Jerarquía de los actores del sistema.	45
Figura 8. Ejemplos de asociaciones de casos de uso.....	47
Figura 9. Diagrama de casos de uso.	48
Figura 10. Diagrama entidad-relación.....	56
Figura 11. Arquitectura lógica <i>WebMVC</i>	72
Figura 12. Arquitectura lógica <i>ServiciosREST</i>	73
Figura 13. Arquitectura física.	74
Figura 14. Modelo relacional.	75
Figura 15. Clases de los paquetes <i>webmvc.config</i> y <i>webmvc.formatter</i>	76
Figura 16. Clases del paquete <i>webmvc.email</i>	76
Figura 17. Clases del paquete <i>webmvc.controller 1</i>	77
Figura 18. Clases del paquete <i>webmvc.controller 2</i>	77
Figura 19. Clases del paquete <i>webmvc.controller 3</i>	78
Figura 20. Clases del paquete <i>webmvc.controller.bean 1</i>	79
Figura 21. Clases del paquete <i>webmvc.controller.bean 2</i>	79
Figura 22. Clases del paquete <i>webmvc.controller.bean 3</i>	80
Figura 23. Clases del paquete <i>webmvc.controller.bean 4</i>	80
Figura 24. Clases del paquete <i>webmvc.service 1</i>	81
Figura 25. Clases del paquete <i>webmvc.service 2</i>	81
Figura 26. Clases del paquete <i>webmvc.service.impl 1</i>	82
Figura 27. Clases del paquete <i>webmvc.service.impl 2</i>	82
Figura 28. Clases del paquete <i>webmvc.clientes 1</i>	83
Figura 29. Clases del paquete <i>webmvc.clientes 2</i>	84
Figura 30. Clases del paquete <i>webmvc.converter</i>	84
Figura 31. Clases del paquete <i>webmvc.exceptions</i>	84
Figura 32. Clases del paquete <i>webmvc.clientes.formatters</i>	85
Figura 33. Clases del paquete <i>webmvc.clientes.dto 1</i>	86

Figura 34. Clases del paquete webmvc.clientes.dto 2.....	86
Figura 35. Clases del paquete webmvc.clientes.dto 3.....	87
Figura 36. Clases del paquete webmvc.clientes.dto 4.....	87
Figura 37. Clases del paquete webmvc.clientes.dto 5.....	88
Figura 38. Clases del paquete webmvc.clientes.dto 6.....	88
Figura 39. Clases del paquete webmvc.clientes.dto 7.....	89
Figura 40. Clases del paquete webmvc.clientes.dto 8.....	89
Figura 41. Clases del paquete webmvc.clientes.dto 9.....	90
Figura 42. Clases del paquete webmvc.clientes.dto 10.....	90
Figura 43. Clases del paquete webmvc.clientes.dto 11.....	91
Figura 44. Clases del paquete webmvc.clientes.dto 12.....	91
Figura 45. Clases del paquete webmvc.clientes.dto 13.....	92
Figura 46. Clases del paquete serviciosrest.controller 1.....	93
Figura 47. Clases del paquete serviciosrest.controller 2.....	93
Figura 48. Clases del paquete serviciosrest.controller 3.....	94
Figura 49. Clases del paquete serviciosrest.controller.bean 1.....	95
Figura 50. Clases del paquete serviciosrest.controller.bean 2.....	95
Figura 51. Clases del paquete serviciosrest.controller.bean 3.....	96
Figura 52. Clases del paquete serviciosrest.controller.dto 1.....	96
Figura 53. Clases del paquete serviciosrest.controller.dto 2.....	97
Figura 54. Clases del paquete serviciosrest.controller.dto 3.....	97
Figura 55. Clases del paquete serviciosrest.controller.dto 4.....	98
Figura 56. Clases del paquete serviciosrest.controller.dto 5.....	98
Figura 57. Clases del paquete serviciosrest.controller.dto 6.....	99
Figura 58. Clases del paquete serviciosrest.controller.dto 7.....	99
Figura 59. Clases del paquete serviciosrest.controller.dto 8.....	100
Figura 60. Clases del paquete serviciosrest.controller.dto 9.....	100
Figura 61. Clases del paquete serviciosrest.controller.dto 10.....	101
Figura 62. Clases del paquete serviciosrest.controller.dto 11.....	102
Figura 63. Clases del paquete serviciosrest.controller.formatter.....	102
Figura 64. Clases del paquete service 1.....	103
Figura 65. Clases del paquete service 2.....	104
Figura 66. Clases del paquete service.converter 1.....	104
Figura 67. Clases del paquete service.converter 2.....	105
Figura 68. Clases del paquete service.impl 1.....	105
Figura 69. Clases del paquete service.impl 2.....	106

Figura 70. Clases del paquete service.impl 3.	106
Figura 71. Clases del paquete service.impl 4.	107
Figura 72. Clases del paquete entity 1.....	107
Figura 73. Clases del paquete entity 2.....	108
Figura 74. Clases del paquete entity 3.....	108
Figura 75. Clases del paquete entity 4.....	109
Figura 76. Clases del paquete entity 5.....	109
Figura 77. Clases del paquete entity.repository.	110
Figura 78. Diagrama de secuencia registrarse.....	111
Figura 79. Diagrama de secuencia login.	111
Figura 80. Diagrama de secuencia recuperar contraseña.	112
Figura 81. Diagrama de secuencia logout.	112
Figura 82. Diagrama de secuencia ver perfil detallado.	113
Figura 83. Diagrama de secuencia de consultar configuración propia.....	113
Figura 84. Diagrama de secuencia de gestionar la configuración.	114
Figura 85. Diagrama de secuencia de cambiar foto de perfil.	114
Figura 86. Diagrama de secuencia de editar perfil.	115
Figura 87. Diagrama de secuencia de eliminar perfil.....	115
Figura 88. Diagrama de secuencia de cambiar contraseña.....	116
Figura 89. Diagrama de secuencia de crear publicación.	116
Figura 90. Diagrama de secuencia de editar publicación.	117
Figura 91. Diagrama de secuencia de dar me gusta.	117
Figura 92. Diagrama de secuencia de buscar perfiles.	118
Figura 93. Diagrama de secuencia de gestionar la inclusión como favorito.	118
Figura 94. Diagrama de secuencia de buscar publicaciones.	119
Figura 95. Diagrama de secuencia de reportar.	119
Figura 96. Diagrama de secuencia de acceder al chat.	120
Figura 97. Diagrama de secuencia de escribir mensaje privado al perfil.	120
Figura 98. Diagrama de secuencia de buscar perfil reportado.	121
Figura 99. Diagrama de secuencia de buscar publicación reportada.....	121
Figura 100. Diagrama de secuencia de visualizar reportes.	122
Figura 101. Diagrama de secuencia de gestionar la activación del perfil.	122
Figura 102. Diagrama de secuencia de gestionar la activación de la publicación.....	123
Figura 103. Pantalla registrarme.	153
Figura 104. Pantalla iniciar sesión.	154
Figura 105. Pantalla home.....	155

Figura 106. Pantalla aficionados.....	156
Figura 107. Pantalla publicaciones.....	157
Figura 108. Pantalla mensajes.....	158
Figura 109. Pantalla con el menú desplegable.....	158
Figura 110. Pantalla eliminar contraseña.....	159
Figura 111. Pantalla cambio de contraseña.....	160
Figura 112. Pantalla configuración.....	161
Figura 113. Pantalla mi perfil - info.....	161
Figura 114. Modal crear y modificar la publicación.....	162
Figura 115. Modal modificar perfil.....	163
Figura 116. Pantalla mi perfil - publicaciones.....	164
Figura 117. Pantalla mi perfil - favoritos.....	165
Figura 118. Modal enviar un mensaje.....	167
Figura 119. Modal reportar.....	168
Figura 120. Modal ver reportes.....	169
Figura 121. Pantalla buscar perfiles reportados.....	170
Figura 122. Pantalla buscar publicaciones reportadas.....	171

Índice de tablas

Tabla 1. Principales diferencias entre las diferentes herramientas.	30
Tabla 2. Detalle de la planificación de las iteraciones y las etapas.	33
Tabla 3. Complejidad.	35
Tabla 4. Puntos de función sin ajustar.	37
Tabla 5. Factores de ajuste y su valoración.	38
Tabla 6. Equivalencias entre LDC y PF.	38
Tabla 7. Estimaciones finales de líneas de código.	39
Tabla 8. Coeficientes COCOMO.	40
Tabla 9. Factores de ajuste COCOMO.	40
Tabla 10. Presupuesto software.	41
Tabla 11. Presupuesto hardware.	42
Tabla 12. Presupuesto estimado para el proyecto.	43
Tabla 13. Listado de actores del sistema.	44
Tabla 14. Casos de uso encuadrados por actores.	46
Tabla 15. Caso de uso UNR1.	49
Tabla 16. Caso de uso UR1.	49
Tabla 17. Caso de uso UR2.	50
Tabla 18. Caso de uso UR3.	50
Tabla 19. Reglas de negocio.	51
Tabla 20. Requisitos funcionales en el caso de uso UNR1.	52
Tabla 21. Requisitos funcionales en el caso de uso UR1.	52
Tabla 22. Requisitos funcionales en el caso de uso UR2.	53
Tabla 23. Requisitos funcionales en el caso de uso UR3.	53
Tabla 24. Atributos de calidad para la disponibilidad.	53
Tabla 25. Atributos de calidad para la escalabilidad.	54
Tabla 26. Atributos de calidad para el rendimiento.	54
Tabla 27. Atributos de calidad para la seguridad.	54
Tabla 28. Atributos de calidad para la usabilidad.	54
Tabla 29. Restricciones de arquitectura e implementación.	55
Tabla 30. Requisitos de interfaces externas.	55
Tabla 31. Tabla ‘usuario’ con sus atributos.	57
Tabla 32. Tabla ‘rol_app’ con sus atributos.	58
Tabla 33. Tabla ‘detalle_usuario’ con sus atributos.	59

Tabla 34. Tabla ‘configuracion’ con sus atributos.....	59
Tabla 35. Tabla ‘reporte’ con sus atributos.....	60
Tabla 36. Tabla ‘motivo_reporte’ con sus atributos.	60
Tabla 37. Tabla ‘baile’ con sus atributos.	61
Tabla 38. Tabla ‘ciudad’ con sus atributos.	61
Tabla 39. Tabla ‘publicacion’ con sus atributos.	62
Tabla 40. Tabla mensaje con sus atributos.	63
Tabla 41. Tabla ‘usuario_rol’ con sus atributos.....	63
Tabla 42. Tabla ‘configuracion_usuario’ con sus atributos.	64
Tabla 43. Tabla ‘relacionar’ con sus atributos.....	65
Tabla 44. Tabla ‘detalleusuario_baile’ con sus atributos.	66
Tabla 45. Tabla ‘publicacion_baile’ con sus atributos.....	67
Tabla 46. Tabla ‘likes’ con sus atributos.....	68
Tabla 47. Esbozo interfaz de registro.	124
Tabla 48. Esbozo interfaz de login.	124
Tabla 49. Esbozo interfaz de recuperación de contraseña por email.....	125
Tabla 50. Esbozo interfaz de home.	125
Tabla 51. Esbozo interfaz de info. de mi perfil.	126
Tabla 52. Esbozo interfaz de publicaciones de mi perfil.....	126
Tabla 53. Esbozo interfaz de añadidos a favoritos mi perfil.	127
Tabla 54. Esbozo interfaz configuración.....	127
Tabla 55. Esbozo interfaz de cambio de contraseña.....	128
Tabla 56. Esbozo interfaz de eliminación del perfil.....	128
Tabla 57. Esbozo interfaz de búsqueda de perfiles.....	129
Tabla 58. Esbozo interfaz de búsqueda de publicaciones.....	129
Tabla 59. Esbozo interfaz de mensajes.....	130
Tabla 60. Esbozo interfaz de info. de perfil ajeno.....	130
Tabla 61. Esbozo interfaz de publicaciones de perfil ajeno.	131
Tabla 62. Esbozo interfaz de creación y edición de publicación.....	131
Tabla 63. Esbozo interfaz de edición de perfil.	132
Tabla 64. Esbozo interfaz de envío de mensaje.....	132
Tabla 65. Esbozo interfaz de creación de reporte.....	133
Tabla 66. Esbozo interfaz de búsqueda de aficionados para administradores.....	133
Tabla 67. Esbozo interfaz de búsqueda de publicaciones para administradores.	134
Tabla 68. Esbozo interfaz de visualización de reportes para administradores.	134
Tabla 69. Prueba P1.1.....	140

Tabla 70. Prueba P1.2.....	140
Tabla 71. Prueba P1.3.....	140
Tabla 72. Prueba P1.4.....	141
Tabla 73. Prueba P1.5.....	141
Tabla 74. Prueba P1.5.1.....	141
Tabla 75. Prueba P2.1.....	141
Tabla 76. Prueba P2.2.....	142
Tabla 77. Prueba P2.3.....	142
Tabla 78. Prueba P2.4.....	142
Tabla 79. Prueba P2.5.....	142
Tabla 80. Prueba P2.4.6.....	143
Tabla 81. Prueba P2.7.....	143
Tabla 82. Prueba P2.8.....	143
Tabla 83. Prueba P2.9.....	143
Tabla 84. Prueba P3.1.....	144
Tabla 85. Prueba P3.2.....	144
Tabla 86. Prueba P3.3.....	144
Tabla 87. Prueba P3.3.1.....	145
Tabla 88. Prueba P3.4.....	145
Tabla 89. Prueba P3.5.....	145
Tabla 90. Prueba P3.6.....	145
Tabla 91. Prueba P3.7.....	146
Tabla 92. Prueba P3.8.....	146
Tabla 93. Prueba P3.9.....	146
Tabla 94. Prueba P3.9.1.....	146
Tabla 95. Prueba P3.10.....	147
Tabla 96. Prueba P3.10.....	147
Tabla 97. Prueba P3.10.1.....	147
Tabla 98. Prueba P4.1.....	148
Tabla 99. Prueba P4.2.....	148
Tabla 100. Prueba P4.3.....	148
Tabla 101. Prueba P4.4.....	149
Tabla 102. Prueba P4.5.....	149
Tabla 103. Prueba P4.6.....	149
Tabla 104. Prueba P4.7.....	149
Tabla 105. Prueba P4.8.....	150

Tabla 106. Prueba P4.9.....	150
Tabla 107. Caso de uso UG1.....	178
Tabla 108. Caso de uso UG2.....	178
Tabla 109. Caso de uso UG3.....	179
Tabla 110. Caso de uso UG4.....	179
Tabla 111. Caso de uso UG5.....	180
Tabla 112. Caso de uso UG6.....	180
Tabla 113. Caso de uso UG7.....	181
Tabla 114. Caso de uso UG8.....	181
Tabla 115. Caso de uso UG9.....	182
Tabla 116. Caso de uso UG10.....	182
Tabla 117. Caso de uso UG11.....	183
Tabla 118. Caso de uso UG12.....	184
Tabla 119. Caso de uso UG13.....	184
Tabla 120. Caso de uso UG14.....	184
Tabla 121. Caso de uso UG15.....	185
Tabla 122. Caso de uso UG16.....	185
Tabla 123. Caso de uso UG17.....	186
Tabla 124. Caso de uso UG18.....	186
Tabla 125. Caso de uso UG19.....	187
Tabla 126. Caso de uso UG20.....	187
Tabla 127. Caso de uso UG21.....	188
Tabla 128. Caso de uso UG22.....	188
Tabla 129. Caso de uso UG23.....	189
Tabla 130. Caso de uso UG24.....	189
Tabla 131. Caso de uso UG25.....	190
Tabla 132. Caso de uso UG26.....	190
Tabla 133. Caso de uso UA1.....	191
Tabla 134. Caso de uso UA2.....	191
Tabla 135. Caso de uso UA3.....	192
Tabla 136. Caso de uso UA4.....	192
Tabla 137. Caso de uso UA5.....	193
Tabla 138. Caso de uso UA6.....	193
Tabla 139. Requisitos funcionales en el caso de uso UG1.....	194
Tabla 140. Requisitos funcionales en el caso de uso UG2.....	194
Tabla 141. Requisitos funcionales en el caso de uso UG3.....	194

Tabla 142. Requisitos funcionales en el caso de uso UG4.	195
Tabla 143. Requisitos funcionales en el caso de uso UG5.	195
Tabla 144. Requisitos funcionales en el caso de uso UG6.	195
Tabla 145. Requisitos funcionales en el caso de uso UG7.	196
Tabla 146. Requisitos funcionales en el caso de uso UG8.	196
Tabla 147. Requisitos funcionales en el caso de uso UG9.	197
Tabla 148. Requisitos funcionales en el caso de uso UG10.	197
Tabla 149. Requisitos funcionales en el caso de uso UG11.	197
Tabla 150. Requisitos funcionales en el caso de uso UG12.	198
Tabla 151. Requisitos funcionales en el caso de uso UG13.	198
Tabla 152. Requisitos funcionales en el caso de uso UG14.	198
Tabla 153. Requisitos funcionales en el caso de uso UG15.	199
Tabla 154. Requisitos funcionales en el caso de uso UG16.	199
Tabla 155. Requisitos funcionales en el caso de uso UG17.	199
Tabla 156. Requisitos funcionales en el caso de uso UG18.	200
Tabla 157. Requisitos funcionales en el caso de uso UG19.	200
Tabla 158. Requisitos funcionales en el caso de uso UG20.	200
Tabla 159. Requisitos funcionales en el caso de uso UG21.	201
Tabla 160. Requisitos funcionales en el caso de uso UG22.	201
Tabla 161. Requisitos funcionales en el caso de uso UG23.	201
Tabla 162. Requisitos funcionales en el caso de uso UG24.	202
Tabla 163. Requisitos funcionales en el caso de uso UG25.	202
Tabla 164. Requisitos funcionales en el caso de uso UG26.	202
Tabla 165. Requisitos funcionales en el caso de uso UA1.	203
Tabla 166. Requisitos funcionales en el caso de uso UA2.	203
Tabla 167. Requisitos funcionales en el caso de uso URN1.	203
Tabla 168. Requisitos funcionales en el caso de uso UA4.	204
Tabla 169. Requisitos funcionales en el caso de uso UA5.	204
Tabla 170. Requisitos funcionales en el caso de uso UA6.	204
Tabla 171. Contenido del repositorio.	205

1. Introducción

Los bailes de pareja abarcan diferentes bailes, los cuales uno puede practicarlos por ser una buena afición, por socializar, hacer ejercicio o incluso practicarlos por tener la ambición de bailar participando en el ámbito competitivo.

En todos ellos hay que seguir el ritmo de la música, siguiendo una serie de pasos previamente establecidos, propios y característicos de ese baile en concreto, junto con la pareja de baile y en armonía. Para ello, es necesario una buena coordinación acompañada de una técnica para así lograr un movimiento hermoso, que incluso pueda transmitir la impresión de que la pareja se fusiona moviéndose de manera simple, elegante y sin esfuerzo alguno. Usualmente, por tradición y también por la corporalidad característica del hombre, este suele tomar el rol dirigente. En cambio, la mujer por los mismos motivos solo que de manera complementaria, suele tomar el rol seguidor. No obstante, no siempre es así, se pueden cambiar los roles debido al interés por mejorar y perfeccionar un determinado baile, o simple y sencillamente por variar y/o por diversión.

El hecho de bailar junto a otra persona no solo es una actividad social, sino que también da lugar a la expresión artística con múltiples beneficios físicos y psicológicos para los participantes. Teniendo la capacidad de poderse convertir en una afición muy saludable para los participantes, además de desarrollarse en pareja. Algunos de los posibles beneficios de los cuales los participantes se pueden beneficiar son: la diversión, la alegría, el ejercicio físico, el trabajo de la psicomotricidad, el conocimiento de otras personas con la misma afición, una evasión de la rutina del día a día y de los problemas, así como comprender que a base de errores se aprende siendo esto parte importante del aprendizaje, reforzando así el tener confianza en uno mismo.

Es de relevancia tener presente, que a pesar de que los bailes de pareja se bailen en pareja, como bien indica su nombre, uno puede por el motivo o motivos que sea, no tener pareja. Sin embargo, no debería ser un motivo para renunciar a ellos. Tan solo sería necesario encontrar a alguien para la ocasión, y que esté dispuesto a animarse. De hecho, en los cursos de iniciación se suele fomentar el encontrar pareja en el mismo curso o incluso algunas escuelas permiten dejar sus tarjetas de presentación y contacto para que quienes estén empezando, les pueda ser más fácil encontrar una pareja afín.

Ante el abanico de bailes bailados en pareja, con sus correspondientes particularidades como puedan ser la música, el ritmo, el estilo y demás. Estas pueden generar a uno, una mayor o menor afinidad. Por lo que sería una muy buena idea investigar previamente los diferentes bailes por cualquier medio, ya sea a través de vídeos, eventos u otra forma, para así tratar determinar por qué baile uno debería decantarse y así poder buscar dónde y cómo aprender.

Algunos de los bailes por los que uno podría encontrar el atractivo y querer bailar en pareja son los siguientes: tango, bachata, salsa, kizomba, swing, zouk, entre otros.

1.1. Motivación

La motivación de la aplicación *Bailando en pareja* proviene de la intención de crear una red social, que ofrezca la posibilidad de poner en contacto a los aficionados de los bailes para el intercambio de información e impresiones entre ellos dentro del territorio nacional, especialmente cuando no se conocen previamente.

La aplicación pretende ser muy útil, en particular para aquellos que quieren ir por primera vez a clases, ofreciendo la oportunidad de tener un perfil y una descripción, publicar preguntando algo, posibilitando así conocer a alguien afín con quien ir a clase, cuando nunca se ha ido, posibilitando además el poder preguntar por la opinión de las academias cercanas que imparten clases. Bien es sabido que los comienzos no suelen ser fáciles. Pues el aventurarse a ir solo, puede ser una manera poco óptima de empezar, por la posibilidad de encontrarse con una proporción de hombres y mujeres poco equilibrada, pudiéndose dar la situación de haber más aficionados interesados en seguir que en dirigir. Y eso, sin olvidar que lo suyo sería dar con una pareja que esté dispuesta a practicar los pasos aprendidos después, para así fomentar el aprendizaje y combatir el olvido.

A veces, se da el caso de que algunos apasionados de los bailes de pareja quieren y están dispuestos a practicar acrobacias. Esta práctica, es poco probable que todos estén dispuestos practicar, tanto por la exigencia física, como por el riesgo que puede suponer para la integridad física. Por consiguiente, a la mayoría les puede resultar muy difícil dar con alguien con tal interés. Para ello, pueden ser muy convenientes las quedadas entre interesados. Quedadas que durante los tiempos del COVID florecieron ante las restricciones impuestas por motivos sanitarios, puesto que no era muy legal y seguro bailar en sitios cerrados, lo cual llevó a muchos a realizar quedadas en los parques al aire libre para bailar con menos riesgos. Práctica que también es muy habitual en los días de verano, e implica poner a mucha gente en contacto.

También se da la situación en la que aficionados aprovechan desplazamientos por otros motivos como puedan ser laborales, visitas turísticas a otras ciudades, o incluso muchos tienen el interés de conocer las escenas del baile de otros lugares en los que también se baila ese baile con el que tanto disfrutan. O bien, en otras ocasiones podría ser que un aficionado tenga intención de ir a un determinado evento, del cual la única referencia al respecto es tan solo un flyer publicitario visualizado en alguna parte. Dicho flyer trata animar y promocionar el evento mostrando sus bondades y puntos fuertes. Sin embargo, tener referencias podría evitar encontrarse con un evento no muy allá organizado, con pocos aficionados, un nivel de baile por debajo del esperado, etc. Entonces, brindar la posibilidad de preguntar a otras personas aficionadas por opiniones puede ser de gran utilidad a la hora de decidir, si merece la pena ir, evitando el aventurarse con el consiguiente riesgo de insatisfacción.

Concluyendo, ya sea por encontrarse uno con interés de empezar, por saber a qué academia acudir, querer practicar acrobacias, ir a quedadas, conocer otras escenas, determinar si un evento merece la pena, hay una demanda de comunicación y de intercambio de comunicación entre los aficionados que *Bailando en pareja* pretende cubrir.

1.2. Objetivos

El **objetivo general** de la aplicación *Bailando en pareja* es:

- Posibilitar la comunicación e intercambio de información acerca de los bailes de pareja entre sus aficionados.

Para alcanzar el objetivo general anterior se presentan los **objetivos específicos** siguientes:

- Permitir el registro de nuevos aficionados, así como el inicio de sesión y fin de sesión de los ya registrados.
- Crear un perfil tras el registro y dar la posibilidad de editar el perfil con características personales como son la foto de perfil, la ciudad en la que se reside, los bailes por los que se muestra interés, el género que se tiene, el rol que se está dispuesto a tomar y una descripción de intereses personales, pese a que el email y la edad de registro no se puedan cambiar. Además de posibilitar la eliminación del perfil priorizando la seguridad.
- Proporcionar a cada perfil la posibilidad de cambiar la configuración de las notificaciones de la recepción de mensajes privados y de las nuevas publicaciones de los aficionados añadidos a favoritos, además de la posibilidad de cambiar la configuración de la privacidad para poder ocultar el email y la edad.
- Posibilitar a todos los perfiles la creación de publicaciones, la edición y la eliminación de sus publicaciones.
- Ofrecer a los usuarios generales la búsqueda de perfiles de otros aficionados en base a la ciudad, los bailes de interés, el género y el rol.
- Ofrecer a los usuarios generales la búsqueda de publicaciones de otros aficionados en base a la ciudad, los bailes de interés, la fechas y el tipo de evento.
- Posibilitar la visualización de los perfiles, así como la posibilidad de ser añadidos y quitados de ser favoritos, la posibilidad de ser escritos por privado y también ser reportados.
- Posibilitar la visualización de las publicaciones, así como la posibilidad de escribir a su autor por privado, recibir me gusta, y también ser reportadas.
- Ofrecer un chat para la visualización de las conversaciones privadas del perfil con otros perfiles, viabilizando tanto leer los mensajes como responder.
- Aportar una búsqueda de perfiles para los usuarios administradores en base al email o nombre, la inactividad, y los motivos del reporte.

- Aportar una búsqueda de publicaciones para los usuarios administradores en base al email o nombre de autor o descripción del evento, la inactividad, y los motivos del reporte.
- Aportar a los usuarios administradores la posibilidad de ver los reportes de los perfiles y de las publicaciones, además de la posibilidad de inactivar tanto publicaciones como perfiles con sus publicaciones.

1.3. Descripción del proyecto

La aplicación *Bailando en pareja*, pretende ser la red social de los aficionados al baile, en concreto de aquellos apasionados por los bailes de pareja, pretendiendo ser un complemento a las relaciones sociales que se puedan dar entre los entusiastas de los bailes de pareja.

La aplicación tras el registro crea un perfil a sus usuarios, permite la edición de los perfiles creados, configurar pequeños ajustes en las notificaciones y la privacidad. Los usuarios pueden crear publicaciones públicas, que dan la posibilidad recibir me gusta, dando también la posibilidad de ver el perfil del autor y de escribir al autor por privado. Los mensajes privados se pueden visualizar en el chat, que ofrece la posibilidad de volver a escribir a los perfiles con los que previamente se ha intercambiado algún mensaje privado. En caso de que ciertos aficionados presenten un comportamiento inadecuado, las publicaciones pueden ser reportadas y los perfiles pueden ser reportados. Dichos reportes pueden ser visualizados por los usuarios administradores y de considerarlo necesario, pueden inactivar las publicaciones o incluso el perfil con todas sus publicaciones.

La descripción aportada sobre el proyecto, junto con los objetivos recogidos en el apartado previo, dan lugar a las siguientes causas con diferentes niveles, las cuales se muestran de manera resumida en el siguiente diagrama de Ishikawa que se adjunta seguidamente (ver figura 1).

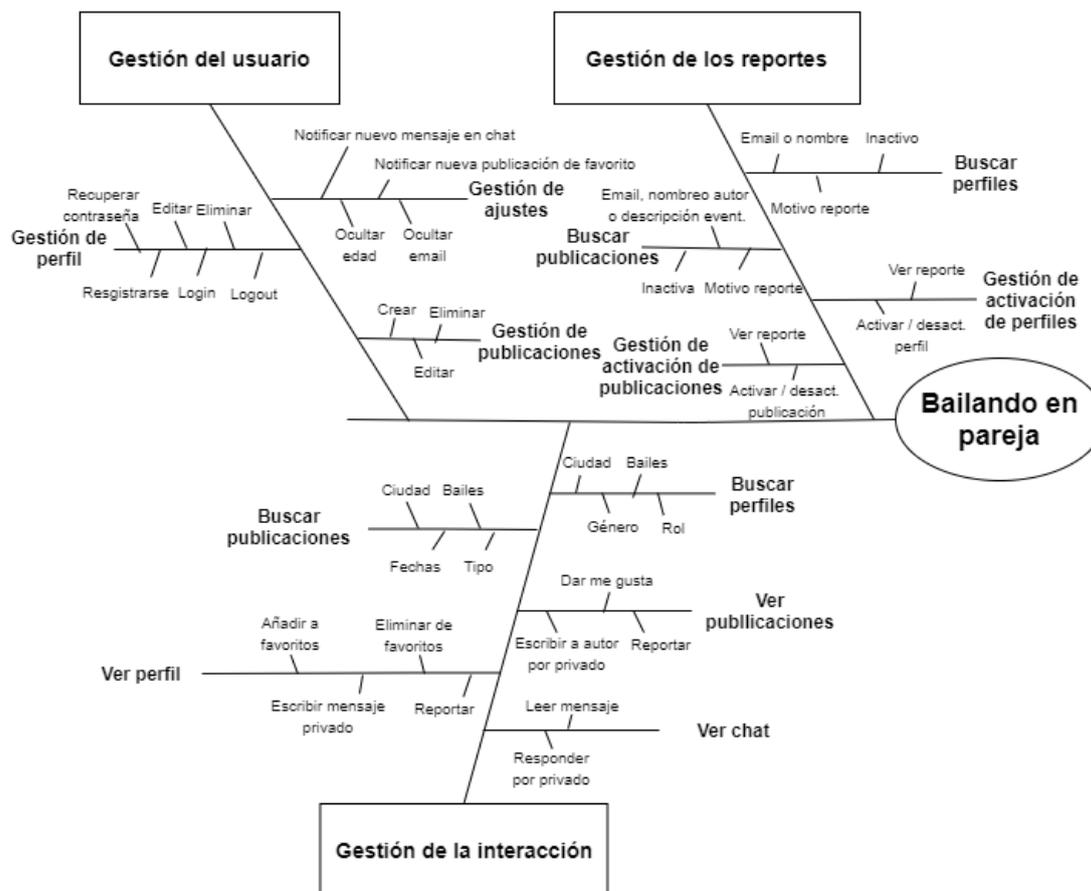


Figura 1. Diagrama de Ishikawa de la aplicación de *Bailando en pareja*.

Las diferentes causas se encuentran organizadas en tres niveles. En el primer nivel se encuentran las tres causas principales, de las cuales derivan las causas de segundo nivel. Y a su vez, de las causas de segundo nivel derivan las causas de tercer nivel.

- **Gestión del usuario:**

- **Gestión de perfil:** ofrece las operaciones esenciales para la gestión del propio perfil de un usuario como son: registrarse, hacer login y logout, editar el perfil con características determinantes de cara a ser encontrados por otros aficionados, y también la posibilidad de eliminar el propio perfil.
- **Gestión de ajustes:** posibilita la configuración de la privacidad permitiendo ocultar la edad y el email de registro en la aplicación; también posibilita configurar las notificaciones que llegan por email si se recibe un nuevo mensaje por el chat privado y la notificación de publicaciones de los aficionados añadidos previamente como favoritos.
- **Gestión de publicaciones:** permite la creación publicaciones con características relevantes para ser descubiertas por otros aficionados, y eliminación, las cuales también se pueden editar.

- **Gestión de la interacción:**
 - **Buscar perfil:** ofrece la búsqueda de otros aficionados a los bailes de pareja gracias a las características de los perfiles y configuración de los siguientes los filtros de: ciudad, bailes, género, y rol.
 - **Buscar publicación:** ofrece la búsqueda de publicaciones de otros aficionados debido a las características de las publicaciones y a los ajustes de los siguientes filtros en la búsqueda: ciudad, bailes, fechas y tipo.
 - **Ver perfil:** posibilita la visualización más detallada del perfil, que este pueda ser añadido y quitado de la lista de favoritos e incluso ser escrito por y reportado por los perfiles de otros aficionados.
 - **Ver publicación:** posibilita dar con perfiles de aficionados que tras la visualización se decida a escribir al perfil creador de la publicación, pudiendo las publicaciones recibir me gusta e incluso ser reportadas por otros aficionados.
 - **Ver chat:** aporta la visualización de mensajes intercambiados previamente con otros perfiles, viabilizando volver a escribir a los perfiles con los que previamente ya se ha enviado o recibido algún mensaje.
- **Gestión de los reportes:**
 - **Buscar perfiles:** permite a los usuarios administradores exclusivamente una búsqueda más apropiada para sus funciones como es buscar perfiles por: el email, el nombre del perfil o una parte de ambos, la inactivación y/o el motivo del reporte del perfil.
 - **Gestión de activación de perfiles:** permite exclusivamente a los usuarios administradores ver los detalles de los reportes asociados a los perfiles, así como la inactivación del perfil con sus correspondientes publicaciones.
 - **Buscar publicaciones:** permite exclusivamente a los usuarios administradores una búsqueda más adecuada para el desarrollo de sus funciones buscando por: el email, el nombre del perfil, la descripción o una parte de estos, la inactivación y/o el motivo del reporte de la publicación.
 - **Gestión de activación de publicaciones:** permite exclusivamente a los usuarios administradores ver los detalles de los reportes asociados a las publicaciones, además de la inactivación de las publicaciones.

1.4. Estructura del documento

La estructura de la memoria está compuesta por los siguientes capítulos posteriormente descritos:

- **Capítulo 1: Introducción.** Presentación de la temática, las motivaciones, los objetivos, y la descripción de las características más importantes del proyecto, y también la estructura de su memoria.
- **Capítulo 2: Estado del arte.** Recopilación y análisis de las tecnologías y productos existentes en la actualidad, que pueden ofrecer posibilidades similares a la desarrollada.
- **Capítulo 3: Planificación del proyecto.** Explicación de cómo se ha llevado a cabo la metodología, la planificación, además de abordarse la previsión de tiempo y costes.
- **Capítulo 4: Análisis.** Especificación de los actores, funcionalidades del sistema mediante casos de uso, además de un análisis de la base de datos.
- **Capítulo 5: Diseño del sistema.** Descripción de la arquitectura lógica y los componentes usados para hacer posible las funcionalidades especificadas en el análisis.
- **Capítulo 6: Implementación.** Aborda las tecnologías, las herramientas, así como las dificultades encontradas en el desarrollo del proyecto.
- **Capítulo 7: Pruebas.** Recopila las pruebas más relevantes a para probar si el sistema funciona correctamente.
- **Capítulo 8: Manuales.** Recoge información relevante para el despliegue del proyecto y también para su uso.
- **Capítulo 9: Conclusiones.** Aporta las reflexiones tras haber desarrollado el proyecto, además de las posibles líneas de trabajo detectadas de cara al futuro.
- **Bibliografía:** Libros, documentos y sitios web consultados durante el desarrollo del proyecto.
- **Anexos:**
 - Anexo I: Glosario.
 - Anexo II: Casos de uso
 - Anexo III: Requisitos funcionales
 - Anexo IV: Contenido del repositorio.

2. Estado del arte

El estado del arte consiste en efectuar un estudio de las tecnologías y productos existentes, para exponer la actualidad sobre la temática del proyecto desarrollado. Por ello, se van a analizar las aplicaciones usadas por los aficionados a los bailes de pareja.

2.1. Estudio de las herramientas

El estudio de las herramientas se va a realizar a continuación teniendo en cuenta la información encontrada en las interfaces de las herramientas la experiencia tras el registro e inicio de sesión como usuario (siendo la última fecha de consulta de 09.06.23), así como reflexiones y comentarios que han sido intercambiados con aficionados (anteriores al 09.06.23). Sin embargo, en ningún momento se ha accedido a alguna herramienta como profesional, ni se han intercambiado opiniones y experiencias con profesionales y, por lo tanto, no están reflejadas en este estudio.

La salsa del baile

La salsa del baile (<https://lasalsadelbaile.com/>) es una web de congresos de baile, venta de entradas, inscripciones online, control de acceso y ticketera de eventos. Los bailes ofrecidos por la web son únicamente salsa, bachata y kizomba, donde el cliente paga una comisión sobre el precio de la entrada. Dicha plataforma dice haber sido desarrollada por programadores aficionados al baile, siendo una web a medida y en constante desarrollo.

Las funcionalidades ofrecidas por la web, previo alta en la misma, son la de crear y modificar eventos, con diferente acceso para socios, agencias o promotores. En cuanto a las entradas y las ofertas, las entradas se pueden crear ilimitadamente pudiendo tener diferentes precios en función de lo que ofrezca la entrada, como pueda ser el tener alojamiento incluido o no, además permite controlar la proporción de chicas y chicos, la proporción del aforo, los cambios de titularidad, o incluso añadir campos. Los promotores se les puede habilitar el acceso posibilitándoles que puedan añadir más entradas, así como la gestión de deuda de cada promotor controlando quien vende más en cualquier momento; asimismo se puede configurar de manera individual a cada promotor para limitar la venta y establecer tarifas personalizadas. Además, la plataforma genera estadísticas en las que se puede acceder a registros para controlar el evento, generando los informes sobre varios eventos, permitiendo compararle con otros años, además de observar que clientes repiten o que promotores mantienen sus ventas.

En definitiva, lo ofrecido por la plataforma web es información por parte de organizadores y/o promotores sobre sus eventos de salsa, bachata y/o kizomba posibilitando a los clientes la venta de entradas generando beneficios para la web por cada entrada vendida, generando con ello datos para el control y la obtención de conclusiones.

Salsero

Salsero (<https://www.salsero.es/bailas>) ofrece para salsa, bachata y kizomba una aplicación nativa para Android e IOS con funciones básicas, una web con mayor contenido que la aplicación, ofreciendo: compras, ayuda y blog con profesores y pinchadiscos. Además de lo anterior cuenta con expertos gestionando Facebook e Instagram que analizan las analíticas de usuario. También cuenta con un canal de YouTube dirigido por el departamento audiovisual, aparte de promociones como pueda ser el merchandising, regalos promocionales y colaboraciones.

Salsero aporta información sobre academias, artistas, ya sean profesores de baile o pinchadiscos, noticias salseras y artículos de salsa, lo que sucede los fines de semana, los eventos próximos, información rigurosa sobre las salas de baile con posibilidad de posicionamiento, servicios de comunicación en las redes sociales y mailing mediante pago, venta de entradas para los tres bailes (salsa, bachata y kizomba) con diferentes tarifas. Salsero es un proyecto de carácter global, que trata de tener información local, incluso se pueden dejar comentarios sobre los eventos, teniendo por cometido ser el instrumento de todo bailaror, aportando además beneficio para salas, academias y promotores entre otros.

Lo ofrecido por salsero es un proyecto bastante completo, el cual entre otras cosas aporta información sobre el baile, ya sea sobre escuelas, eventos, valoraciones de asistentes, ofreciendo la promoción de artistas, eventos, y la venta de entradas mediante tarifas.

Go&dance

El portal Go&dance (<https://www.goandance.com/>) es un portal para el baile que empezó con la publicación de cursos y eventos de salsa, bachata y kizomba por parte de los profesionales, ofreciendo la posibilidad de que los amantes de estos bailes conozcan locales y eventos para el desarrollo de su afición.

Pretende ser una buena opción para mostrar información sobre los diferentes productos y servicios sobre el baile de una infinidad de bailes, no solo de salsa, bachata y kizomba. Todo ello gracias a los buscadores de eventos, de salidas de baile, de clases de baile y seguimiento de artistas con multitud de filtros. Los eventos y las clases, aparte de aportar la correspondiente información, brinda la posibilidad de compra de clases y eventos. Los artistas, cuentan con una foto y una breve descripción de ellos, tienen la posibilidad de enlazar fotos y vídeos, así como crear publicaciones y mostrar datos de contacto, si hay un evento en el que aparezcan parecerán asociados a los eventos en los que participen, y pueden ser seguidos por los usuarios. En caso de que los artistas reclamen su perfil, estos podrán editar su contenido.

Lo aportado por el portal es la información sobre las posibilidades de muchos bailes para asistir a eventos, salidas de baile, clases, aparte de seguir a artistas. Asimismo, el portal permite la venta de entradas llevándose una comisión por entrada e incluso el cambio de nombre pagando una comisión extra, así como la posibilidad de vender la entrada junto con una habitación de hotel de una agencia colaboradora.

Mydance zone

La plataforma Mydance zone (<https://www.mydance.zone/>) pretende ser la primera red social web y móvil exclusiva para la gente del baile. Siendo útil para buscar donde aprender a bailar, dar con los mejores sitios de baile, festivales y congresos.

Dicha pretensión de ser la primera red social viene acompañada de las posibilidades anteriores, y por otras muchas, siendo una iniciativa gratuita. Por lo que la plataforma incluso ofrece la posibilidad de promover negocios relacionados con la danza, aportando la posibilidad de gestionar de manera gratuita pagos en línea, ya sea para la venta de clases en el caso de escuelas o para la venta de entradas en el caso de los eventos. Próximamente, la plataforma tiene planeado brindar profesionales un entorno digital para que puedan contactar con otros profesionales, creen o se unan a iniciativas e incluso den con oportunidades laborales.

En definitiva, la plataforma apuesta por la gratuidad, por ser la primera red social para el baile, aportando información y gestión para los negocios de los bailes en general, así como en un futuro cercano, la posibilidad de que los profesionales generen conexiones laborales.

Meetup

La red social Meetup de reuniones (<https://www.meetup.com/>) que cuenta con web y aplicación móvil. La finalidad de dicha red social es organizar quedadas con usuarios que tenga intereses comunes.

Esta red social es muy diversa en cuanto a temáticas siendo útil para aprender a hacer cosas nuevas, conocer gente, abandonar la zona de confort, etc. Entre las posibilidades anteriores existe la posibilidad de iniciarse en el baile y dar con ciertos eventos puesto que es muy probable que haya grupos organizados de bailes de pareja en diferentes ciudades de gran tamaño. Asimismo, es posible encontrar gente organizadora de un grupo de baile y también asistentes, con la posibilidad de iniciar una conversación por el chat, en caso de los asistentes, solo si así lo tienen configurado. Las reuniones pueden ser gratis o tener un coste. Siendo usuario si uno quiere disfrutar de más posibilidades en la red social, esta requiere de pago. En caso de pago estos van por PayPal.

Sintetizando, es una red social para reuniones, no específica para el baile, en la cual podría haber alguna que otra reunión relacionada con los bailes de pareja. Sí se podría generar comunicación con interesados de los bailes de pareja, siendo esta dependiente de la configuración de los usuarios. Para disfrutar de ciertas funcionalidades avanzadas en la red social, el pago es un requisito.

Facebook

Facebook (<https://www.facebook.com/>) es una red social accesible como web y también a través de las aplicaciones móviles cuya intención inicial, fue la de que las

personas se pudieran mantenerse en contacto, posibilitar la compartición de información, noticias, imágenes, contenido audiovisual con amigos y/o familiares.

La red social no es exclusivamente para el baile, no obstante, mucha gente aficionada al baile cuenta con una cuenta Facebook, además hay escuelas, organizadores y otros interesados en los bailes de parejas que cuentan con grupos ya sean públicos o privados. De hecho, es una herramienta altamente utilizada sin coste alguno (excepto la publicidad) usada para la promoción del baile ya sea por parte de profesores, academias, locales, organización de eventos, puesto que permite subir fotos, vídeos, crear publicaciones. Tras escándalos, también por cierta desconfianza percibida y verbalizada por algunos aficionados ha dado como fruto una mayor preocupación por la privacidad. Por ello, la exposición de datos es a menudo manejada desde la aportación de los datos exclusivamente necesarios, a restringir la visibilidad de los datos a través de la configuración. Dando como resultado un menor acceso a cierta información relevante a los aficionados a los bailes de salón. Cuenta con chat entre usuarios y por esta red social no se gestionan pagos.

La red social es ampliamente usada para la promoción de los bailes de salón. Gratuita pese a que los anuncios son de pago. Encontrar a aficionados, acceder a cierta información puede ser fácil o difícil en función de los datos expuestos por los usuarios y los grupos.

Instagram

Instagram (<https://www.instagram.com/>) es una red social que empezó a ser muy popular entre los jóvenes, la cual permitía subir fotografías, recibir me gusta. Además, actualmente cuenta con la posibilidad de subir vídeos cortos, comentarles, así como la posibilidad de usar el chat.

Es una red social con un enfoque más visual y juvenil que Facebook pese a que pertenezca a la misma compañía. Pese a ser gratuita permite anuncios de pago, permitiendo promociones. Dada su enfoque más visual es usada como escaparate por muchos usuarios ajenos al baile, pero también por los aficionados y los profesionales del baile. Por ello es normal que muchos artistas, aficionados, academias, organizadores, hagan uso de dicho escaparate digital. Los usos de dicho escaparate van desde la muestra de habilidades por parte de los artistas y aficionados, a la muestra de posibilidades y éxito de academias y eventos organizados. También cuenta con un chat, con el que se podría conseguir con diferentes participantes implicados en los bailes de salón. En esta red social, la exposición es diferente, pero la configuración y cierta minimización de la exposición reduce el acceso a la información y comunicación, también por chat.

En definitiva, es ampliamente usado por jóvenes, más visual de cara a la promoción. Gratuita, aunque crear anuncios promocionales supondría costes. A veces hay restricciones a la exposición y configuraciones que dificultan el acceso a la información y a la comunicación.

WhatsApp

Whatsapp (<https://www.whatsapp.com/>) es una aplicación de mensajería instantánea para móviles inteligentes permitiendo la comunicación con otros teléfono que cuenten con la aplicación móvil.

La aplicación es gratuita, se podría usar en ordenador, pero es dependiente de un número de teléfono móvil. Es muy utilizada principalmente para mandar mensajes a conocidos con los que haya confianza, siempre que se esté dispuesto a intercambiar el número de teléfono y a instalar la aplicación en el teléfono. Cabe la posibilidad de crear grupos, ya sea entre aficionados que se llevan bien, aficionados dispuestos a quedar en un parque para bailar, una sala que quiere mandar flyer o vídeos y así promocionar sus eventos, etc. Eso sí, tenido por presente que uno dentro de un grupo expone el número a el resto de los miembros. Recientemente ha desarrollado ha desarrollado canales, una nueva funcionalidad con potencial, pero no lleva mucho tiempo parecen estar reservados a personalidades, marcas y medios de comunicación.

Es resumidas cuentas WhatsApp es ampliamente usado por su mensajería instantánea, para mandar mensajes, da la posibilidad de crear grupos, pero siempre exponiendo el número de teléfono por lo que hasta ahora se mueve en círculos de confianza.

2.2. Conclusiones de las herramientas

Las conclusiones a las que se puede llegar de las herramientas anteriores tras el estudio es la disparidad que hay entre ellas y las funcionalidades que ofrecen, se van a comentar seguidamente. La idea de es dar con funcionalidades diferenciadoras que puedan aportar diferenciación a la aplicación.

Muchas de las herramientas sí están desarrolladas para el baile como tal. La salsa del baile y Salsero están muy focalizadas en la salsa, bachata y kizomba ofreciendo información sobre los eventos y la venta de entradas, llevándose una comisión por la venta. Otras abarcan más bailes como Go&dance, ofreciendo más cosas como pueda ser el seguimiento de artistas, aparte de la venta de entradas en base a una tarifa. Mydance zone también abarca más bailes, aspirando a ofrecer más cosas como pueda ser un entorno digital para los profesionales sin costes. Aunque ninguna de ellas permite en principio la comunicación por chat con los aficionados.

Sin embargo, otras no están pensadas exclusivamente para el baile. Meetup está pensada para crear reuniones, aunque pueden ser de bailes de pareja. Facebook dirigida a posibilitar el mantener personas en contacto, las cuales pueden ser también aficionados. Instagram, una red social más dirigida hacia lo visual por estar pensada para publicaciones de fotos y videos cortos, en la que se pueden promocionar artistas, eventos, quedadas, etc. WhatsApp ofrece mensajería instantánea muy útil para intercambiar información entre los aficionados ya conocidos. Pese que estas sí permiten la comunicación entre aficionados, suelen presentar barreras en caso de que los aficionados no se conozcan previamente.

Es por ello, por lo que la aplicación *Bailando en pareja* está enfocada en posibilitar la comunicación entre los aficionados a los bailes de pareja, tratando de evitar así las barreras habituales que puedan estar presentes en otras herramientas, sobre todo en caso de no conocerse previamente. Tratando de diferenciarse por su enfoque exclusivo a los bailes de pareja, con una exposición baja y focalizada, siendo también posible configurar la privacidad. Pretendiendo facilitar la comunicación de los aficionados al comenzar con los bailes de pareja, en caso de ir más allá practicando acrobacias, indagar si ciertas academias están bien, si ciertos eventos son interesantes, proponer quedadas, conocer la escena de los bailes de pareja en otros sitios, e incluso intercambiar otro tipo de información entre los aficionados.

A continuación, se reflejan las principales diferencias más relevantes entre las diferentes herramientas de cara a cubrir las necesidades de los aficionados de los bailes de pareja:

Aplicaciones	Específica para el baile	Bailes que abarca	Chat entre usuarios	Obtener referencias sobre clases Preguntar por la escena de los bailes Saber si un evento merece la pena Organizar una posible quedada Tratar de conocer una pareja
La salsa del baile	Sí	Salsa, bachata y kizomba.	No	Solo está la información del anuncio de algo profesional.
Salsero	Sí	Salsa, bachata y kizomba.	No	Información del anuncio de algo profesional y valoraciones.
Go&dance	Sí	Todos	No	Solo está la información del anuncio de algo profesional.
Mydance zone	Sí	Todos	No	Información del anuncio de algo profesional y comunicación con organizador.
Meetup	No	Todos	Limitado por configuración	Información del anuncio, comunicación con organizador y puede que, con los asistentes. Hay posibilidad de organizar.
Facebook	No	Todos	Limitado por configuración	Información del anuncio, posibles comentarios y comunicación por privado con organizador y aficionados.
Instagram	No	Todos	Limitado por configuración	Información del anuncio o publicación, posibles comentarios y comunicación por privado con organizador y aficionados.
WhatsApp	No	Todos	Limitado por intercambio de números.	Comunicación por privado, en caso de grupos se expone el número de los integrantes.
Bailando en pareja	Sí	Muchos de pareja	Sí	Publicación, posible comunicación por privado entre aficionados.

Tabla 1. Principales diferencias entre las diferentes herramientas.

Una vez estudiadas y relejadas las diferentes herramientas empleadas por los aficionados en los bailes de pareja, ya sean centradas en los bailes o no. Se va a dar lugar a la planificación de la aplicación *Bailando en pareja*.

3. Planificación

La planificación detalla las tareas llevadas a cabo en el desarrollo del proyecto, además de incluir estimaciones tanto temporales como económicas para el desarrollo del mismo.

3.1. Metodología de desarrollo

La metodología de desarrollo software precisa de un modelo a seguir establecido previamente, el cual guía la secuencia de tareas, ayudando así tanto al desarrollador como al cliente a lograr producto deseado de manera organizada.

En el desarrollo de la aplicación *Bailando en pareja*, se va a hacer uso de la metodología incremental, por permitir el desglose en múltiples partes, es decir, incrementos. Por lo que se irán aportando funcionalidades y mejorando las funcionalidades al iterar las ya previamente obtenidas, obteniendo un producto completamente funcional después de cada incremento. Así hasta llegar a desarrollar el producto final. De modo que, una vez creada completamente la arquitectura del proyecto, se continua con los incrementos que mayor utilidad y valor aportan al proyecto hasta llegar a la completitud del mismo.

Los incrementos están compuestos por las siguientes etapas, que se repiten hasta la obtención del producto final:

- 1) **Análisis:** Al comenzar, hay que centrarse en la especificaciones de las características más relevantes. Aunque, en incrementos posteriores, las especificaciones serán en base a lo que se evalúe en el incremento previo, considerando si añadir, quitar o modificar los requisitos.
- 2) **Diseño:** una vez realizado el análisis, se toman descripciones para llevar a cabo como deben ser los nuevos componentes.
- 3) **Implementación:** consiste en la codificación de los prototipos funcionales que se deben conseguir en el incremento.
- 4) **Pruebas:** hay que probar si realmente funciona bien la codificación en el incremento.
- 5) **Documentación:** todo lo añadido, modificado y probado, tiene que quedar recogido de manera escrita al finalizar.

El primer incremento del proyecto *Bailando en pareja* comienza implicando a ambos proyectos *ServiciosRest* y *WebMVC*, abarcando la persistencia y seguridad. Una vez logrado el correcto funcionamiento del primer incremento, se continua por los que mayor valor aportan al proyecto final. Así, siguiendo dicho orden, se continua con: *Gestión del usuario* por girar la aplicación entorno a los aficionados permitiéndoles estar en relación con la aplicación; *Gestión de la interacción* puesto que ofrece a los aficionados la capacidad de relacionarse unos con otros; y *Gestión de los reportes*

posibilitando la protección de la red social a través de los usuarios administradores en base a los reportes.

La figura de abajo (ver figura 2) muestra el desarrollo de los 4 incrementos y sus etapas.

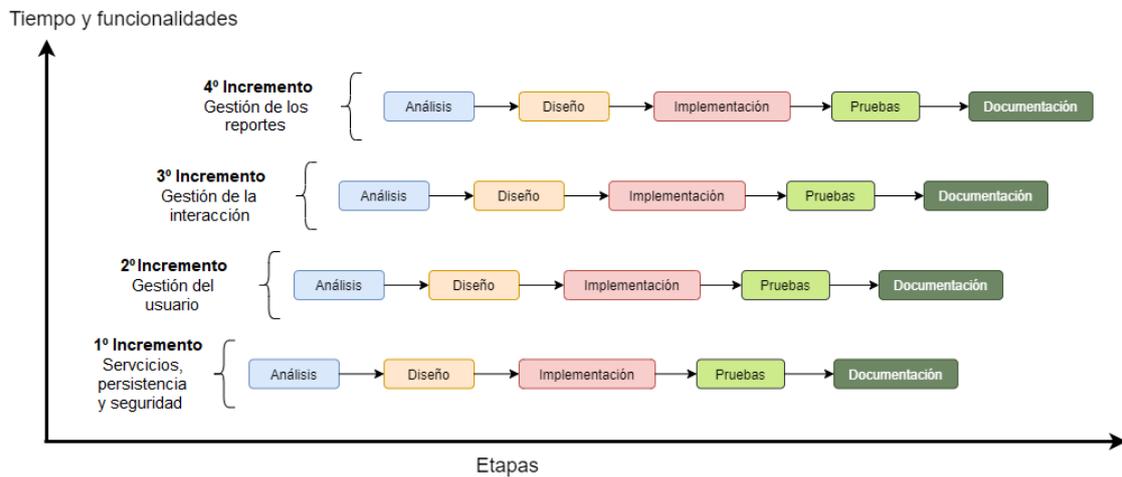


Figura 2. Diagrama de incrementos y etapas.

3.2. Detalle de planificación

El detalle de la planificación del proyecto se da a través de un diagrama de Gantt. En él se reflejan las estimaciones de tiempo para cada iteración, además de las etapas de cada iteración.

Se establece como fecha de comienzo 27.03.23 y de finalización el 02.02.24 del proyecto. Los detalles de inicio, el fin, la duración en días de las iteraciones y etapas se muestran en la siguiente figura (ver tabla 2).

Iteración/Etapa	Días		Inicio	Fin
1° Iteración	75.0	150.0	27-03-23	07-07-23
Análisis	30.0		27-03-23	05-05-23
Diseño	22.0		08-05-23	06-06-23
Implementación	18.0		07-06-23	30-06-23
Pruebas	3.0		03-07-23	05-07-23
Documentación	2.0		06-07-23	07-07-23
2° Iteración	60.0	120.0	10-07-23	29-09-23
Análisis	10.0		10-07-23	21-07-23
Diseño	10.0		24-07-23	04-08-23
Implementación	30.0		07-08-23	15-09-23
Pruebas	5.5		18-09-23	25-09-23
Documentación	4.5		25-09-23	29-09-23
3° Iteración	52.0	104.0	02-10-23	12-12-23
Análisis	10.0		02-10-23	13-10-23
Diseño	10.0		16-10-23	27-10-23
Implementación	25.0		30-10-23	01-12-23
Pruebas	3.5		04-12-23	07-12-23
Documentación	3.5		07-12-23	12-12-23
4° Iteración	38.0	76.0	13-12-23	02-02-24
Análisis	8.0		13-12-23	22-12-23
Diseño	9.0		25-12-23	04-01-24
Implementación	9.0		05-01-24	17-01-24
Pruebas	7.0		18-01-24	26-01-24
Documentación	5.0		29-01-24	02-02-24

Tabla 2. Detalle de la planificación de las iteraciones y las etapas.

El diagrama de Gantt del proyecto se divide por iteraciones en las siguientes figuras (ver figura 3, figura 4, figura 5 y figura 6):



Figura 3. Gantt correspondiente a la primera iteración.

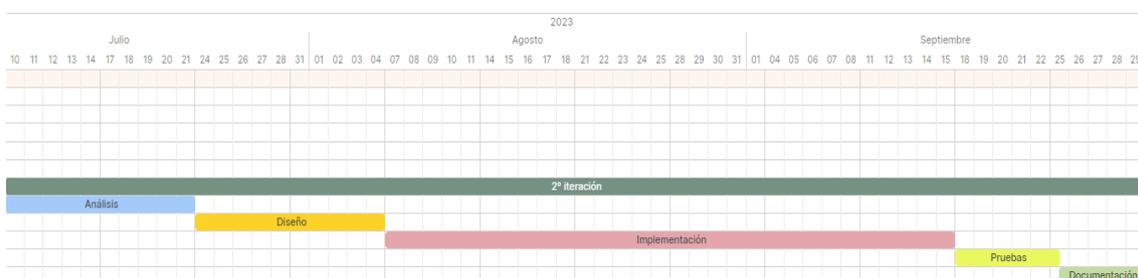


Figura 4. Gantt correspondiente a la segunda iteración.

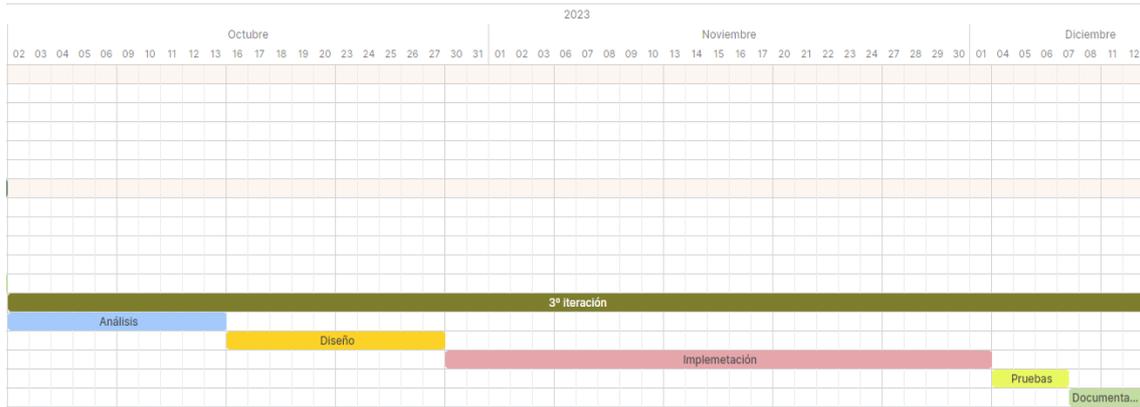


Figura 5. Gantt correspondiente a la tercera iteración.

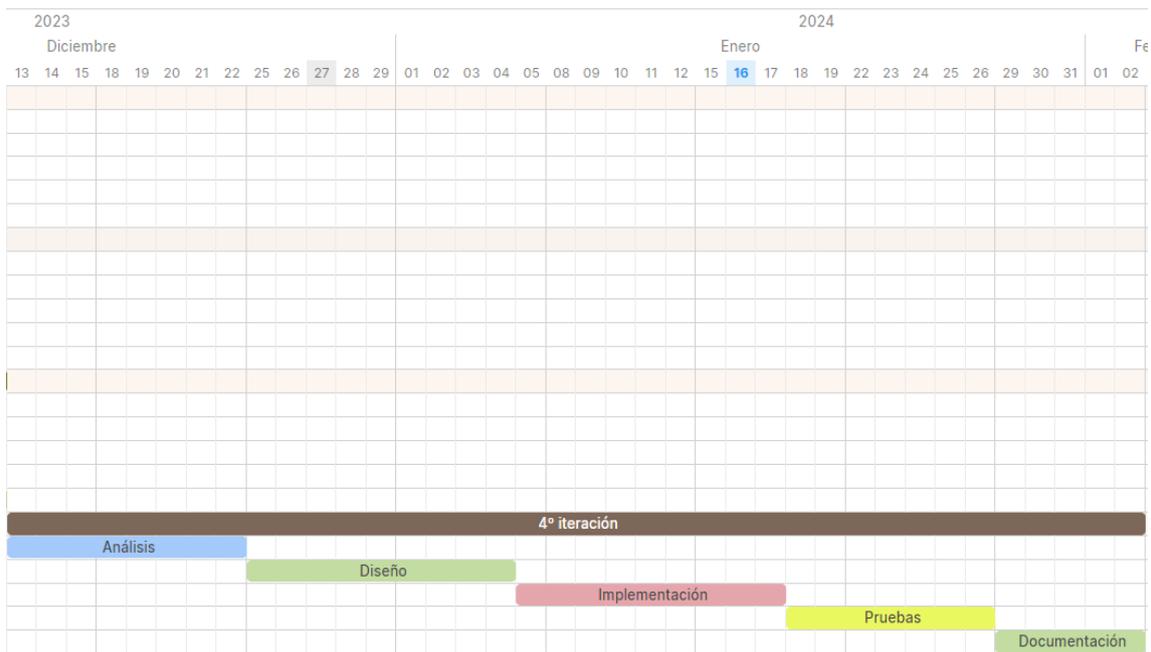


Figura 6. Gantt correspondiente a la cuarta iteración.

3.3. Estimaciones temporales

Las estimaciones temporales del software se pueden realizar con diferentes técnicas. No obstante, en el presente apartado se van a realizar dos estimaciones diferentes para aportar una idea sobre la implicación temporal del presupuesto y poder emitir un presupuesto.

A continuación, se van a realizar dos estimaciones: la estimación por puntos de función y la estimación por COCOMO.

3.3.1. Estimación por puntos de función

La estimación por puntos de función [1] es una técnica desarrollado por Allan Albrecht para la estimación del software. Con dicha técnica, se estima la cantidad de funcionalidad de un software en base a requisitos, categorizándose en: entradas, salidas, consultas, ficheros internos y las interfaces con otras aplicaciones.

Los siguientes pasos son seguidos para la estimación por puntos de función:

- 1) Se identifican y clasifican la fluidez de la información y se les asigna su complejidad correspondiente según la siguiente tabla de determinación de la complejidad (ver tabla 3), que puede ser: sencilla, media o compleja.

Entradas			
	Tipos de datos elementales		
Ficheros referenciados	1 a 4	5 a 15	> 16
0 ó 1	baja	baja	media
2	baja	media	alta
≥ 3	media	alta	alta
Salidas y consultas			
	Tipos de datos elementales		
Ficheros referenciados	1 a 5	6 a 19	> 20
0 ó 1	baja	baja	media
2 ó 3	baja	media	alta
≥ 4	media	alta	alta
Ficheros lógicos internos Ficheros lógicos externos			
	Tipos de datos elementales		
Tipos de registros	1 a 19	20 a 50	> 51
1	baja	baja	media
2 a 5	baja	media	alta
≥ 6	media	alta	alta

Tabla 3. Complejidad.

- **Entrada:** datos introducidos por el usuario a la aplicación.
 - Registro: baja complejidad.
 - Editar perfil: media complejidad.
 - Crear publicación: media complejidad.
 - Editar publicación: media complejidad.
 - Modificar configuración: baja complejidad.

- Dar me gusta: baja complejidad.
 - Añadir a favorito: baja complejidad.
 - Quitar de favorito: baja complejidad.
 - Escribir mensaje: baja complejidad.
 - Reportar: baja complejidad.
 - Desactivar perfil: baja complejidad.
 - Desactivar publicación: baja complejidad.
 - Marcar como leído: baja complejidad.
- **Salida:** datos presentados por la aplicación para el usuario.
 - Ver perfiles: baja complejidad.
 - Ver perfil en detalle: media complejidad.
 - Ver publicaciones: media complejidad.
 - Ver chat: alta complejidad.
 - Ver configuración: baja complejidad.
 - Ver reportes: baja complejidad.
- **Consultas:** peticiones que generan una salida como respuesta inmediata.
 - Login: baja complejidad.
 - Buscar perfiles: media complejidad.
 - Buscar perfiles: media complejidad.
 - Buscar publicaciones: media complejidad.
 - Buscar perfiles reportados: baja complejidad.
 - Buscar publicaciones reportadas: baja complejidad.
- **Ficheros lógicos internos:** datos que son mantenidos por la propia aplicación.
 - Base de datos: alta complejidad.

- **Ficheros lógicos externos:** datos que se encuentran mantenidos por otra aplicación.
 - Servidor SMTP de Gmail: media complejidad.

2) Se recogen los puntos de función no ajustados (PFNA) en la siguiente tabla la contabilidad de los puntos de función sin ajustar (ver tabla 4) según las ponderaciones

Fluidez	Cantidad	Complejidad	Total
Entradas	10	Baja (x3)	30
	3	Media (x4)	12
	0	Alta (x6)	0
Salidas	3	Baja (x4)	12
	2	Media (x5)	10
	1	Alta (x7)	7
Consultas	3	Baja (x3)	9
	3	Media (x4)	12
	0	Alta (x6)	0
Ficheros lógicos internos	0	Baja (x7)	0
	0	Media (x10)	0
	1	Alta (x15)	15
Ficheros lógicos externos	0	Baja (x5)	0
	1	Media (x7)	7
	0	Alta (x10)	0
Total de puntos de función sin ajustar (PFSA)			114

Tabla 4. Puntos de función sin ajustar.

3) Se suman los valores de los 14 factores de ajuste de la tabla de abajo (ver tabla 5). El valor de dichos los factores va de cero a cinco, inclusive, representando la influencia: donde 0 representa una ausencia de influencia, 1 una mínima, 2 una moderada, 4 una significativa y 5 una fuerte.

14 factores de ajuste	Valor [0-5]
1. Comunicación de datos.	2
2. Procesamiento distribuido.	3
3. Rendimiento.	4
4. Abundante empleo de configuración.	2
5. Transacciones frecuentes.	4
6. Entrada en línea de los datos.	4
7. Diseño eficiente para el usuario final.	3
8. Actualización en línea de los datos.	3
9. Complejidad de la lógica interna de la aplicación.	2
10. Instalación sencilla.	2
11. Código reusable.	2
12. Facilidad de conversión e instalación.	2
13. Múltiples localizaciones.	2
14. Facilidad ante cambios.	3
Suma de los valores de ajuste (SVA)	38

Tabla 5. Factores de ajuste y su valoración.

- 4) Se calculan los puntos de función ajustados (PFA), para adaptar las estimaciones a las condiciones de trabajo.

$$\text{PFA} = \text{PFSA} \cdot (0,65 + (0,01 \cdot \text{SVA}))$$

$$\text{PFA} = 114 \cdot (0,65 + (0,01 \cdot 38)) = 114 \cdot (1,03)$$

$$\text{PFA} = 117,42$$

- 5) Se obtienen la estimación de líneas de código mediante las equivalencias entre los lenguajes de programación y los puntos de función. La primera tabla de abajo (ver tabla 6) toma las equivalencias publicadas por QSM, Quantitative Software Management [2] y la segunda tabla de abajo (ver tabla 7) muestra tanto las líneas de código estimadas por lenguaje como las totales.

Equivalencias entre líneas de código y puntos de función	
Lenguaje	LDC/PF
HTML	34
JavaScript	47
J2EE	46

Tabla 6. Equivalencias entre LDC y PF.

Lenguaje	LDC/PF	Proporción	LDC
HTML/Thymeleaf	34	25%	998
JavaScript	47	15%	828
J2EE	46	60%	3241
Total de líneas de código			5067

Tabla 7. Estimaciones finales de líneas de código.

3.3.2. Estimación de esfuerzo por COCOMO

La estimación de esfuerzo por COCOMO o modelo constructivo de costes [1] fue desarrollado por Barry W. Boehm, para estimar los costes en proyectos software. El modelo constructivo de costes cuenta con tres submodelos: el básico, el intermedio y el detallado. Así como tres modos de desarrollo: orgánico, semiseparado y el empotrado.

Los tres modos de desarrollo son:

- Orgánico: para proyectos pequeños con pocas innovaciones tecnológicas, a menudo desarrollados en un pequeño grupo en el que todos tienen experiencia en las tareas a realizar.
- Semiseparado: para proyectos medianos en cuanto a tamaño y complejidad, siendo este modo el intermedio de los otros modos.
- Empotrado: proyectos grandes y complejos en cuanto a procedimientos operacionales, altamente condicionados por el software y el hardware, en el que el equipo está formado por diferentes equipos.

Los tres modelos son:

- Básico: útil para la estimación del esfuerzo y tiempo de desarrollo en función del tamaño del proyecto y la manera de desarrollarlo para así determinar el coste.
- Intermedio: estima de manera más compleja al incorporar quince factores de ajuste del esfuerzo.
- Detallado: aporta mejoras sobre el intermedio, puesto que las variables son dependientes de la fase de la estimación y establece jerarquías entre niveles de productos.

La estimación de esfuerzo por COCOMO se va a realizar tomando el modo de desarrollo orgánico por el reducido número de trabajadores, pese a no contar con una amplia experiencia en el desarrollo de software. Asimismo, se va a tomar también el modelo intermedio por considerarse más adecuado al considerar las quince variables de predicción.

Según el modo se aplican sus correspondientes coeficientes, tal y como se muestra en la tabla de abajo (ver tabla 8):

Modo COCOMO	Valores			
	a	b	c	d
Orgánico	2.4	1.05	2.5	0.38
Semi-empotrado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.2	2.5	0.32

Tabla 8. Coeficientes COCOMO.

Los quince factores de ajuste del esfuerzo que se tienen en cuenta en el modelo intermedio se encuentran en la siguiente tabla (ver tabla 9), en la que los factores seleccionados se encuentran en gris para, los cuales son multiplicados para obtener $m(x)$:

Factores de ajuste	Muy bajo	Bajo	Medio	Alto	Muy alto	Extra
Fiabilidad exigida	0.75	0.88	1.00	1.15	1.40	
Dimensión de la BBDD		0.94	1.00	1.08	1.16	
Complejidad del producto	0.70	0.85	1.00	1.15	1.30	1.65
Restricciones de tiempo de ejecución			1.00	1.11	1.30	1.66
Restricciones de almacenamiento			1.00	1.06	1.21	1.56
Volatilidad en el hardware		0.87	1.00	1.15	1.30	
Restricciones de tiempo de respuesta		0.87	1.00	1.07	1.15	
Habilidad analista	1.46	1.19	1.00	0.86	0.71	
Experiencia con el software	1.29	1.13	1.00	0.91	0.82	
Habilidad codificadora	1.42	1.17	1.00	0.86		
Experiencia con el hardware	1.21	1.10	1.00	0.90		
Experiencia con aplicaciones	1.14	1.07	1.00	0.95		
Empleo de téc. de programación modernas	1.24	1.10	1.00	0.91	0.82	
Empleo de herramientas software	1.24	1.10	1.00	0.91	0.83	
Restricciones de duración del proyecto	1.23	1.08	1.00	1.04	1.10	

Tabla 9. Factores de ajuste COCOMO.

$$\text{Esfuerzo (E)} = a \cdot (\text{KLDC})^b \cdot m(x)$$

$$\text{Esfuerzo (E)} = 2,4 \cdot (5067)^{1,05} \cdot 0,819027109 = 15.258$$

$$\text{Esfuerzo (E)} = 15.258 \text{ personas/mes}$$

$$\text{Tiempo de desarrollo (Tdev)} = c \cdot E^d$$

$$\text{Tiempo de desarrollo (Tdev)} = 2,5 \cdot (15.258)^{0,38} = 7,041$$

$$\text{Tiempo de desarrollo (Tdev)} \approx 7 \text{ meses}$$

$$\text{Número de personas} = E/Tdev$$

$$\text{Número de personas} = 15,258/7,041$$

$$\text{Número de personas} \approx 3 \text{ personas}$$

3.4. Presupuesto

El presupuesto tiene por cometido obtener una estimación del coste real del desarrollo de la aplicación *Bailando en pareja*. Para ello, se van a incluir los recursos software, hardware y humanos empleados en dicho desarrollo.

3.4.1. Presupuesto software

El presupuesto software refleja el software usado para desarrollar proyecto software y su memoria. Para la elaboración del presupuesto, se tiene presente que ciertos software de pago permiten su uso más allá de la duración del proyecto, por lo que lo más adecuado sería reflejar un porcentaje de uso estimado sobre el total en el presupuesto del proyecto.

Concepto	Precio	Porcentaje de uso	Coste en proyecto
Ubuntu Server 20.04	0€	-	0€
Windows 10 (incluido con el portátil)	0€	-	0€
Java (TM) SE Development Kit 17	0€	-	0€
MySQL Workbench 8	0€	-	0€
Eclipse IDE 2022-09 R	0€	-	0€
Apache Netbeans IDE 15	0€	-	0€
Postman 10.16.0	0€	-	0€
Office 2021 Professional Plus	29.95€	10%	2.99€
Draw.io	0€	-	0€
Tom's planner	0€	-	0€
Template	20€	100%	20€
Presupuesto total para el software			22.99€

Tabla 10. Presupuesto software.

3.4.2. Presupuesto hardware

El presupuesto hardware plasma el hardware empleado para el desarrollo del proyecto incluyendo la memoria.

El presupuesto software tiene en cuenta todo el software usado para desarrollar proyecto software y su memoria. Para la elaboración del presupuesto, se tiene presente que ciertos software de pago permiten su uso más allá de la duración del proyecto, por lo que lo más adecuado sería reflejar un porcentaje de uso estimado sobre el total en el presupuesto del proyecto.

Concepto	Precio	Porcentaje de uso	Coste en proyecto
Lenovo IdeaPad 3 i7-1065G7 8GB	649€	15%	97,35€
Inphic ratón inalámbrico y recargable	10,99€	15%	1,65€
WiFi – Router (en biblioteca)	0€	-	0€
Presupuesto total para el hardware			99€

Tabla 11. Presupuesto hardware.

3.4.3. Presupuesto en recursos humanos

El presupuesto de recursos humanos tiene en cuenta las estimaciones relacionadas con las compensaciones los empleados por el trabajo realizado.

Partiendo de que un analista-programador recién egresado, con menos de 3 años de experiencia laboral, puede esperar un salario medio de alrededor de 20.100 € brutos por año. (Tomando como fuente plataforma de búsqueda de empleo Jobted, según lo recoge en el siguiente enlace: <https://www.jobted.es/salario/analista-programador>, consultado por última vez el 24.01.2024).

No obstante, el coste para la empresa de un trabajador puede ser de oscilar entre 31,98% a 37,63%. Tomándose un 38%, aunque este trabajo no cuente con peligrosidad, pretendiendo así compensar posibles incidencias y algún otro coste. Ascendiendo el coste empresarial de un analista-programador a 27.738€ por año. (Según el artículo publicado por el abogado laboralista Alejandro Aradas sobre la base de cotización del 2022, que se encuentra en el enlace: <https://www.cuestioneslaborales.es/porcentaje-de-cotizacion-a-la-seguridad-social-del-trabajador-y-empresa/>, consultado por última vez 26.01.2024).

Dicho sueldo dividido en 12 pagas son 2.311,5 €/mes. Y como el proyecto supondría el pago de 3 desarrolladores por 7 meses, resultaría un desembolso de 48.541,5 €.

3.4.4. Presupuesto total del proyecto

El presupuesto total del proyecto se compone de la estimación de costes del software, hardware y recursos humanos. Estas estimaciones se recogen en la siguiente tabla (ver tabla 12) para mostrar la suma que da como resultado presupuesto estimado para el proyecto.

Concepto	Importe
Software	22.99€
Hardware	99€
Recursos humanos	48.541,5€
Presupuesto estimado para el proyecto	48.663,49€

Tabla 12. Presupuesto estimado para el proyecto.

Finalmente se ha obtenido un presupuesto estimado de 48.663,49 € para la creación de la aplicación *Bailando en pareja*.

3.5. Desarrollo y esfuerzo real de la aplicación

El desarrollo y el esfuerzo real de la aplicación viene a dejar constancia de cómo ha ido progresando el avance de la aplicación con su documentación.

Durante el desarrollo, ha habido días en muchos casos con jornadas de más de ocho horas e incluso muchos fines de semana. No obstante, también ha habido bastantes días en los cuales apenas se han dedicado horas o incluso ninguna hora a progresar. Lo cual ha dado lugar a retrasos, cuyos principales motivos han sido los siguientes:

- Compaginación con el estudio de un máster, con infinidad de trabajos a entregar y periodos de exámenes.
- Días de agotamiento en los que ha sido mejor hacer una pequeña pausa.
- Complicaciones con el desarrollo de la aplicación y con algún diagrama.
- La documentación no acaba con tener todo documentado, sino que hay que pasar todo a la memoria del TFG y que quede bien, enfrentándose a que todo se descuadre, suponiendo más tiempo del estimado para ello.

En lugar de haber terminado el 02.02.24, se ha generado un retraso de entorno a tres meses con respecto a la estimación. Siendo difícil determinar si se hubiera podido cumplir la estimación, principalmente por la irregularidad del ritmo llevado, la presión y en algunos casos el agotamiento producido por los estudios con los que se ha compaginado.

El presupuesto se mantiene por ser el estipulado en función de las funcionalidades ofrecidas por el software y de los costes según el mercado. Sin embargo, al coste estimado para la aplicación habría que añadirle los costes de despliegue de la aplicación en una plataforma.

4. Análisis

El análisis va a abordar los comportamientos que el sistema debe desempeñar a través de diferentes procesos de estudio.

4.1. Actores

Los actores del sistema van a ser nombrados y descritos, exponiendo así todos los actores participantes en la aplicación *Bailando en pareja*. Los actores son usuarios y/o sistemas externos al sistema que interaccionan para demandar una funcionalidad del sistema y que este genera respuestas.

Los actores de la siguiente tabla (ver tabla 13) son los que van a interactuar con el sistema:

Actor	Descripción
Usuario no registrado	Actor que simboliza a aquellos usuarios que no se han registrado previamente ante el sistema y cuya única interacción posible con la aplicación es la de registrarse.
Usuario registrado	Actor que representa a los usuarios que ya se han registrado previamente en la aplicación, creando así una identificación en el sistema y por lo tanto pueden recuperar la contraseña, iniciar sesión y cerrarla.
Usuario general	Actor que engloba a los usuarios ya registrados previamente y que además también han iniciado sesión en la aplicación. Por lo tanto, pueden interactuar con el sistema, a excepción de la posibilidad de ver reportes y gestionar la desactivación de los perfiles y publicaciones.
Usuario administrador	Actor que comprende al usuario o usuarios añadidos como administradores, pese a haberse registrados previamente como usuario, estos han sido añadidos manualmente en la BBDD como administradores. Tras el inicio de sesión pueden hacer uso de las funcionalidades de cualquier <i>usuario general</i> , solo que cuentan con una búsqueda diferente enfocada en buscar usuarios y publicaciones concretas, incluso inactivadas y con reportes. Teniendo la capacidad de ver los reportes y gestionar la activación, a diferencia que el resto de los usuarios. La posibilidad de eliminar el perfil está capada por seguridad.

Tabla 13. Listado de actores del sistema.

Los actores mostrados anteriormente en la tabla (ver tabla 13), tienen una relación jerárquica, por lo que hay una relación de herencia entre ellos. De esta manera, las funcionalidades que puede realizar un actor son recibidas por otro como propias. La jerarquía existente entre los actores se puede ver en la siguiente figura (ver figura 7).

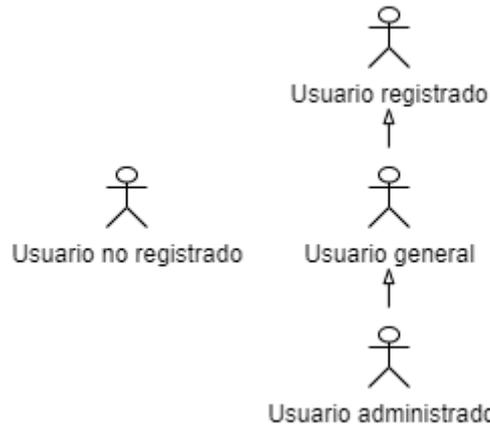


Figura 7. Jerarquía de los actores del sistema.

La jerarquía de la figura anterior (ver figura 7) indica que el actor *Usuario administrador* hereda los casos de uso del actor *Usuario general* y a su vez, el *Usuario general* los hereda del *Usuario registrado*. Pese a la herencia de casos de uso, hay que destacar que para los administradores la búsqueda está modificada y la posibilidad eliminar el perfil está capada por seguridad.

Las acciones posibles que se pueden producir entre los actores y el sistema se representan mediante los casos de uso. Los casos de uso aportan los requisitos de usuario mostrando el contexto del sistema tras la interacción por parte del usuario.

4.2. Requisitos de usuario

Los requisitos de usuario son las funcionalidades que los usuarios pueden ejecutar usando el sistema. A continuación, en los siguientes apartados se van a especificar los requisitos de usuario mediante los casos de uso.

4.2.1. Listado de casos de uso

Antes de proceder a la especificación de los diferentes de casos de uso se van a encuadrar en la siguiente tabla (ver tabla 14) por actores:

Actor	Casos de uso
Usuario no registrado	UNR1 - Registrarse.
Usuario registrado	UR1 - Login. UR2 - Recuperar contraseña por email. UR3 - Logout.
Usuario general	UG1 - Ver perfil propio. UG2 - Consultar configuración propia. UG3 - Gestionar las notificaciones por email de los mensajes recibidos. UG4 - Gestionar las notificaciones por email de las publicaciones de los favoritos. UG5 - Gestionar la visibilidad de la edad. UG6 - Gestionar la visibilidad del email. UG7 - Editar perfil. UG8 - Eliminar perfil. UG9 - Cambiar contraseña. UG10 - Ver publicaciones propias. UG11 - Crear publicación. UG12 - Editar publicación. UG13 - Eliminar publicación. UG14 - Dar me gusta. UG15 - Consultar la lista propia de favoritos UG16 - Buscar perfiles. UG17 - Ver perfiles. UG18 - Gestionar la inclusión como favorito. UG19 - Reportar perfil. UG20 - Escribir mensaje privado al perfil. UG21 - Buscar publicaciones. UG22 - Ver publicaciones. UG23 - Reportar publicación UG24 - Acceder al chat UG25 - Visualizar mensajes. UG26 - Marcar como leído.
Usuario administrador	UA1 - Buscar perfil reportado UA2 - Buscar publicación reportada. UA3 - Visualizar reportes del perfil. UA4 - Visualizar reportes de la publicación. UA5 - Gestionar la activación del perfil. UA6 - Gestionar la activación de la publicación.

Tabla 14. Casos de uso encuadrados por actores.

La ejecución de los casos de uso puede darse como consecuencia de la ejecución directa por parte del actor representándose con una línea continua o por parte de otro caso de uso vinculado representándose con líneas discontinuas. Habiendo dos posibles tipos de relaciones entre los casos de uso:

- **Extend:** la ejecución de la secuencia de casos de uso puede ser parcial y viene determinada por el actor, que puede hacer que se ejecute el

siguiente caso de uso o no. Tratándose de una relación en la que se da la especialización. Para un mejor entendimiento se utiliza la figura (ver figura 8) presentada tras sus explicaciones, en la cual se observa que la ejecución del CU1 podría dar lugar a la ejecución de su secuencia CU2, en función de lo que determine el usuario en CU1.

- **Include:** la ejecución total de un caso de uso implica la ejecución de otro caso de uso. Para aportar un mejor comprensión se utiliza la figura (ver figura 8) presentada después de la explicación, en la cual se ve que la ejecución del CU3 lleva necesariamente a la ejecución de CU4.

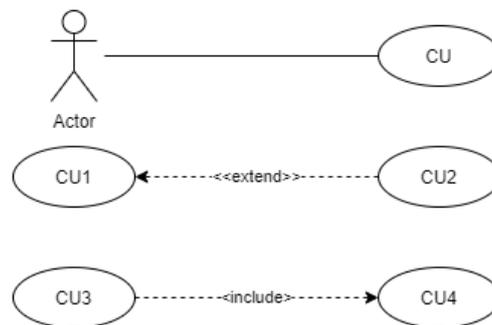


Figura 8. Ejemplos de asociaciones de casos de uso.

Los diagramas de casos de uso permiten la representación tanto de los actores como de las relaciones existentes entre los casos de uso, como se puede apreciar en el siguiente diagrama de casos de uso (ver figura 9):

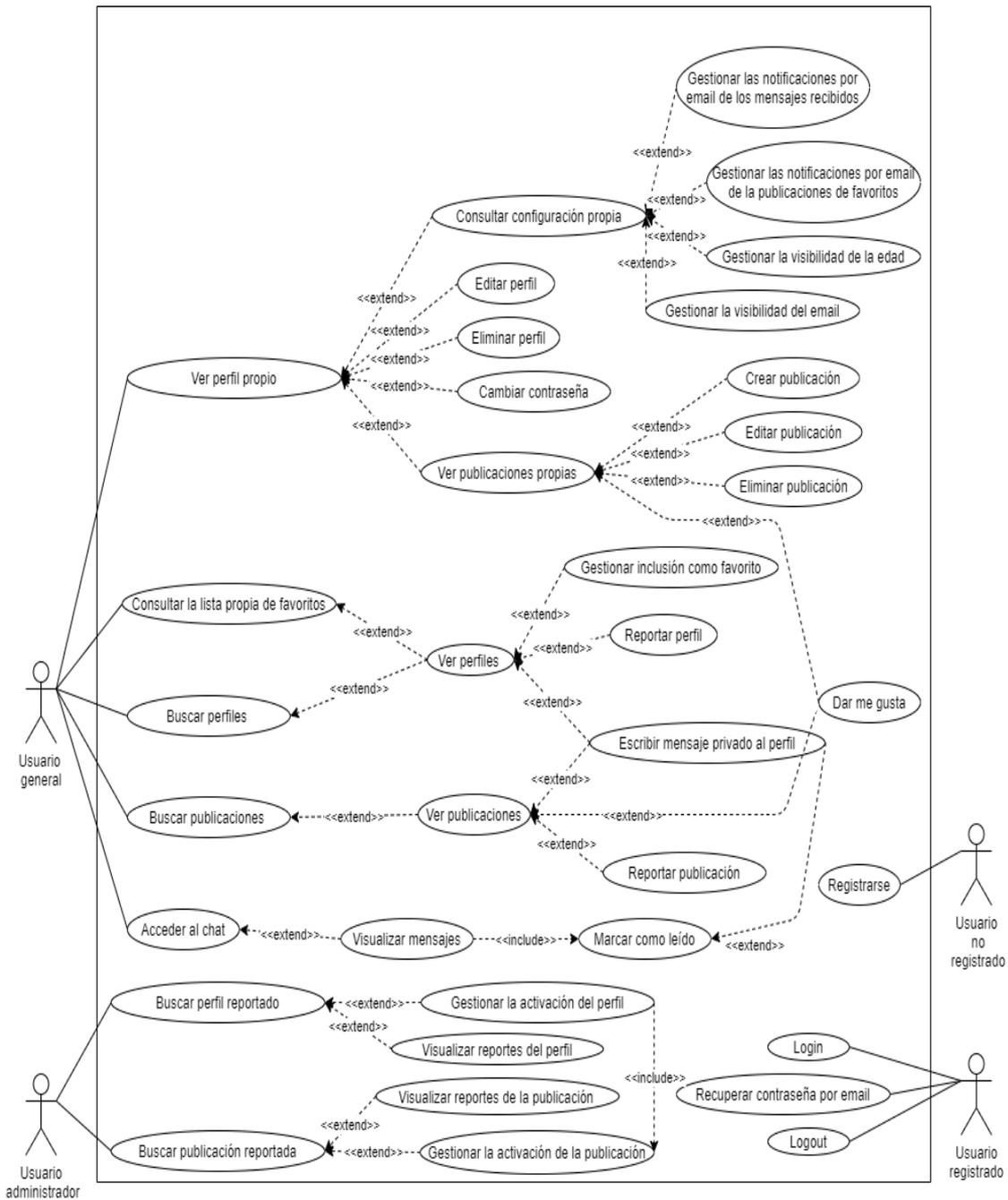


Figura 9. Diagrama de casos de uso.

4.2.2. Especificación de casos de uso

En el presente apartado se aportan las tablas de especificación de los casos de uso de los actores *Usuario no registrado* y *Usuario registrado*. No obstante, para ver la especificación de todos los casos de uso es preciso acudir al **ANEXO II** donde se encuentra al resto por su gran extensión.

Usuario no registrado

Id y caso de uso	UNR1 - Registrarse
Descripción	Un usuario se inscribe en la aplicación.
Actor	Usuario no registrado.
Precondición	Pre-1: El email del usuario no se puede haber registrado previamente.
Secuencia	1. El usuario inserta los campos pedidos por la aplicación enfrentándose a la validación (Exc-1 yExc-2). 3. Los datos se guardan en la BBDD, previo cifrado de la contraseña.
Postcondición	Pos-1: Se notifica el correcto registro.
Excepciones	Exc-1: Si algún campo no pasa la validación, muestra el error y vuelve al paso 1. Exc-2: Si ya existe ese email de usuario en la BBDD, muestra error y no permite la reinscripción.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EU.02, RI.EU.03, RI.EU.05, RI.EDU.01, RI. EDU.05, RI. EDU.07, RI.ERA.01, RI. ERA.02, RI.RT1.01, RI. RT1.02 Dep. - Reglas de negocio: RN.01, RN.02, RN.03
Observaciones	La contraseña, se almacena en la base de datos cifrada. El usuario debe ser mayor de edad. La contraseña debe tener de 8 a 41 caracteres, 1 letra mayúscula, 1 letra minúscula, 1 número y 1 carácter especial.

Tabla 15. Caso de uso UNR1.

Usuario registrado

Id y caso de uso	UR1 - Login
Descripción	Un usuario se identifica en la aplicación para iniciar sesión.
Actor	Usuario registrado.
Precondición	Pre-1: El usuario tiene que haberse registrado previamente en la aplicación.
Secuencia	1. El usuario inserta las credenciales en los campos de la aplicación. 2. La aplicación comprueba los campos con la BBDD (Exc-1).
Postcondición	Pos-1: La aplicación comienza la sesión de usuario.
Excepciones	Exc-1: Si los credenciales no son correctos y coincide con los de algún usuario, la aplicación muestra error y se comienza en el punto 1. Exc-2: Si el email del usuario no está activado, notifica el error y se comienza en el punto 1.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EU.02, RI.EU.05. Dep. - Reglas de negocio: RN.01, RN.02.
Observaciones	

Tabla 16. Caso de uso UR1.

Id y caso de uso	UR2 - Recuperar contraseña por email
Descripción	Un usuario solicita recibir la contraseña en su correo electrónico.
Actor	Usuario registrado.
Precondición	Pre-1: El usuario tiene que haberse registrado previamente en la aplicación.
Secuencia	1. El usuario inserta el email de registro en la aplicación. 2. La aplicación verifica el email (Exc-1 yExc-2) a través e una consulta en la BBDD.
Postcondición	Pos-1: La aplicación notifica el envío del correo.
Excepciones	Exc-1: Si el email no coincide con los de algún usuario, la aplicación muestra error y se comienza en el punto 1. Exc-2: Si el email del usuario no está activado, la aplicación muestra error y se comienza en el punto 1.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EU.02, RI.EU.05. Dep. - Reglas de negocio: RN.01, RN.04
Observaciones	La contraseña se descifra para mostrarse en el email, recomendando su cambio.

Tabla 17. Caso de uso UR2.

Id y caso de uso	UR3 - Logout
Descripción	Un usuario cierra su sesión.
Actor	Usuario registrado.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación.
Secuencia	1. El usuario solicita acabar con su sesión en la aplicación. 2. La aplicación redirige a la página de inicio de sesión y registro.
Postcondición	Pos-1: La aplicación acaba con la sesión del usuario.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01 Dep. - Reglas de negocio: RN.01, RN.02.
Observaciones	

Tabla 18. Caso de uso UR3.

4.3. Reglas de negocio

Las reglas de negocio son normas previamente establecidas, normalmente antes del comienzo del desarrollo software, las cuales apoyan el modelado de la lógica de negocio. Estas reglas de negocio están formadas por regulaciones, estándares y políticas para dirigir las funcionalidades de un sistema, organizaciones y empresas.

ID	Reglas de negocio
RN.01	La aplicación web será accesible desde cualquier navegador web convencional.
RN.02	Los usuarios accederán a la aplicación web mediante un identificador único, no modificable y una contraseña modificable.
RN.03	Las contraseñas serán seguras y se almacenarán de forma ilegible en la base de datos.
RN.04	Los usuarios podrán recuperar su contraseña a través de su email de registro.
RN.05	Todo usuario contará con un perfil propio, podrá publicar y comunicarse por privado.
RN.06	Los usuarios dispondrán de buscadores con filtros para encontrar perfiles y publicaciones.
RN.07	Un usuario podrá añadir a otros usuarios como favoritos y quitarles de favoritos.
RN.08	Los perfiles serán configurables para recibir notificaciones por email, así como para preservar la privacidad.
RN.09	Solo habrá un máximo de tres publicaciones vigentes por usuario y estas podrán recibir me gustas.
RN.10	Tanto los perfiles de un usuario como las publicaciones serán editables por el propio usuario.
RN.11	Tanto los perfiles como las publicaciones podrán ser reportadas por otros usuarios.
RN.12	Los usuarios administradores contarán con una búsqueda especial permitiéndoles buscar por email, nombre, activación y motivo del reporte.
RN.13	Solo los usuarios administradores podrán ver los reportes de publicaciones y perfiles.
RN.14	Solo los usuarios administradores podrán desactivar y activar tanto las publicaciones como los perfiles.
RN.15	Solo los usuarios administradores no podrán eliminar su propio perfil en la aplicación.
RN.16	La desactivación de una publicación solo debe afectar a la visibilidad de esa publicación para los usuarios no administradores.
RN.17	La eliminación o desactivación de un perfil, implica la invisibilidad del perfil junto con sus publicaciones y conversaciones del chat, excepto para los administradores.

Tabla 19. Reglas de negocio.

4.4. Requisitos funcionales

Los requisitos funcionales tratan de describir como debe responder el sistema satisfaciendo las necesidades de los usuarios bajo ciertas condiciones.

Motivado por un mejor entendimiento y organización, los casos de uso se van a presentar siguiendo el mismo orden en el que aparecen en la tabla del apartado 4.2.1 (ver

tabla 14). No obstante, en este apartado solo se muestran los cuatro primeros casos de uso, correspondientes a los actores *Usuario no registrado* y *Usuario registrado*. Debido a la extensión, el resto de las especificaciones de los requisitos funcionales se encuentran en el ANEXO III de esta memoria.

Usuario no registrado

ID	Caso de uso
UNR1	Registrarse
ID	Requisitos funcionales
RF.001	El sistema mostrará el formulario de registro tras seleccionar registrarme.
RF.002	El sistema validará que el email de registro no esté ya registrado.
RF.003	El sistema validará que la nueva contraseña sea segura.
RF.004	El sistema validará que ambas contraseñas introducidas sean iguales.
RF.005	El sistema validará que la fecha de nacimiento corresponda a la de un mayor de edad.
RF.006	El sistema mostrará el error si algún campo introducido no supera la validación.
RF.007	El sistema registrará al usuario si todos los campos superan la validación mostrado que el usuario ha sido registrado exitosamente.
RF.008	El sistema almacenará la contraseña de manera cifrada.
RF.009	El sistema registrará los datos editables del perfil no pedidos en el registro lo más neutros y ambiguos posibles, es decir, el género con otro, los bailes con otros y la ciudad con Madrid.

Tabla 20. Requisitos funcionales en el caso de uso UNR1.

Usuario registrado

ID	Caso de uso
UR1	Login
ID	Requisitos funcionales
RF.010	El sistema mostrará el formulario de inicio de sesión tras seleccionar iniciar sesión.
RF.011	El sistema verificará que el email esté registrado en la base de datos.
RF.012	El sistema verificará que el usuario del email esté activo.
RF.013	El sistema comprobará si coinciden las contraseñas cifradas.
RF.014	El sistema mostrará un error en caso de que los credenciales de acceso no sean correctos.
RF.015	El sistema iniciará una sesión con el usuario si los credenciales introducidos son correctos redirigiéndole a la página de inicio.

Tabla 21. Requisitos funcionales en el caso de uso UR1.

ID	Caso de uso
UR2	Recuperar contraseña por email
ID	Requisitos funcionales
RF.016	El sistema mostrará el formulario de recuperación de contraseña tras seleccionar que olvido la contraseña.
RF.011	El sistema verificará que el email esté registrado en la base de datos.
RF.012	El sistema verificará que el usuario del email esté activo.
RF.017	El sistema mostrará un error en caso de que el formato del email no sea correcto.
RF.018	El sistema mostrará un error en caso de que el email no exista.
RF.019	El sistema creará un correo con la contraseña y será enviado gracias al servicio SMTP de Gmail.

Tabla 22. Requisitos funcionales en el caso de uso UR2.

ID	Caso de uso
UR3	Logout
ID	Requisitos funcionales
RF.020	El sistema invalidará la sesión de usuario.
RF.021	El sistema redirigirá al usuario a la página de <i>login</i> .

Tabla 23. Requisitos funcionales en el caso de uso UR3.

4.5. Requisitos no funcionales

Los requisitos no funcionales imponen restricciones y particularidades describiendo como llevará a cabo el sistema su funcionamiento.

4.5.1. Atributos de calidad

Los atributos de calidad especifican las características del sistema para satisfacer adecuadamente las funcionalidades del usuario. Se van a exponer seguidamente los atributos de calidad de varios tipos: disponibilidad, escalabilidad, rendimiento, seguridad y usabilidad.

Disponibilidad: Atributos que indican el tiempo funcionamiento del sistema.

ID	Descripción
ATD1	La aplicación ofrecerá su funcionalidades las 24 horas los 365 días del año.

Tabla 24. Atributos de calidad para la disponibilidad.

Escalabilidad: Atributos que se refieren a la capacidad de trabajo del sistema.

ID	Descripción
ATE1	La estructura física del sistema deberá permitir ampliar la carga de trabajo frente a un aumento de la demanda de recursos sin que se afecte el funcionamiento, evitando así costes elevados.

Tabla 25. Atributos de calidad para la escalabilidad.

Rendimiento: Atributos vinculados con el tiempo de reacción del sistema.

ID	Descripción
ATR1	La respuesta del sistema no sobrepasará el segundo con conexiones de usuario superiores a los 100 Mbps.
ATR2	El sistema deberá soportar la interacción de al menos 50 usuarios con la aplicación.

Tabla 26. Atributos de calidad para el rendimiento.

Seguridad: Atributos vinculados con la protección del acceso a la información.

ID	Descripción
ATS1	El inicio de sesión solo será posible previo registro de los usuarios con nombre de usuario, email, contraseña y fecha de nacimiento correspondiente a la mayoría de edad.
ATS2	La aplicación solo admitirá contraseñas seguras compuestas por un mínimo de 8 y un máximo de 41 caracteres, con al menos una letra mayúscula, una letra minúscula, un número y un carácter especial.
ATS3	Las contraseñas de los usuarios se almacenarán de manera ininteligible en la base de datos bajo encriptado en base 64.

Tabla 27. Atributos de calidad para la seguridad.

Usabilidad: Atributos que aportan facilidad de uso a los usuarios.

ID	Descripción
ATU1	La aplicación contará con una interfaz intuitiva además de contar con un manual por si hubiera dudas.
ATU2	El usuario verá los datos editables tanto de su perfil como de sus publicaciones en el momento en el momento de editarlos.
ATU3	La aplicación contará con diseño adaptativo siendo accesible desde cualquier navegador convencional.

Tabla 28. Atributos de calidad para la usabilidad.

4.5.2. Restricciones de implementación y arquitectura

Los restricciones de arquitectura e implementación y arquitectura van a estructurar como se va a desarrollar el software haciendo uso de las tecnología.

ID	Descripción
RIYA1	La aplicación contará una arquitectura de 3 capas.
RIYA2	La aplicación se desarrollará haciendo uso del lenguaje Java y del framework Spring Boot.
RIYA3	La capa de presentación usará el motor de plantillas Thymeleaf integrado con el framework Spring Boot.

Tabla 29. Restricciones de arquitectura e implementación.

4.5.3. Requisitos de interfaces externas

Los requisitos de interfaces externas hacen referencia a las tecnologías que ponen el sistema en comunicación con otros sistemas software.

ID	Descripción
RIE1	La aplicación hará uso del protocolo HTTP para la comunicación entre los dos proyectos de la aplicación.
RIE2	Los servicios REST podrán comunicarán la información mediante los formatos JSON y XML.
RIE3	La aplicación contará con un servicio REST proporcionando información necesaria para esta en formato JSON
RIE4	El envío de email se hará mediante el servidor SMTP de Google.

Tabla 30. Requisitos de interfaces externas.

4.6. Esquema conceptual de datos

El esquema conceptual de datos describe la estructuración de los datos. Para ello, se hace uso del modelo entidad-relación y del diccionario de datos. El modelo entidad-relación refleja las relaciones entre entidades indicando su cardinalidad, además de los atributos de cada entidad. En el diccionario de datos aporta más información proporcionando más detalles sobre los elementos de los datos.

4.6.1. Modelo entidad-relación

El modelo entidad-relación es una representación gráfica compuesta principalmente por entidades, relaciones y atributos. Las entidades, además de contar con sus atributos, cuentan con conexiones que representan las relaciones entre estas, así como con una cardinalidad para indicar la cantidad de veces que las entidades se relacionan. Dichas entidades con sus atributos, en base a la cardinalidad darán lugar al diseño de la estructura lógica que posteriormente dará lugar a la implementación de una base de datos. Para ello, se hará uso del diagrama entidad-relación, puesto que es una manera de reflejar gráficamente como está estructurada la información.

El esquema conceptual se refleja gráficamente, tras el estudio de las necesidades de estructuración de los datos, gracias a la siguiente figura (ver figura 10).

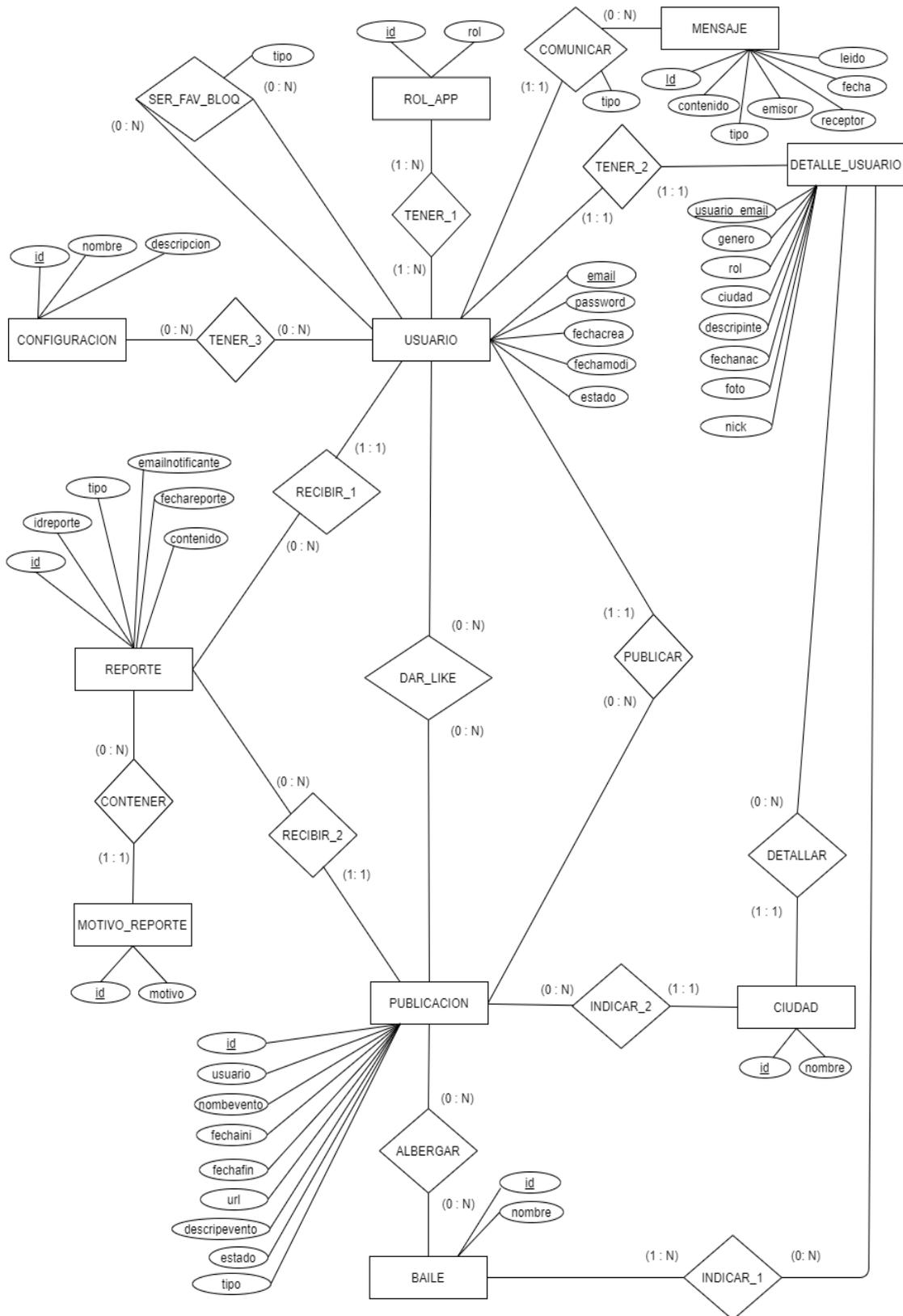


Figura 10. Diagrama entidad-relación.

4.6.2. Diccionario de datos

El diccionario de datos recoge en su documentación información técnica sobre diferentes elementos de la base de datos. En la documentación se encuentra recogida información detallada sobre las tablas de la base de datos y sus atributos. Estos datos técnicos aportados por el diccionario son de gran ayuda de cara a la usabilidad y explotación de la base de datos.

Los siguientes requisitos de información se corresponden con un atributo estando clasificados por tablas:

Entidad ‘usuario’

Entidad encargada de representar a los aficionados registrados con sus datos principales en la aplicación.

ID	Atributo	Descripción	Tipo y longitud	Características
RI.EU.01	email	Email del registro, que identifica de manera única al aficionado.	varchar(45)	Primary key, not null.
RI.EU.02	password	Contraseña del aficionado cifrada.	varchar(45)	Not null. (Tiene de 8 a 41 caracteres, 1 letra mayúscula, 1 letra minúscula, 1 número y 1 carácter especial).
RI.EU.03	fechacrea	Fecha de creación del registro del usuario.	date	Not null.
RI.EU.04	fechamodi	Fecha de modificación del algún dato del usuario.	date	Default null. (Para futuras implementaciones).
RI.EU.05	estado	Determina si el usuario está activo.	int,	Not null, (0: “desactivado”, 1: “activado”).

Tabla 31. Tabla ‘usuario’ con sus atributos.

Entidad 'rol_app'

Entidad encargada de representar el rol o roles del usuario en la aplicación. La asignación para pasar de usuario aficionado a ser además usuario administrador se hace manualmente en la base de datos, en concreto en la tabla 'usuario_rol'.

ID	Atributo	Descripción	Tipo y longitud	Características
RI.ERA.01	id	Identificador único de los roles de la aplicación.	int	Primary key, not null, auto_increment.
RI. ERA.02	rol	Determina el rol que tiene el usuario.	varchar(45)	Default null, (1: 'Amin', 2: 'Aficionado').

Tabla 32. Tabla 'rol_app' con sus atributos.

Entidad 'detalle_usuario'

Entidad encargada de representar las particularidades de los usuarios de la aplicación. De esta manera, los usuarios tienen una tabla con los datos más relevantes 'usuario' y otra tabla con sus detalles 'detalle_usuario'.

ID	Atributo	Descripción	Tipo y longitud	Características
RI.EDU.01	usuario_email	Email del registro que identifica de manera única los detalles del usuario.	varchar(45)	Primary key, not null.
RI.EDU.02	genero	Determina el género del usuario.	int	Not null. (1: "Masculino", 2: "Femenino", 3: "Otro")
RI. EDU.03	rol	Determina el rol del usuario al bailar.	int	Not null. (1: "Dirige", 2: "Sigue", 3: "Ambos")
RI. EDU.04	descripinte	Lugar para explicar los intereses del perfil de usuario.	varchar(255)	Default null.

RI. EDU.05	fechanac	Fecha de nacimiento del usuario.	date	Not null.
RI. EDU.06	foto	Imagen par el perfil del usuario.	varchar(255)	Default null.
RI. EDU.07	nick	Nombre y apellido/s del usuario.	varchar(45)	Default null.

Tabla 33. Tabla 'detalle_usuario' con sus atributos.

Entidad 'configuracion'

Entidad encargada de representar los posibles ajustes de configuración de los usuarios de la aplicación en cuanto a notificaciones y seguridad.

ID	Atributo	Descripción	Tipo y longitud	Características
RI.EC.01	id	Identificador único de la configuración.	int	Primary key, not null, auto increment.
RI.EC.02	nombre	Denominación de la parte configurable.	varchar(45)	Not null.
RI.EC.03	descripcion	Explicación de la parte configurable.	varchar(45)	Not null.

Tabla 34. Tabla 'configuracion' con sus atributos.

Entidad 'reporte'

Entidad encargada de representar los denuncias dirigidas hacia los perfiles de los usuarios o hacia las publicaciones publicadas por los usuarios en la aplicación.

ID	Atributo	Descripción	Tipo y longitud	Características
RI.ER.01	id	Identificador único de los reportes.	int	Primary key, not null, auto increment.
RI.ER.02	idreportado	Almacena la identificación del perfil o publicación reportada.	varchar(45)	Default null. (La identificación de un perfil es un email, y la de una publicación es un entero).

RI. ER.03	tipo	Referencia al origen, si es perfil o una publicación.	int	Default null. (1: Perfil, 2: Publicación).
RI. ER.04	fechareporte	Fecha con el día del reporte.	date	Default null.
RI. ER.05	contenido	Fecha de nacimiento del usuario.	varchar(45)	Default null.

Tabla 35. Tabla 'reporte' con sus atributos.

Entidad 'motivo_reporte'

Entidad encargada de representar las causas de los reportes hacia los perfiles de los usuarios o hacia sus publicaciones en la aplicación.

ID	Atributo	Descripción	Tipo y longitud	Características
RI.EMR.01	id	Identificador único de los motivos de reporte.	int	Primary key, not null, auto_increment.
RI.EMR.02	motivo	Causa por la que se ha decidido denunciar.	varchar(45)	Not null. (1: 'Perfil falso', 2: 'Contenido inapropiado', 3: 'Estafa o uso comercial', 4: 'Discurso de odio', 5: 'Comportamiento fuera de la app', 6: 'Menor de edad', 7: 'No me interesa').

Tabla 36. Tabla 'motivo_reporte' con sus atributos.

Entidad 'baile'

Entidad encargada de representar los bailes de los usuarios y de las publicaciones de la aplicación.

ID	Atributo	Descripción	Tipo y longitud	Características
RI.EB.01	id	Identificador único de los bailes.	int	Primary key, not null, auto increment.
RI.EDU.02	nombre	Denominación del baile.	varchar(20)	Not null. (1: 'Salsa', 2: 'Bachata', 3: 'Kizomba', 4: 'Tango', 5: 'Swing', 6: 'Zouk', 7: 'Acrobático', 8: 'Otro').

Tabla 37. Tabla 'baile' con sus atributos.

Entidad 'ciudad'

Entidad encargada de representar las ciudades de los usuarios y de las publicaciones de la aplicación.

ID	Atributo	Descripción	Tipo y longitud	Características
RI.ECI.01	id	Identificador único de las ciudades.	int	Primary key, not null, auto increment.
RI.ECI.02	ciudad	Nombre de la ciudad.	varchar(45)	Not null. (1: 'A Coruña', 2: 'Albacete', 3: 'Alicante', 4: 'Almería', 5: 'Ávila', ... 50: 'Valencia', 51: 'Valladolid', 52: 'Vitoria', 53: 'Zamora', 54: 'Zaragoza'). (Ciudades españolas ordenadas por orden alfabético.)

Tabla 38. Tabla 'ciudad' con sus atributos.

Entidad ‘publicacion’

Entidad encargada de representar las publicaciones de los usuarios de la aplicación.

ID	Atributo	Descripción	Tipo y longitud	Características
RI.EP.01	id	Identificador único de las publicaciones.	int	Primary key, not null, auto increment.
RI.EP.02	tipo	Determina el tema de la publicación.	varchar(45)	Default null. (1: “Quedada”, 2: “Clases”, 3: “Evento”).
RI.EP.03	fechaini	Fecha del inicio del evento.	date	Default null.
RI.EP.04	fechafin	Fecha del final del evento.	date	Default null.
RI.EP.05	decripevento	Explicación del evento.	varchar(255)	Default null.
RI.EP.06	url	Localizador uniforme de recursos del evento.	varchar(45)	Default null.
RI.EP.07	estado	Determina el estado de la publicación.	int	Default null. (0: “Desactivada”, 1: “Activa”). Aunque la desactivación de un perfil hace no visibles las publicaciones incluso para los administradores.

Tabla 39. Tabla ‘publicacion’ con sus atributos.

Entidad ‘mensaje’

Entidad encargada de representar la información de comunicación entre los usuarios de la aplicación.

ID	Atributo	Descripción	Tipo y longitud	Características
RI.EM.01	id	Identificador único de los mensajes.	int	Primary key, not null, auto increment.
RI.EM.02	contenido	Mensaje propiamente dicho.	varchar(255)	Default null.
RI.EM.03	Tipo	Determina si el mensaje se muestra o no.	int	Default null. (Todos a 0, más posibilidades de cara al futuro).
RI.EM.04	fecha	Fecha de inicio del evento.	datetime	Default null.
RI.EP.05	leído	Determina si el receptor ha leído el mensaje.	int	Default null. (0: "No leído", 1: "Leído").

Tabla 40. Tabla mensaje con sus atributos.

Relación 'tener_1' => 'usuario_rol'

'tener_1' es como aparece representada la relación en el diagrama entidad-relación por su representatividad en el diagrama entidad-relación, pese a que el nombre de la tabla sea 'usuario_rol' para una mejor identificabilidad. Representa la unión entre el usuario y sus roles en la aplicación. La relación proviene del vínculo entre las entidades 'usuario' y 'rol_app'. Hay que destacar que la aplicación no hace ninguna asignación para los administradores, por lo que el administrador deberá añadir en la base de datos, directamente en la tabla 'usuario_rol' el email de que quiera ser administrador como se aprecia en la estructura de la siguiente sentencia: "INSERT INTO `bailandoenparejadb`.`usuario_rol` (`usuario`, `rol_app`) VALUES ('emaildeejemplo@xxx.xxx', '1');" de la base de datos para conseguir las funcionalidades extra propias de los usuarios administradores.

La cardinalidad de la relación es (N: N), un usuario tiene uno o varios roles en la aplicación (1: N) y un rol en la aplicación es tenido por uno o varios usuarios (1: N).

ID	Atributo	Descripción	Tipo y longitud	Características
RI.RT1.01	usuario	Identificador único del usuario.	varchar(45)	Primary key, not null,
RI. RT1.02	rol_app	Identificador único del rol.	varchar(45)	Primary key, not null.

Tabla 41. Tabla 'usuario_rol' con sus atributos.

Relación ‘tener_2’

‘tener_2’ aparece reflejada en el diagrama entidad-relación como relación. No posee tabla, relacionando la entidad ‘usuario’ y la entidad ‘detalle_usuario’. La relación proviene del vínculo entre las entidades ‘usuario’ y ‘detalle_usuario’ representando la unión entre datos más relevantes y los datos que son detalles del perfil de usuario.

La cardinalidad de la relación es (1: 1), un usuario tiene un detalle (1: 1), y un detalle es tenido por un usuario (1: 1).

Relación ‘tener_3’ => ‘configuración_usuario’

‘tener_3’ es como aparece presentada la relación sobre el diagrama entidad-relación, aunque el nombre de la tabla sea ‘configuración_usuario’. Representa la unión de usuario y sus posibles configuraciones en la aplicación por parte de las entidades ‘usuario’ y ‘configuración’.

La cardinalidad de la relación es (N: N), un usuario puede tener ninguna o varias configuraciones para usar en la aplicación (0: N) y una configuración puede ser tenida por ningún usuario o varios usuarios de la aplicación (0: N).

ID	Atributo	Descripción	Tipo y longitud	Características
RI.RT3.01	configuracion	Identificador único la configuración.	int	Primary key, not null.
RI.RT3.02	usuario	Identificador único del usuario.	varchar(45)	Primary key, not null.

Tabla 42. Tabla ‘configuración_usuario’ con sus atributos.

Relación ‘ser_fav_bloq’ => ‘relacionar’

‘ser_fav_bloq’ es el nombre de la relación que se encuentra sobre el diagrama entidad-relación, en cambio el nombre de la tabla es ‘relacionar’. Representa el vínculo entre los propios usuarios de la aplicación, vinculando la tabla ‘usuario’ con ella misma, posibilitando que los usuarios pueden vincularse como favoritos y en un futuro como bloqueados o de otro modo. Los añadidos a favoritos aparecen en una lista y pueden recibir emails de sus publicaciones si así se configura.

La cardinalidad de la relación es (N: N), un usuario puede hacer favorito a ninguno o varios usuarios (0: N) y un usuario puede ser favorito de ninguno o varios usuarios (0: N).

ID	Atributo	Descripción	Tipo y longitud	Características
RI.RR.01	usuarioori	Identificador único del usuario designador del vínculo.	varchar(45)	Primary key, not null.
RI.RR.02	usuariodes	Identificador único del usuario designado del vínculo.	varchar(45)	Primary key, not null.
RI.RR.03	tipo	Determina el carácter del vínculo.	varchar(45)	Not null. ('Favorito') (En un futuro podría ser 'Bloqueado', por ejemplo).

Tabla 43. Tabla 'relacionar' con sus atributos.

Relación 'recibir_1'

'recibir_1' se encuentra como relación en el diagrama entidad-relación. No hay tabla alguna para esta relación. Vincula las tablas 'usuario' y 'reporte' de manera que representa la conexión entre los usuarios y el posible reporte asociado a un usuario.

La cardinalidad de la relación es (1: N), un reporte es recibido por un usuario (1:1) y un usuario puede recibir ningún o muchos reportes (0: N).

Relación 'recibir_2'

'recibir_2' es como se plasma la relación sobre el diagrama entidad-relación. No existe ninguna tabla para la conexión entre la tabla 'usuario' y la tabla 'reporte', no obstante, la relación representa la asociación de las publicaciones con sus reportes correspondientes.

La cardinalidad de la relación es (1: N), el reporte es recibido por una publicación (1: 1) y una publicación puede recibir ningún o muchos reportes (0: N).

Relación ‘indicar_1’ => ‘detalleusuario_baile’

‘indicar_1’ es una relación que se encuentra en el diagrama entidad-relación, sin embargo, el nombre de la tabla es ‘detalle_usuariobaile’. Con la relación se representa la vinculación de los bailes por los que los usuarios muestran interés en su perfil, así es como se relacionan las tablas ‘detalle_usuario’ y ‘baile’.

La cardinalidad de la relación es (N: N), un detalle de usuario puede indicar uno o varios bailes (1: N) y ninguno o varios bailes pueden ser indicados por los detalles de los usuarios (0: N).

ID	Atributo	Descripción	Tipo y longitud	Características
RI.RDUB.01	baile	Identificador único del baile.	int	Primary key, not null.
RI.RDUB.02	detalleusuario	Identificador único de los detalles de los usuarios.	varchar(45)	Primary key, not null.

Tabla 44. Tabla ‘detalleusuario_baile’ con sus atributos.

Relación ‘indicar_2’

‘indicar_2’ es una relación que aparece en el diagrama entidad-relación. No tiene tabla dicha relación, representando entre las tablas ‘publicacion’ y ‘baile’ indicando la ciudad de una publicación.

La cardinalidad de la relación es (1: N), una publicación indica una ciudad (1: 1), y una ciudad está indicada en ninguna publicación o varias publicaciones (0: N).

Relación ‘albergar’ => ‘publicacion_baile’

‘albergar’ es una relación localizada en el diagrama entidad-relación, aunque el nombre sea ‘publicacion_baile’. La vinculación representa el baile o bailes que contiene la publicación, relacionando las tablas ‘publicacion’ y ‘baile’.

La cardinalidad de la relación es (N: N), una publicación puede indicar ninguno o varios bailes (0: N) y bailes puede estar albergado ninguna o varias publicaciones (0: N).

ID	Atributo	Descripción	Tipo y longitud	Características
RI.RA.01	publicacion	Identificador único de la publicación.	int	Primary key, not null.
RI.RA.02	baile	Identificador único del baile.	int	Primary key, not null.

Tabla 45. Tabla 'publicacion_baile' con sus atributos.

Relación 'comunicar'

'comunicar' es una relación que se encuentra en el diagrama entidad-relación. No posee tabla pese a vincular las tablas 'usuario' y 'mensaje' representando la conexión entre los usuarios y los mensajes que se envían.

La cardinalidad de la relación es (1: N), un motivo es contenido en ningún o varios reportes (0: N), y un reporte contiene por un motivo (1: 1).

Relación 'contener'

'contener' es una relación que se localiza en el diagrama entidad-relación. Pese a no corresponder una tabla, relaciona las tablas 'reporte' y 'motivo_reporte' representando la conexión entre los reportes y los motivos de estos tienen.

La cardinalidad de la relación es (1: N), un usuario comunica ningún o muchos mensajes (0: N), y un mensaje comunicado por un usuario (1: 1).

Relación 'detallar'

'detallar' se encuentra como relación en el diagrama entidad-relación. No posee tabla alguna, pese a relacionar las tablas 'detalle_usuario' y 'ciudad', representando el vínculo entre los detalles del usuario donde se encuentra la ciudad donde dice vivir.

La cardinalidad de la relación es (1: N), un detalle de usuario detalla una ciudad (1: 1), y en una ciudad es detallada por ninguno o varios de los detalles de usuario (0: N).

Relación 'dar_like' => 'likes'

'dar_like' es una relación del diagrama entidad-relación, no obstante, la tabla de esta relación tiene el nombre de 'like'. Representa el vínculo entre los usuarios y las publicaciones que han recibido un me gusta por parte de los usuarios, relacionando las tablas 'usuario' y 'publicacion'.

La cardinalidad de la relación es (N: N), un usuario puede dar al 'like' a ninguna o muchas publicaciones (0: N) y una publicación puede recibir ser dada al 'like' por ninguno o muchos usuarios (0: N).

ID	Atributo	Descripción	Tipo y longitud	Características
RI.EL.01	publicacion	Identificador único de la publicación.	int	Primary key, not null.
RI.EL.02	usuario	Identificador único del usuario.	varchar(45)	Primary key, not null.

Tabla 46. Tabla 'likes' con sus atributos.

5. Diseño del sistema

El diseño del sistema se va a abordar en los siguientes apartados para responder a las funcionalidades especificadas por los requisitos en el análisis, determinando así, los componentes del sistema.

5.1. Arquitectura lógica

La arquitectura lógica está compuesta por los componentes lógicos del sistema, así como los vínculos entre ellos. Para ello se va a hacer uso de la tecnología Thymeleaf y Java Spring Boot, tal y como se ha determinado en el apartado del análisis de requisitos RIYA1, RIYA2 y RIYA3 (ver apartado 4.5.2.).

La arquitectura de la aplicación *Bailando en pareja*, consta de dos proyectos el proyecto *WebMVC* y el proyecto *ServiciosRest*, dando lugar a una arquitectura de tres capas. Estando dichas capas construidas sobre la inmediata inferior, de manera que la capa superior se comunicará a través de la inmediatamente inferior, y también recibirá la comunicación de la capa inmediatamente inferior.

WebMVC es una aplicación web diseñada bajo el patrón MVC. Mientras que *ServiciosRest* es un servicio que aporta una interfaz web apoyada en HTTP posibilitando así a la aplicación web las cuatro operaciones básicas en la base de datos.

La aplicación *Bailando en pareja* consta de tres capas:

- **Presentación:** (Thymeleaf y Java Spring Boot): su cometido es manejar la visualización por pantalla para los usuarios mediante el motor de plantillas html de Thymeleaf a través del navegador web. Gracias a la integración de la librería thymeleaf-spring en el POM de Maven en el proyecto Java *WebMVC* diseñado bajo el framework Spring Boot. Donde los controladores comunican las información entre las vistas y los modelos guardan los datos (controller y controller.bean).
- **Lógica de negocio:** capa que lleva a cabo el control de los movimientos de datos y las operaciones fundamentales conocidas como “CRUD”, que son la creación, la lectura, la actualización y eliminación para gestionar los datos del sistema. En esta capa se distinguen esencialmente los siguientes grupos:
 - **Lógica de la aplicación web (servicios de WebMVC):**
 - **service:** clases interfaz que contienen los métodos sin implementar de la lógica de negocio de *WebMVC* para realizar las llamadas a las operaciones fundamentales de buscar, crear, actualizar y borrar.

- **service.impl:** clases implementadas que cuentan con la anotación '@Service', y la implementación los métodos de la lógica de negocio de *WebMVC*, para realizar las llamadas a las operaciones fundamentales de buscar, crear, actualizar y borrar.
- **Acceso al servicio (clientes de *WebMVC*):**
 - **clientes:** clases declaradas con la notación '@Component' en las que se registra el bean dentro del framework. Tras ello se inyecta el valor con la mediante la anotación '@Value' y llamará solo una vez el método iniciar por tener la anotación 'PostConstruct' después de la inicialización de las propiedades del bean.
 - **clientes.converter:** clase declarada con la notación '@Component' que registra el bean del usuario dentro del framework para convertir el usuario del controlador a los clientes.
 - **clientes.exceptions:** clases para manejar excepciones por errores en las solicitudes HTTP inválidas o no autorizadas.
 - **clientes.formatters:** clases que serializan y desrealizan los objetos de fecha, y también fecha y hora, usadas en las clases DTO del controlador.
 - **clientes.dto:** clases serializadas para la transferencia de datos de objetos con sus métodos getter y setter, además de los métodos hashCode y equals.
 - **email:** clases para el envío de correos electrónicos gracias al servicio SMTP de Google.
- **ApiREST (controladores de *ServiciosREST*):**
 - **controller:** clases controladoras con la anotación '@RestController', para simplificar la creación de servicios web. Se usa en combinación con la anotación '@RequestMapping' para los métodos de manejo de solicitudes: '@GetMapping', '@PostMapping', '@PutMapping' y '@DeleteMapping'.
 - **controller.bean:** con las clases no serializadas que sirven de modelo con sus métodos getter y setter, junto con su hashCode y equals para guardar los datos.
 - **controller.formatter:** clases para serializar y deserializar los objetos de fecha y también fecha y hora usadas en las clases DTO del controlador.
 - **controller.dto:** con las clases serializadas para la transferencia de datos de objetos con sus métodos getter y setter, además de su hashCode y equals.

- **Lógica de datos (servicios de *ServiciosREST*):**
 - **service:** clases interfaz que contienen los métodos sin implementar de la lógica de negocio de *ServiciosREST* para buscar, crear, actualizar y borrar.
 - **service.converter:** clases declaradas con la notación '@Component' registra el *bean* dentro del framework, siendo la anotación padre de las anotaciones '@Controller', '@Service' y '@Repository', para convertir los objetos del repositorio a los controladores.
 - **service.impl:** clases implementadas que cuentan con la anotación '@Service', además de la implementación los métodos de la lógica de negocio de *ServiciosREST*, para buscar, crear, actualizar y borrar entre otros.
- **Acceso a datos (modelos de *ServiciosREST*):**
 - **entity:** clases que entidad declaradas con la anotación '@Entity' y '@Table' para especificar la asignación entre la entidad la tabla de la base de datos permitiendo la abstracción de la base de datos mediante el mapeo de datos en objetos.
 - **entity.repository:** clases interfaz que poseen la anotación '@Repository' para encontrar objetos o colecciones que son grupos de objetos. También se usa la anotación '@Modifying' y '@Query' para ejecutar las sentencias de borrado y actualización.
- **Acceso a los datos:** capa encargada del acceso al almacenaje de la base de datos (MySQL, se puede ver más detalladamente en el punto 6.2.), mediante el Java Persistence Api que contiene JDBC, sirviéndose de Hibernate que es la implementación de JPA y necesitando el drive de Java MySQL Driver.

La seguridad de la arquitectura está representada por un elemento transversal a las capas. Autenticación, la cual hace necesaria el registro previo en la aplicación y el posterior inicio de sesión en para acceder a la aplicación garantizando la autorización del acceso.

Los siguientes figuras muestran la arquitectura lógica de los dos proyectos por separado para más claridad (ver figura 11 y figura 12):

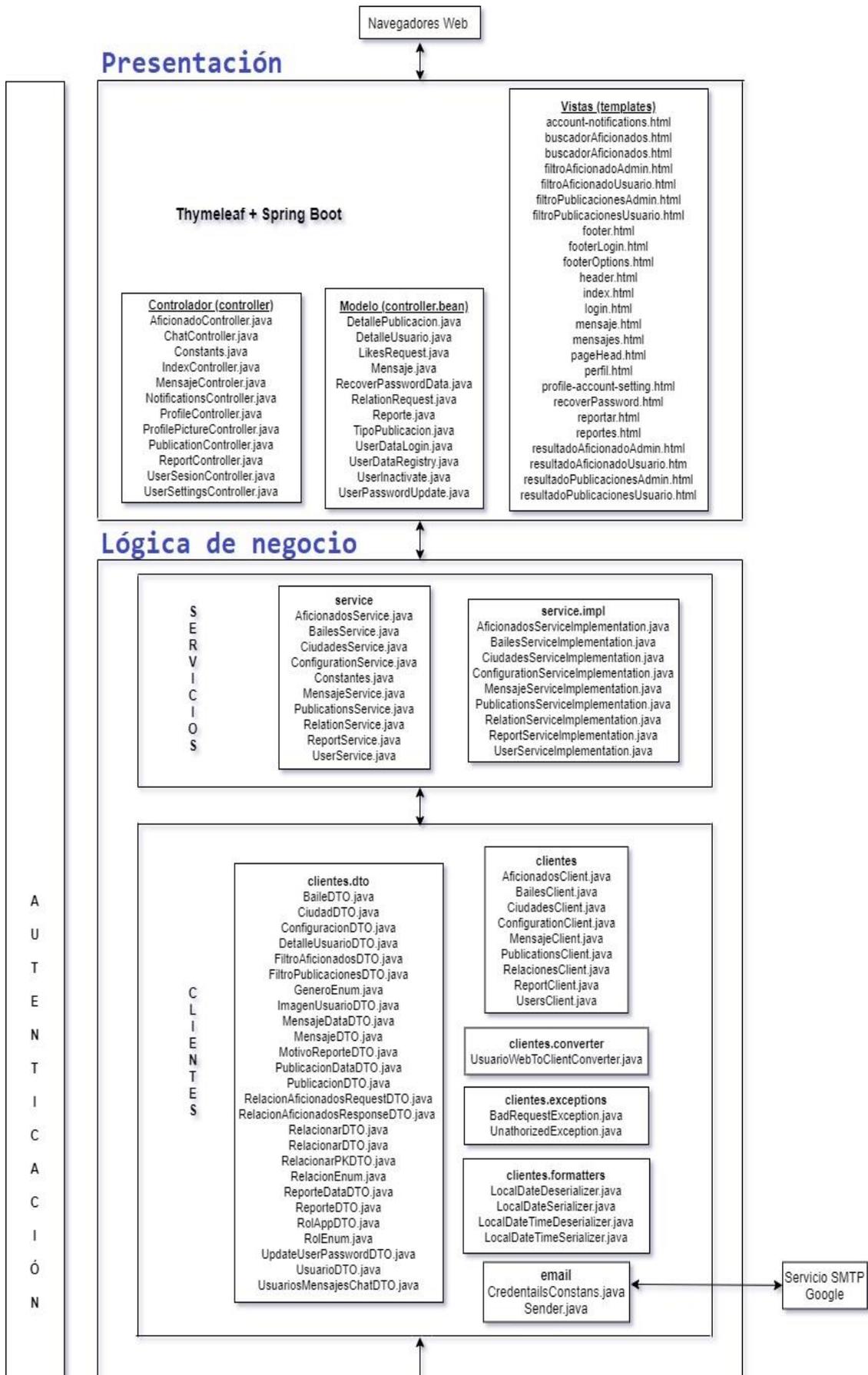


Figura 11. Arquitectura lógica WebMVC.

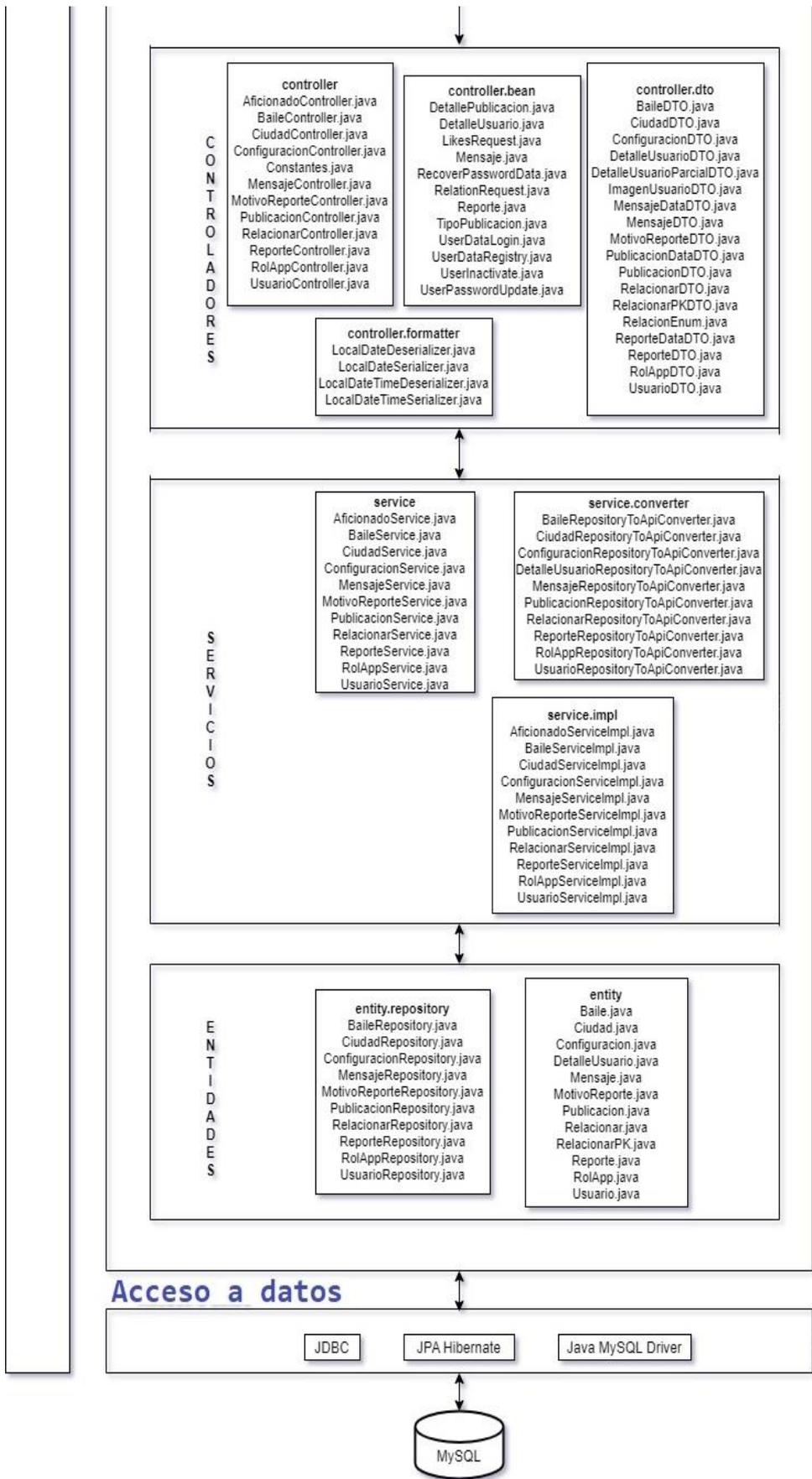


Figura 12. Arquitectura lógica *ServiciosREST*.

5.2. Arquitectura física

La arquitectura física está compuesta por los componentes físicos esenciales para el funcionamiento del sistema, además de la relación entre ellos.

La arquitectura propuesta para la aplicación en caso de despliegue podría ser la siguiente y se puede observar a continuación (ver figura 13).

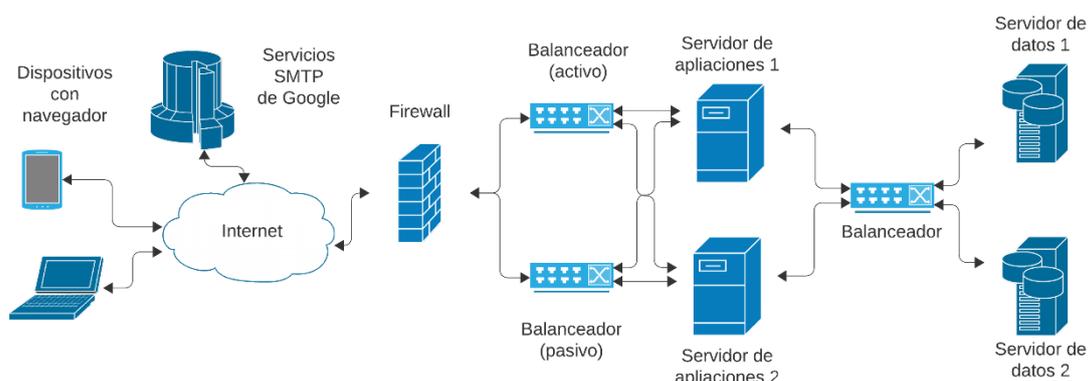


Figura 13. Arquitectura física.

El acceso a la aplicación *Bailando en pareja* por parte de los usuarios se llevaría a cabo mediante un navegador web, conectado a internet teniendo que atravesar un cortafuegos o firewall, barrera con la misión de bloquear los servicios no autorizados mientras permite autorizados. Tras el cortafuegos, vendrían los balanceadores de carga, los cuales uno de ellos sería una réplica por si fallase el activo, teniendo el cometido de dirigir el tráfico web a uno de los servidores de aplicaciones según fuera su carga de trabajo. Entre los dos servidores de datos y los servidores de aplicaciones habría otro balanceador de carga, para equilibrar la carga de trabajo de los servidores de datos.

La elección de la arquitectura propuesta para el despliegue es la de dar la mejor respuesta teniendo el menor coste posible y a su vez ser escalable, puesto que los balanceadores de carga permiten acoplar más servidores si fueran necesarios en un futuro.

5.3. Modelo lógico de datos

Se diseña la base de datos partiendo del análisis realizado en el apartado modelo conceptual de datos (ver apartado 4.6).

La base de datos se va a implementar siguiendo una base relacional por contar con las siguientes características:

- Evitar duplicidad de los registros evitando así la mala interpretación, por el uso de una clave primaria.

- Integridad de los datos impidiendo entradas no válidas y la corrupción gracias a las claves foráneas.
- Además, posibilita representar relaciones reales entre entidades, así como extraer datos de distintas tablas como si fuera de una sola.

El modelo relacional es usado para representar el diseño de una base de datos de tipo relacional, donde la base de datos es considerada como un conjunto de relaciones. Las relaciones se representan a través de tablas formadas por filas compuestas por campos. Las claves primarias están representadas mediante una llave amarilla, si la llave es de color rojo y las claves foráneas mediante un rombo en color rojo o una llave de este mismo color.

En la siguiente figura puede observarse el modelo relacional (ver figura 14):

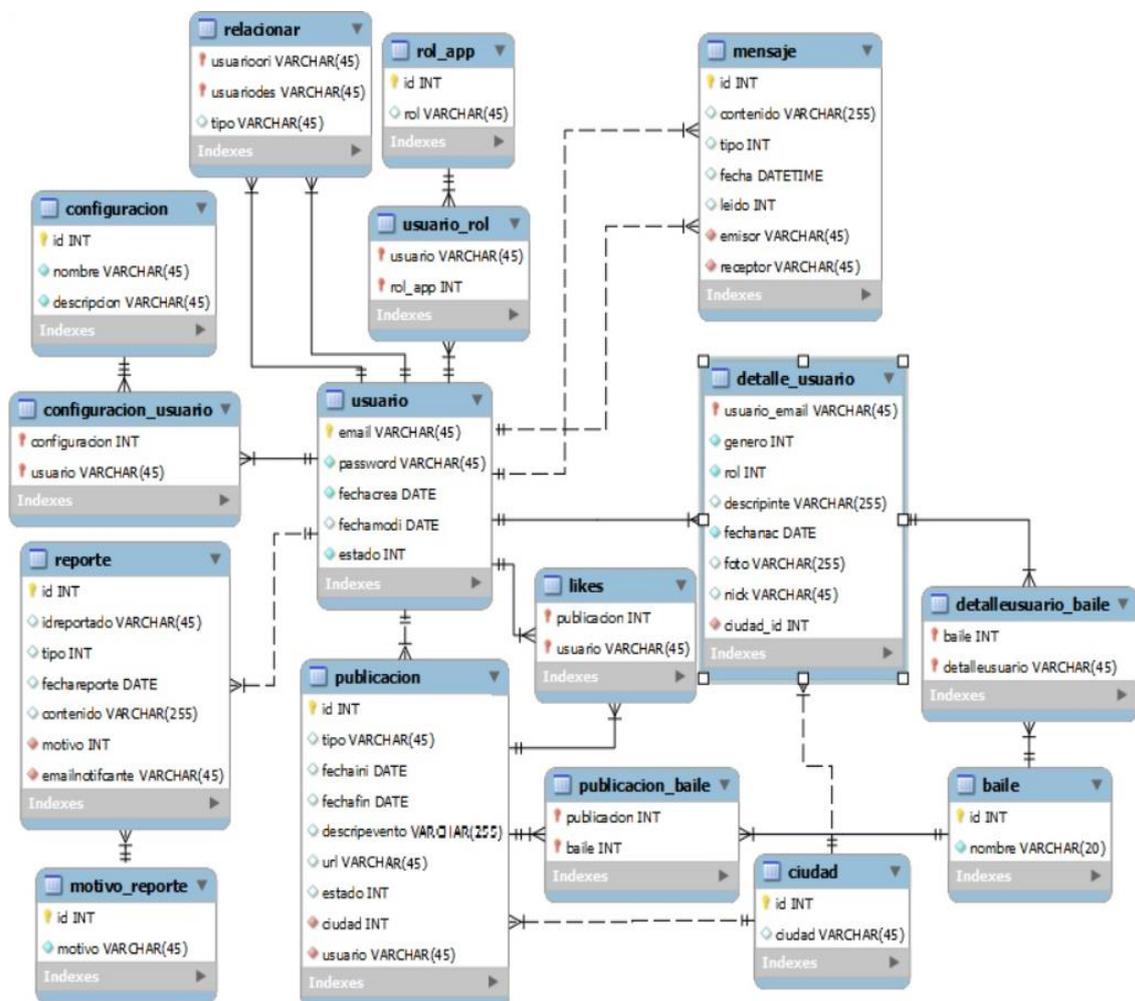


Figura 14. Modelo relacional.

5.4. Diagramas de clases

El diagrama de clases presenta todas las clases a implementar para su posterior codificación del proyecto. Dicho diagrama se va a presentar por partes en varias páginas debido a la cantidad de clases que componen ambos proyectos, tanto de *WebMVC* como de *ServiciosREST*.

A continuación, en la presentación las clases están agrupadas por paquetes, tal y como se va a implementar. Pese a que en el apartado anterior (ver apartado 5.1.) donde aparece la figura de la arquitectura lógica, las clases del email estén donde los clientes por similitud, en realidad va a estar en un paquete diferente llamado email y no clientes.

WebMVC: clases de configuración y de formato de la aplicación web

Clase con la configuración y clase formateadora de las fechas.

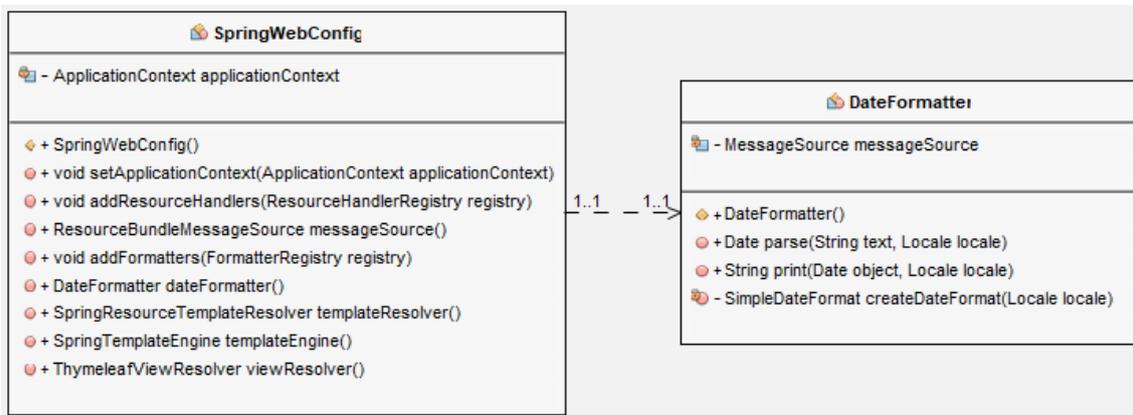


Figura 15. Clases de los paquetes webmvc.config y webmvc.formatter.

WebMVC: clases de envío de emails de la aplicación web

Clases para el envío del email mediante el servicio de SMTP de Google.

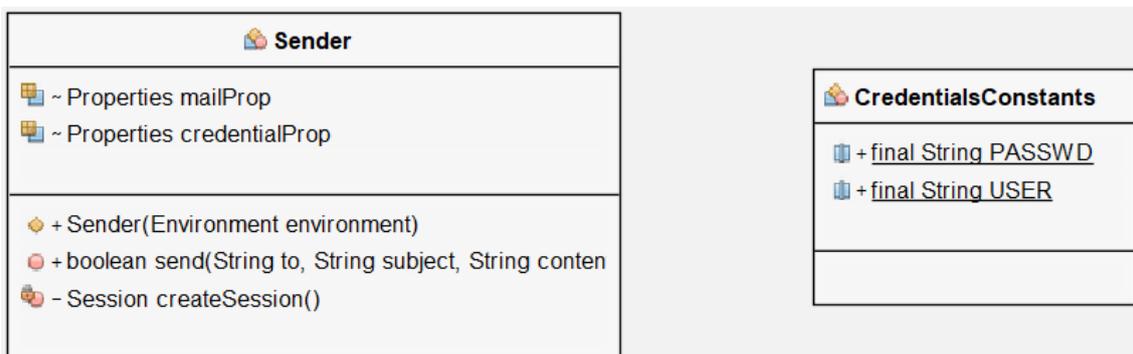


Figura 16. Clases del paquete webmvc.email.

WebMVC: clases controladoras de la aplicación web

Clases para manejar la visualización por pantalla para los usuarios.



Figura 17. Clases del paquete webmvc.controller 1.

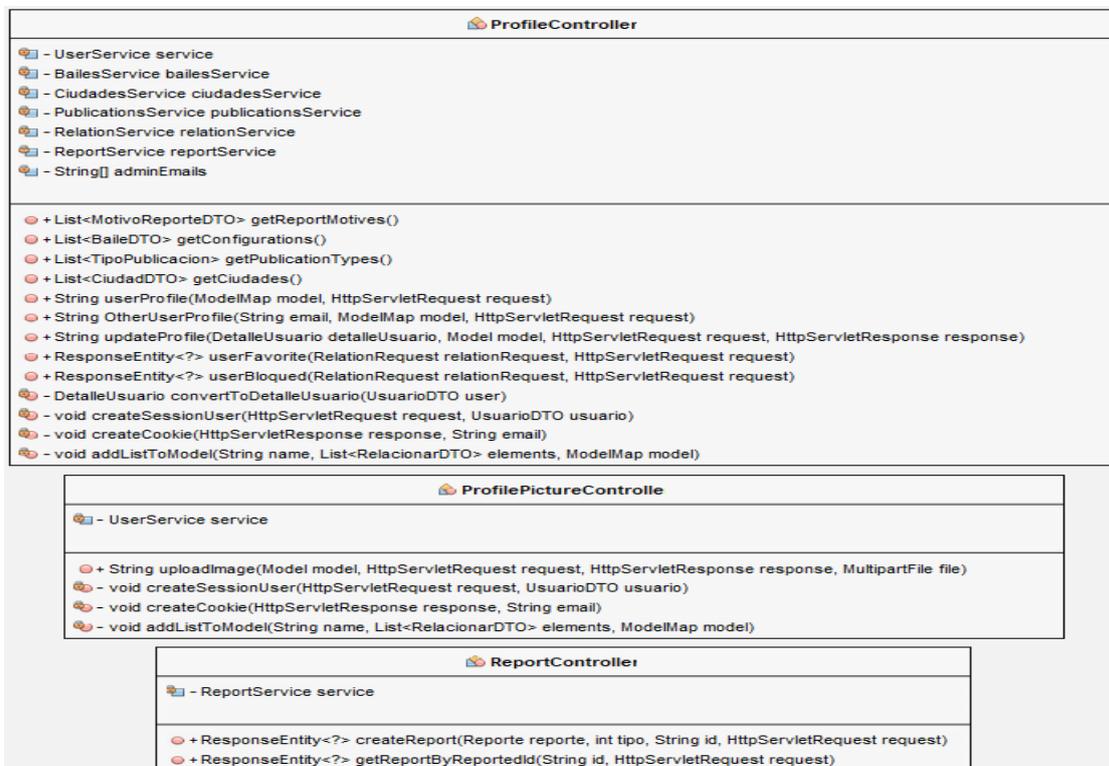


Figura 18. Clases del paquete webmvc.controller 2.

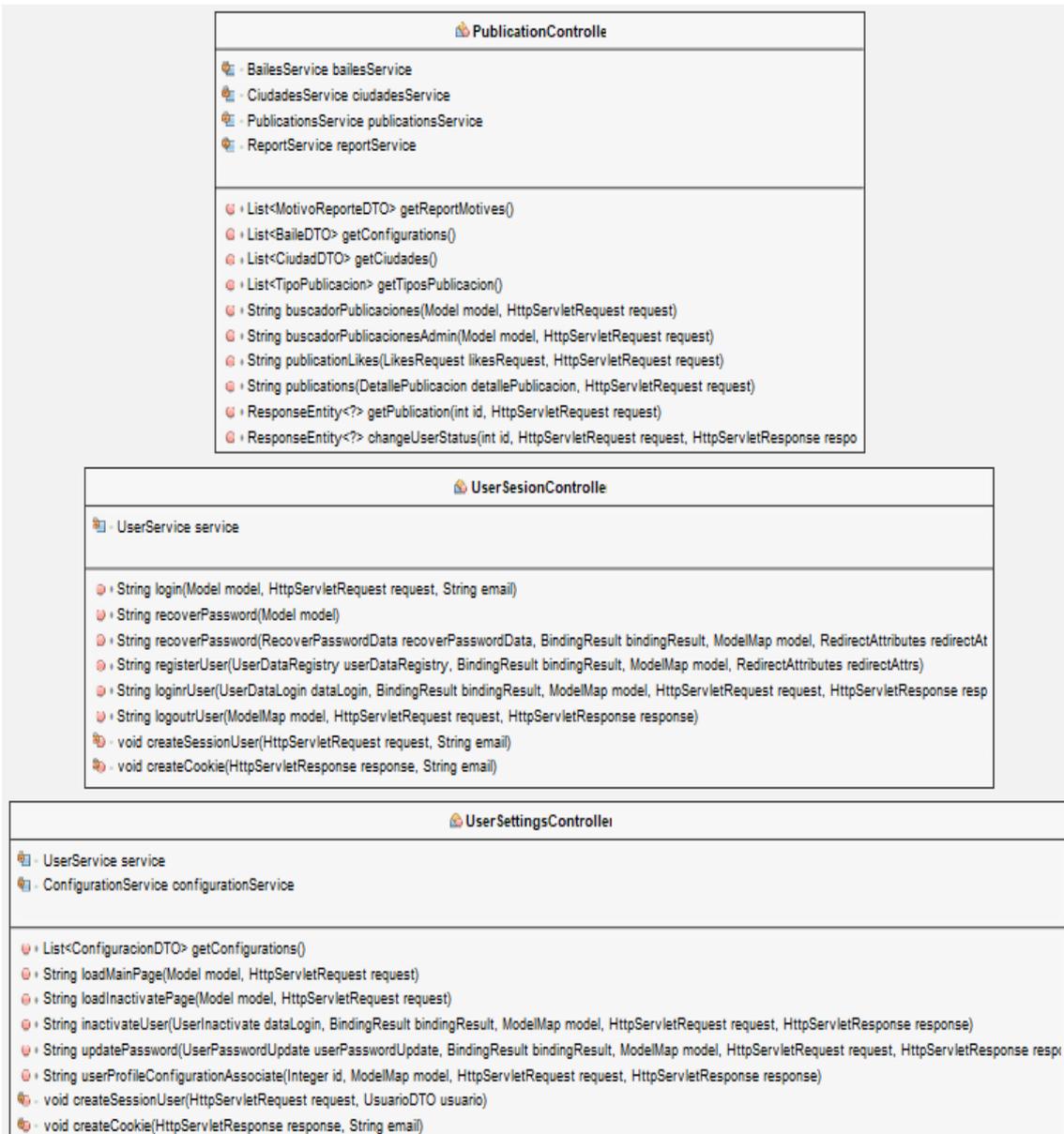


Figura 19. Clases del paquete webmvc.controller 3.

📁 DetallePublicacio
<ul style="list-style-type: none"> 🔒 - final long serialVersionUID 🔒 - Integer id 🔒 - Integer tipo 🔒 - Integer ciudad 🔒 - String fechainicio 🔒 - String fechafin 🔒 - Collection<Integer> bailes 🔒 - String url 🔒 - String describevento
<ul style="list-style-type: none"> 🔹 +DetallePublicacion() 🔹 +DetallePublicacion(Integer id, Integer tipo, Integer ciudad, String fechainicio, String fechafin, Collection<Integer> bailes, String url, String describevento) 🔴 +Integer getId() 🔴 +void setId(Integer id) 🔴 +Integer getTipo() 🔴 +void setTipo(Integer tipo) 🔴 +Integer getCiudad() 🔴 +void setCiudad(Integer ciudad) 🔴 +String getFechainicio() 🔴 +void setFechainicio(String fechainicio) 🔴 +String getFechafin() 🔴 +void setFechafin(String fechafin) 🔴 +Collection<Integer> getBailes() 🔴 +void setBailes(Collection<Integer> bailes) 🔴 +String getUrl() 🔴 +void setUrl(String url) 🔴 +String getDescribevento() 🔴 +void setDescribevento(String describevento) 🔴 +int hashCode() 🔴 +boolean equals(Object obj)

Figura 20. Clases del paquete webmvc.controller.bean 1.

📁 DetalleUsuari
<ul style="list-style-type: none"> 🔒 - final long serialVersionUID 🔒 - String usuarioEmail 🔒 - int genero 🔒 - int rol 🔒 - String descripinte 🔒 - String fechanac 🔒 - String nick 🔒 - Collection<String> bailes 🔒 - Integer ciudad
<ul style="list-style-type: none"> 🔹 +DetalleUsuario() 🔹 +DetalleUsuario(String usuarioEmail, int genero, int rol, String descripinte, String fechanac, String nick, Collection<String> bailes, Integer ciudad) 🔴 +String getUsuarioEmail() 🔴 +void setUsuarioEmail(String usuarioEmail) 🔴 +int getGenero() 🔴 +void setGenero(int genero) 🔴 +int getRol() 🔴 +void setRol(int rol) 🔴 +String getDescripinte() 🔴 +void setDescripinte(String descripinte) 🔴 +String getFechanac() 🔴 +void setFechanac(String fechanac) 🔴 +String getNick() 🔴 +void setNick(String nick) 🔴 +Collection<String> getBailes() 🔴 +void setBailes(Collection<String> bailes) 🔴 +Integer getCiudad() 🔴 +void setCiudad(Integer ciudad) 🔴 +int hashCode() 🔴 +boolean equals(Object obj)

Figura 21. Clases del paquete webmvc.controller.bean 2.

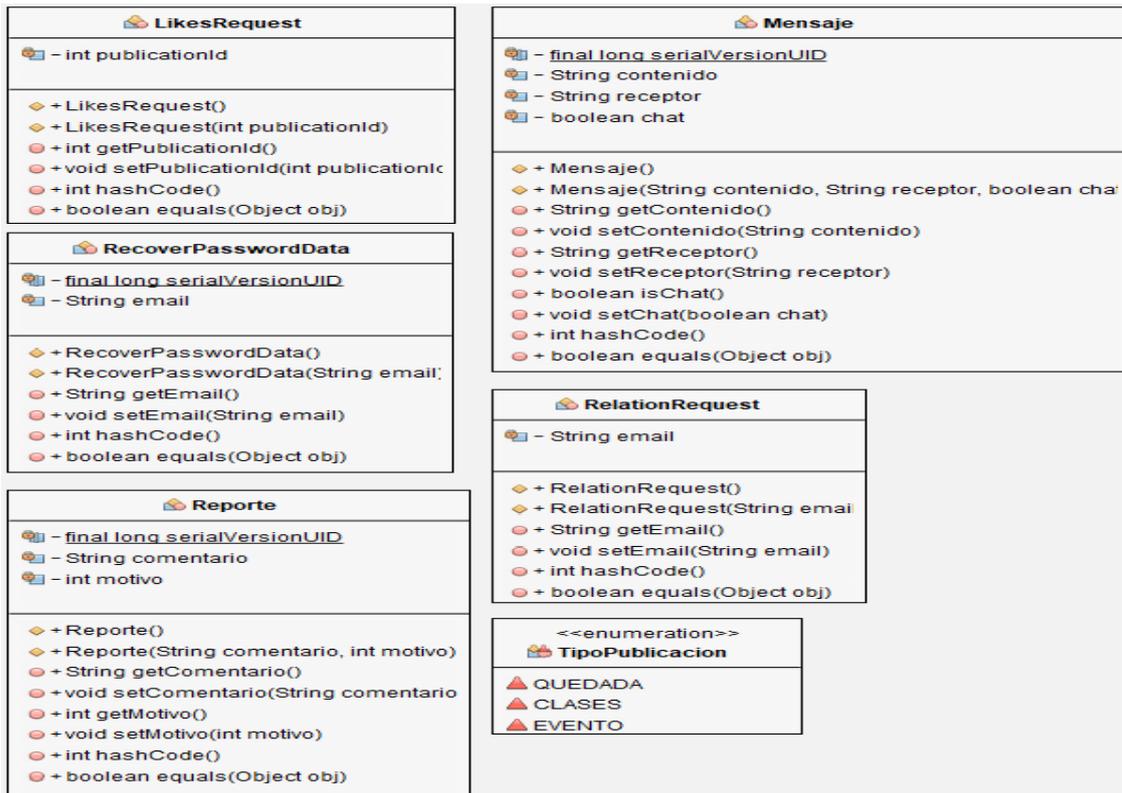


Figura 22. Clases del paquete webmvc.controller.bean 3.

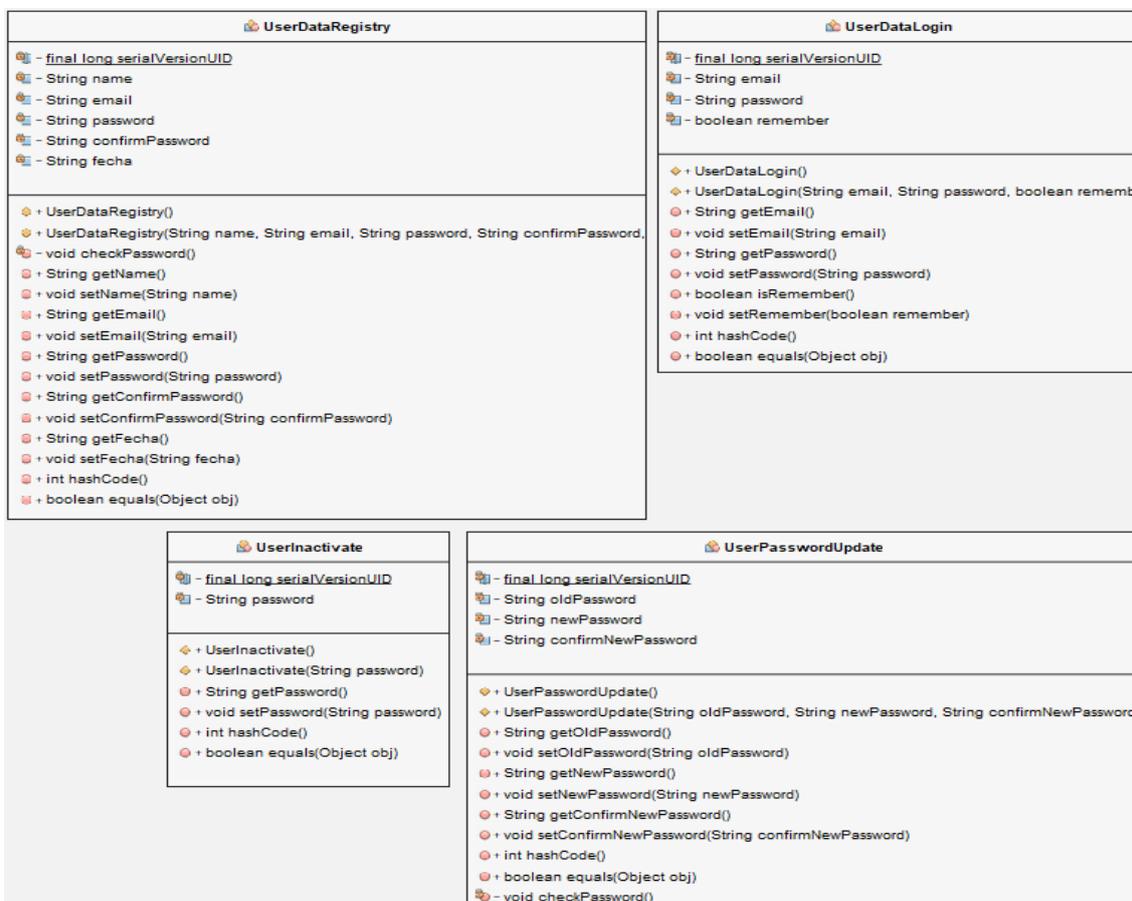


Figura 23. Clases del paquete webmvc.controller.bean 4.

WebMVC: clases servicio de la aplicación web

Clases para manejar la lógica de la aplicación web.

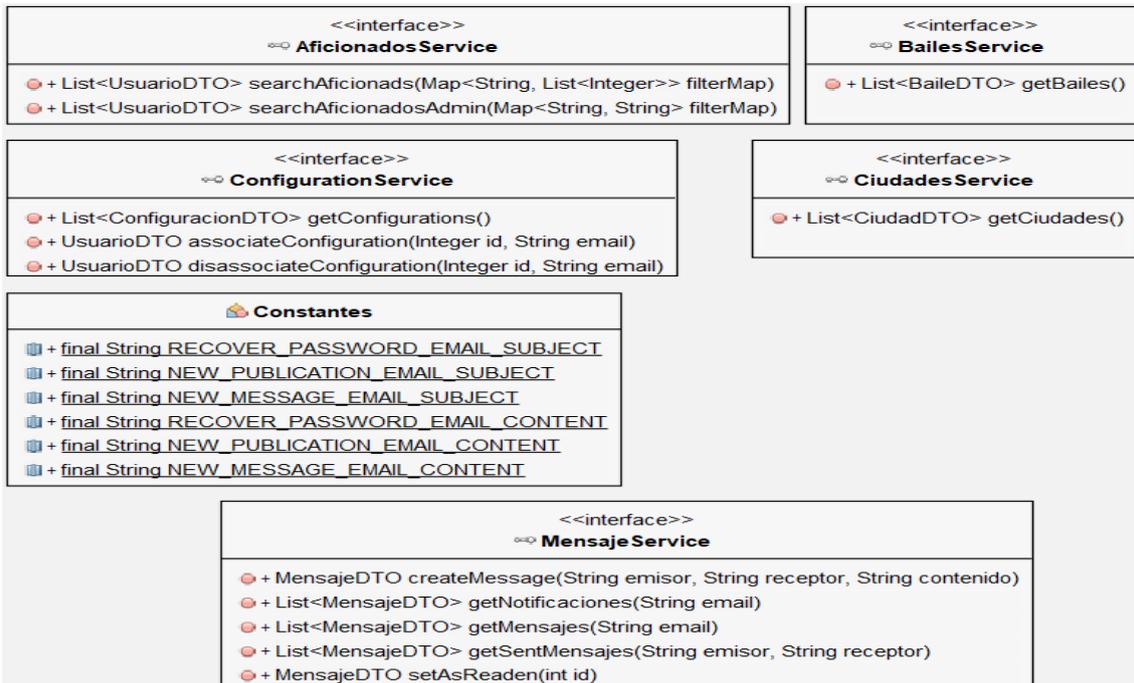


Figura 24. Clases del paquete webmvc.service 1.

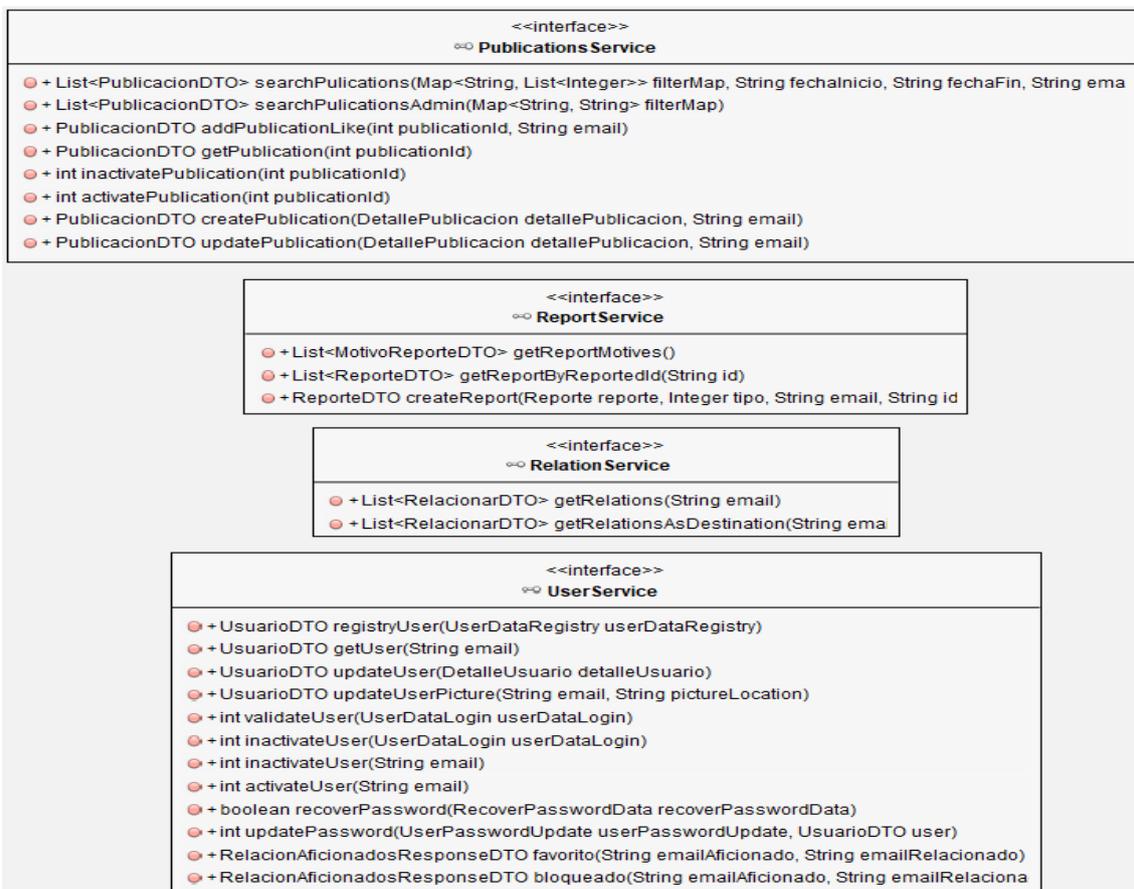


Figura 25. Clases del paquete webmvc.service 2.

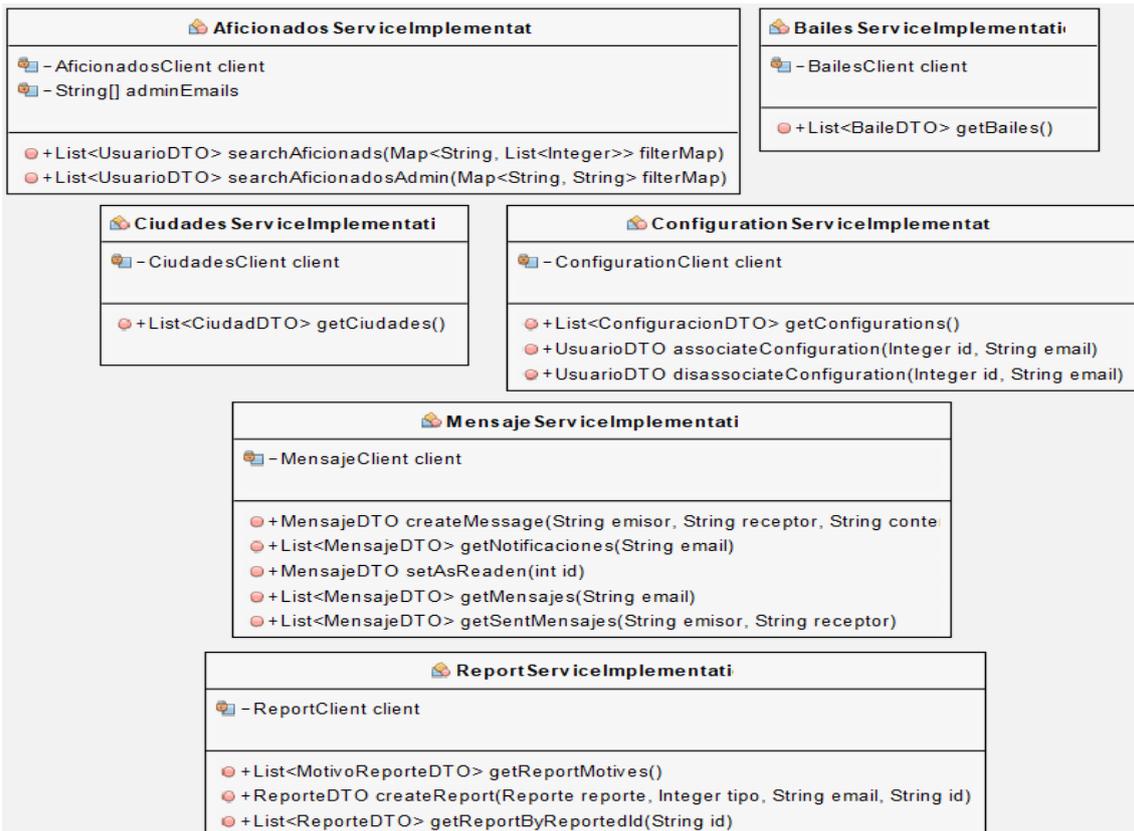


Figura 26. Clases del paquete webmvc.service.impl 1.

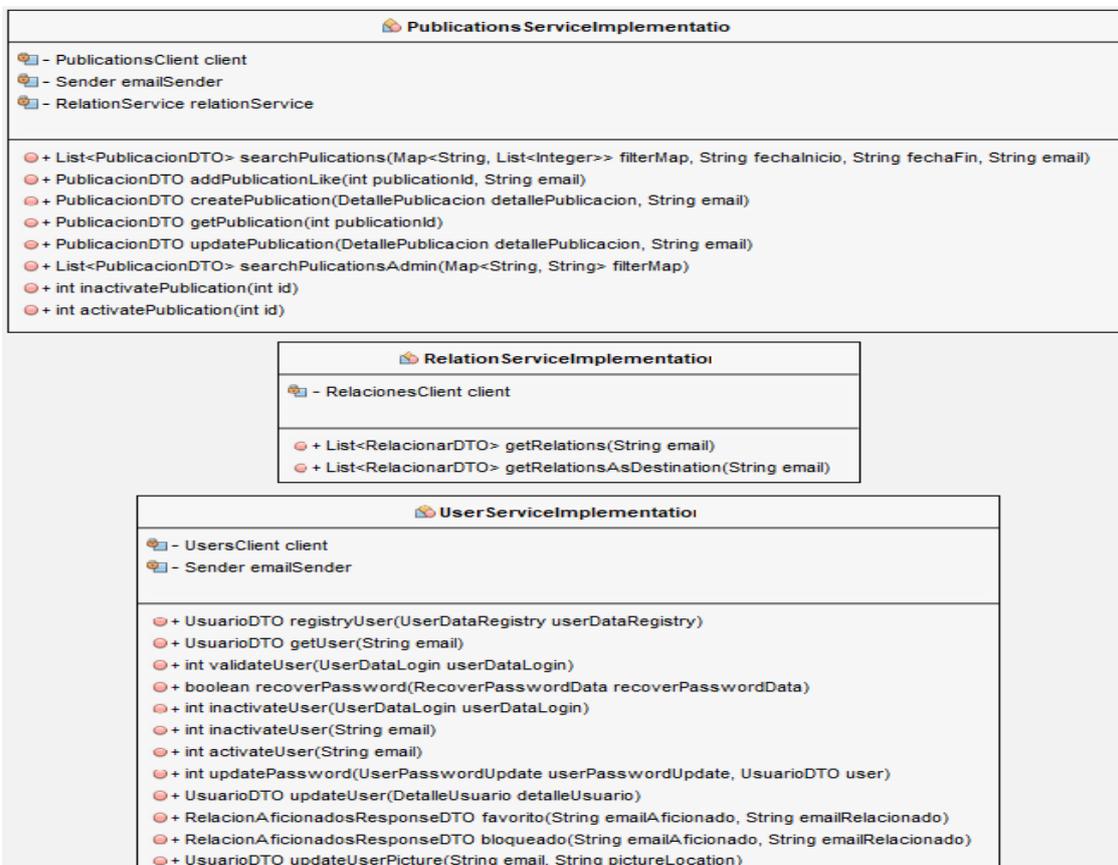


Figura 27. Clases del paquete webmvc.service.impl 2.

WebMVC: clases cliente de la aplicación web

Clase para manejar el acceso a los servicios.

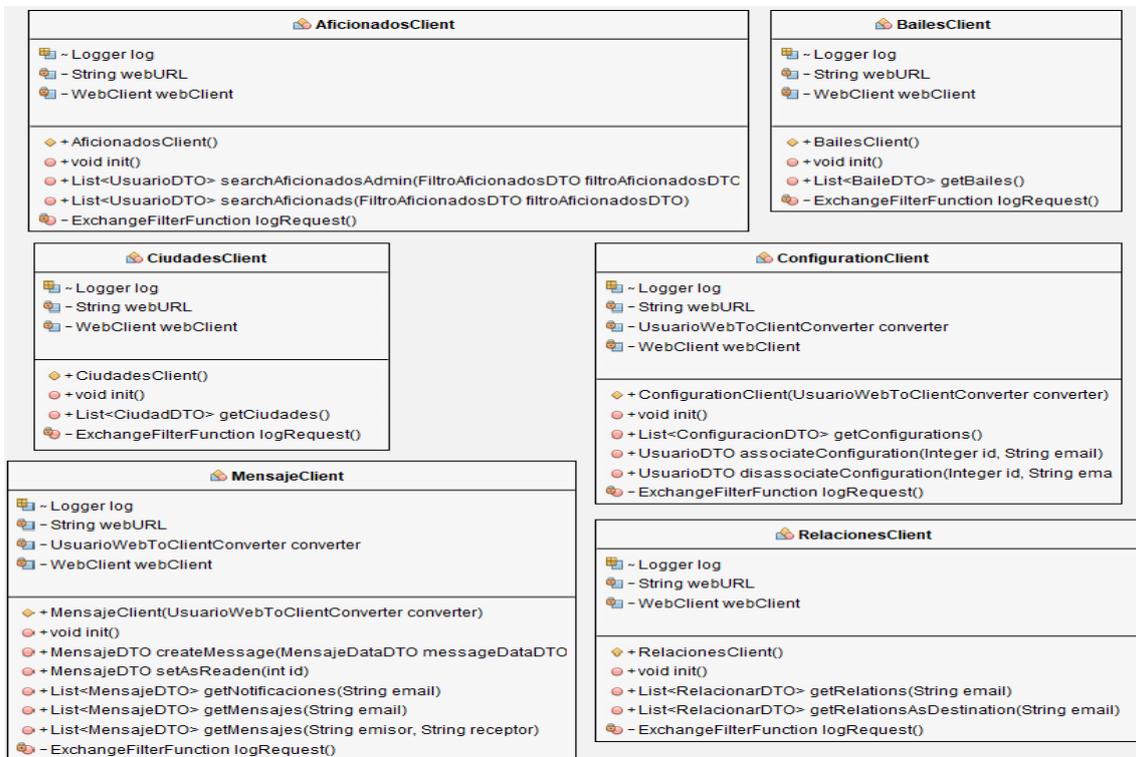


Figura 28. Clases del paquete webmvc.clientes 1.

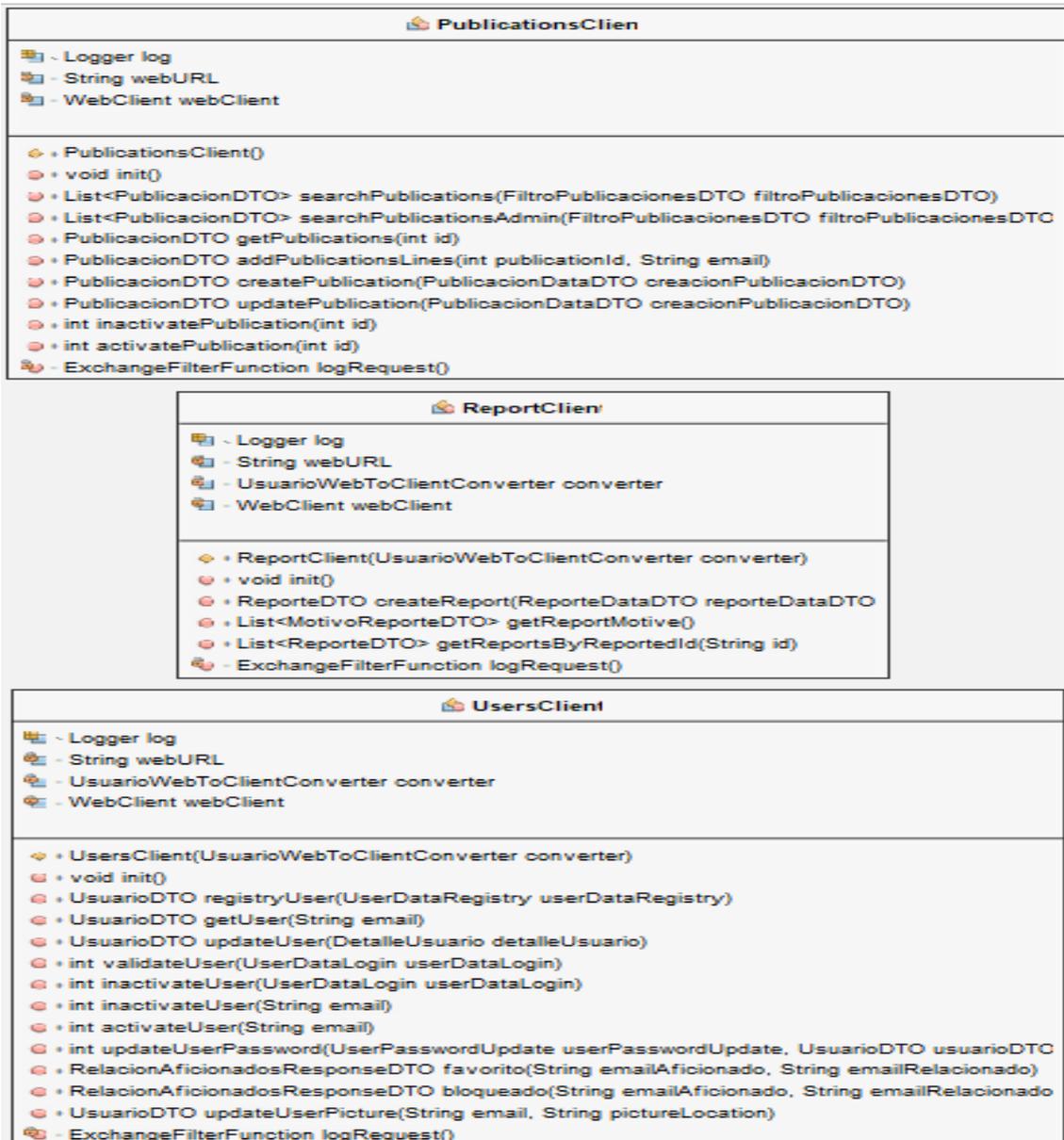


Figura 29. Clases del paquete webmvc.clientes 2.

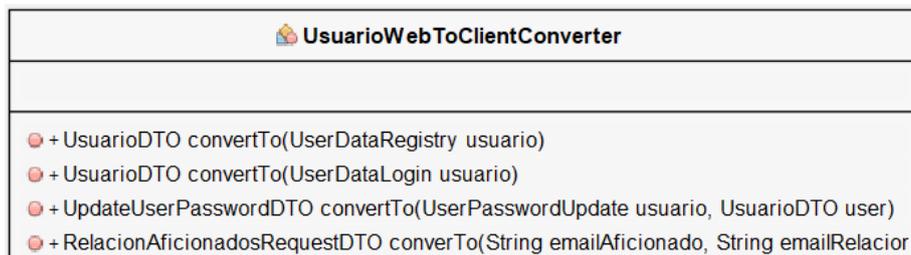


Figura 30. Clases del paquete webmvc.converter.

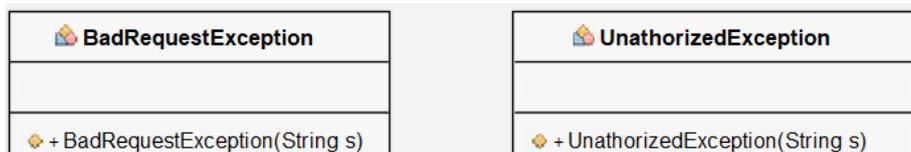


Figura 31. Clases del paquete webmvc.exceptions.

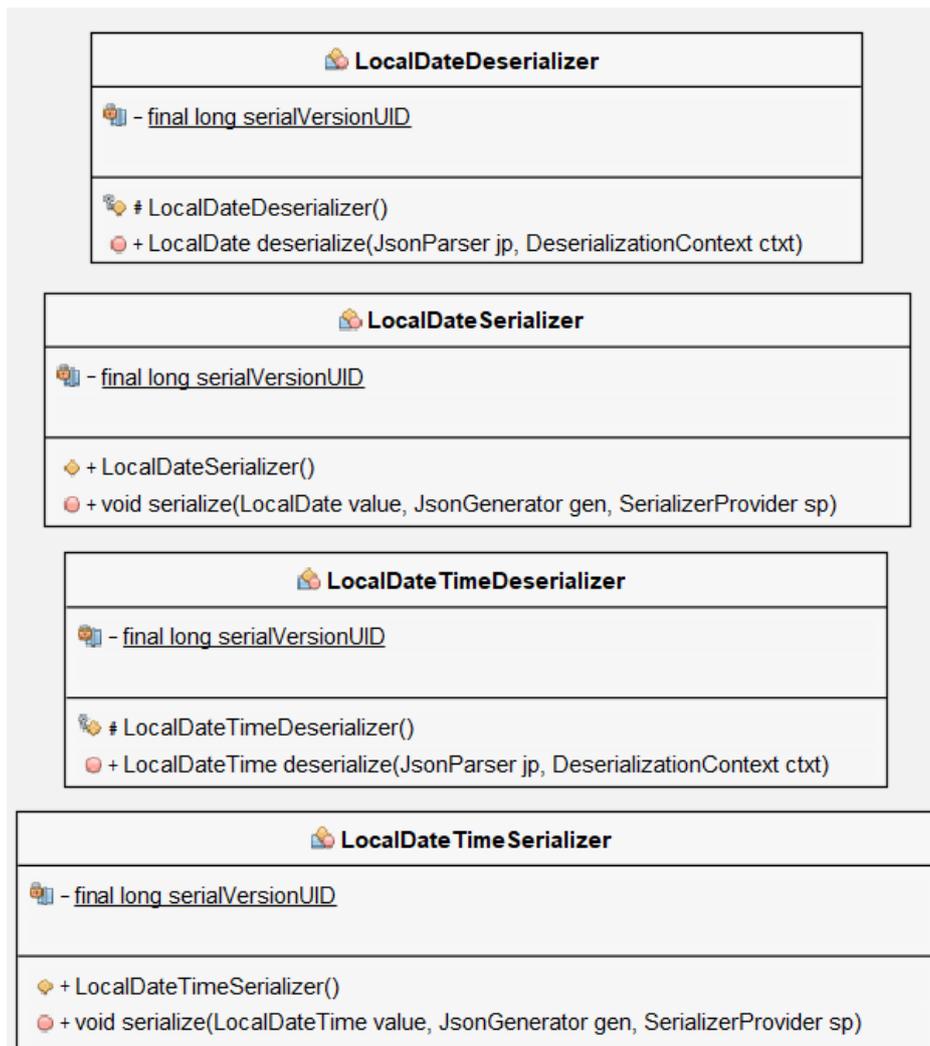


Figura 32. Clases del paquete `webmvc.clientes.formatters`.

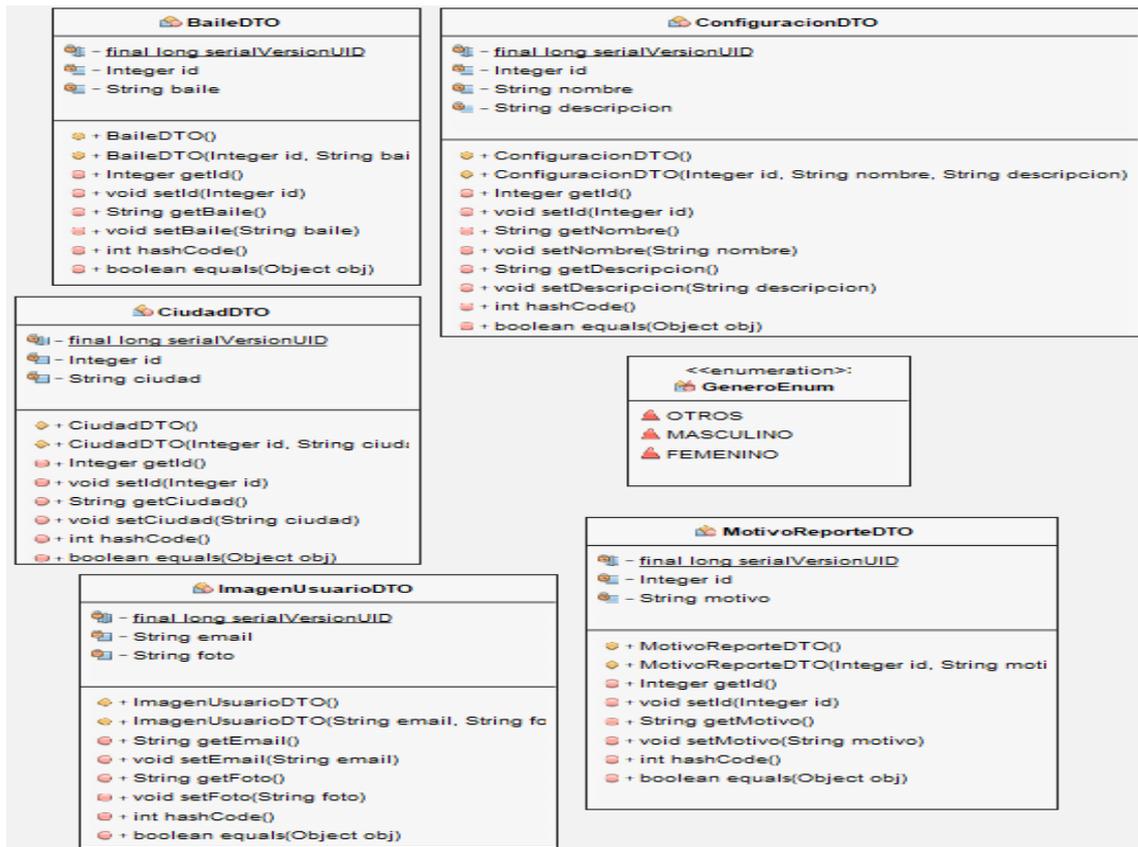


Figura 33. Clases del paquete webmvc.clientes.dto 1.

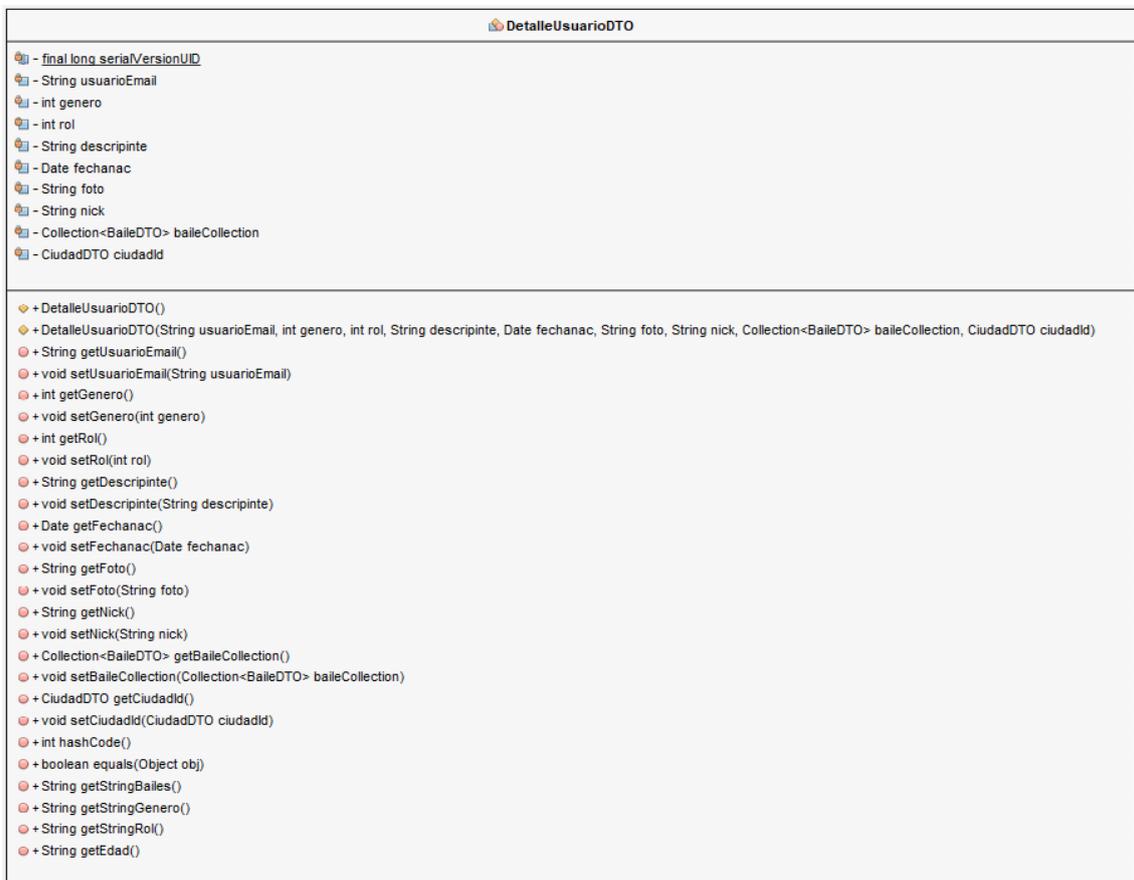


Figura 34. Clases del paquete webmvc.clientes.dto 2.

FiltroAficionadosDTO
<ul style="list-style-type: none"> Integer cityId Integer generoId Integer rolId List<Integer> dancesId String texto boolean isReportado boolean isInactivo String motivoReporte
<ul style="list-style-type: none"> + FiltroAficionadosDTO() + FiltroAficionadosDTO(Integer cityId, Integer generoId, Integer rolId, List<Integer> dancesId, String texto, boolean isReportado, boolean isInactivo, String motivoReporte) + Integer getCityId() + void setCityId(Integer cityId) + Integer getGeneroId() + void setGeneroId(Integer generoId) + Integer getRolId() + void setRolId(Integer rolId) + List<Integer> getDancesId() + void setDancesId(List<Integer> dancesId) + String getTexto() + void setTexto(String texto) + boolean isReportado() + void setReportado(boolean isReportado) + boolean isInactivo() + void setInactivo(boolean isInactivo) + String getMotivoReporte() + void setMotivoReporte(String motivoReporte) + int hashCode() + boolean equals(Object obj)

Figura 35. Clases del paquete webmvc.clientes.dto 3.

FiltroPublicacionesDTO
<ul style="list-style-type: none"> Integer cityId Integer typeId List<Integer> dancesId String startDate String endDate String email String texto boolean isReportado boolean isInactivo String motivoReporte
<ul style="list-style-type: none"> + FiltroPublicacionesDTO() + FiltroPublicacionesDTO(Integer cityId, Integer typeId, List<Integer> dancesId, String startDate, String endDate, String email, String texto, boolean isReportado, boolean isInactivo, String motivoReporte) + Integer getCityId() + void setCityId(Integer cityId) + Integer getTypeId() + void setTypeId(Integer typeId) + List<Integer> getDancesId() + void setDancesId(List<Integer> dancesId) + String getStartDate() + void setStartDate(String startDate) + String getEndDate() + void setEndDate(String endDate) + String getEmail() + void setEmail(String email) + String getTexto() + void setTexto(String texto) + boolean isReportado() + void setReportado(boolean isReportado) + boolean isInactivo() + void setInactivo(boolean isInactivo) + String getMotivoReporte() + void setMotivoReporte(String motivoReporte) + int hashCode() + boolean equals(Object obj)

Figura 36. Clases del paquete webmvc.clientes.dto 4.

MensajeDataDTC
<ul style="list-style-type: none"> 🔒 - final long serialVersionUID 🔒 - Integer id 🔒 - String contenido 🔒 - Integer tipo 🔒 - LocalDateTime fecha 🔒 - Integer leído 🔒 - String emisor 🔒 - String receptor
<ul style="list-style-type: none"> 🔑 + MensajeDataDTO() 🔑 + MensajeDataDTO(Integer id, String contenido, Integer tipo, LocalDateTime fecha, Integer leído, String emisor, String rec) 🔑 + Integer getId() 🔑 + void setId(Integer id) 🔑 + String getContenido() 🔑 + void setContenido(String contenido) 🔑 + Integer getTipo() 🔑 + void setTipo(Integer tipo) 🔑 + LocalDateTime getFecha() 🔑 + void setFecha(LocalDateTime fecha) 🔑 + Integer getLeído() 🔑 + void setLeído(Integer leído) 🔑 + String getEmisor() 🔑 + void setEmisor(String emisor) 🔑 + String getReceptor() 🔑 + void setReceptor(String receptor) 🔑 + int hashCode() 🔑 + boolean equals(Object obj)

Figura 37. Clases del paquete webmvc.clientes.dto 5.

MensajeDTC
<ul style="list-style-type: none"> 🔒 - final long serialVersionUID 🔒 - Integer id 🔒 - String contenido 🔒 - Integer tipo 🔒 - LocalDateTime fecha 🔒 - String fechaFormateada 🔒 - Integer leído 🔒 - UsuarioDTO emisor 🔒 - UsuarioDTO receptor
<ul style="list-style-type: none"> 🔑 + MensajeDTO() 🔑 + MensajeDTO(Integer id, String contenido, Integer tipo, LocalDateTime fecha, Integer leído, UsuarioDTO emisor, UsuarioDTO receptor) 🔑 + Integer getId() 🔑 + void setId(Integer id) 🔑 + String getContenido() 🔑 + void setContenido(String contenido) 🔑 + Integer getTipo() 🔑 + void setTipo(Integer tipo) 🔑 + LocalDateTime getFecha() 🔑 + void setFecha(LocalDateTime fecha) 🔑 + Integer getLeído() 🔑 + void setLeído(Integer leído) 🔑 + UsuarioDTO getEmisor() 🔑 + void setEmisor(UsuarioDTO emisor) 🔑 + UsuarioDTO getReceptor() 🔑 + void setReceptor(UsuarioDTO receptor) 🔑 + String getFechaFormateada() 🔑 + int hashCode() 🔑 + boolean equals(Object obj)

Figura 38. Clases del paquete webmvc.clientes.dto 6.

PublicacionDataDTO

- final long serialVersionUID
- Integer id
- Integer tipo
- Integer ciudad
- String fechainicio
- String fechafin
- Collection<Integer> bailes
- String url
- String describevento
- String email

- + PublicacionDataDTO()
- + PublicacionDataDTO(Integer id, Integer tipo, Integer ciudad, String fechainicio, String fechafin, Collection<Integer> bailes, String url, String describevento, String email)
- + Integer getId()
- + void setId(Integer id)
- + Integer getTipo()
- + void setTipo(Integer tipo)
- + Integer getCiudad()
- + void setCiudad(Integer ciudad)
- + String getFechainicio()
- + void setFechainicio(String fechainicio)
- + String getFechafin()
- + void setFechafin(String fechafin)
- + Collection<Integer> getBailes()
- + void setBailes(Collection<Integer> bailes)
- + String getUrl()
- + void setUrl(String url)
- + String getDescribevento()
- + void setDescribevento(String describevento)
- + String getEmail()
- + void setEmail(String email)
- + int hashCode()
- + boolean equals(Object obj)

Figura 39. Clases del paquete webmvc.clientes.dto 7.

PublicacionDTO

- final long serialVersionUID
- Integer id
- String tipo
- LocalDate fechaini
- LocalDate fechafin
- String describevento
- String url
- Integer estado
- Collection<BaileDTO> baileCollection
- Collection<UsuarioDTO> usuarioCollection
- CiudadDTO ciudad
- UsuarioDTO usuario

- + PublicacionDTO()
- + PublicacionDTO(Integer id, String tipo, LocalDate fechaini, LocalDate fechafin, String describevento, String url, Integer estado, Collection<BaileDTO> baileCollection, Collection<UsuarioDTO> usuarioCollection, CiudadDTO ciudad, U
- + Integer getId()
- + void setId(Integer id)
- + String getTipo()
- + void setTipo(String tipo)
- + LocalDate getFechaini()
- + void setFechaini(LocalDate fechaini)
- + LocalDate getFechafin()
- + void setFechafin(LocalDate fechafin)
- + String getDescribevento()
- + void setDescribevento(String describevento)
- + String getUrl()
- + void setUrl(String url)
- + Integer getEstado()
- + void setEstado(Integer estado)
- + Collection<BaileDTO> getBaileCollection()
- + void setBaileCollection(Collection<BaileDTO> baileCollection)
- + Collection<UsuarioDTO> getUsuarioCollection()
- + void setUsuarioCollection(Collection<UsuarioDTO> usuarioCollection)
- + CiudadDTO getCiudad()
- + void setCiudad(CiudadDTO ciudad)
- + UsuarioDTO getUsuario()
- + void setUsuario(UsuarioDTO usuario)
- + int hashCode()
- + boolean equals(Object obj)
- + String getStringDate()
- + int getTotalLikes()
- + boolean isActive()

Figura 40. Clases del paquete webmvc.clientes.dto 8.

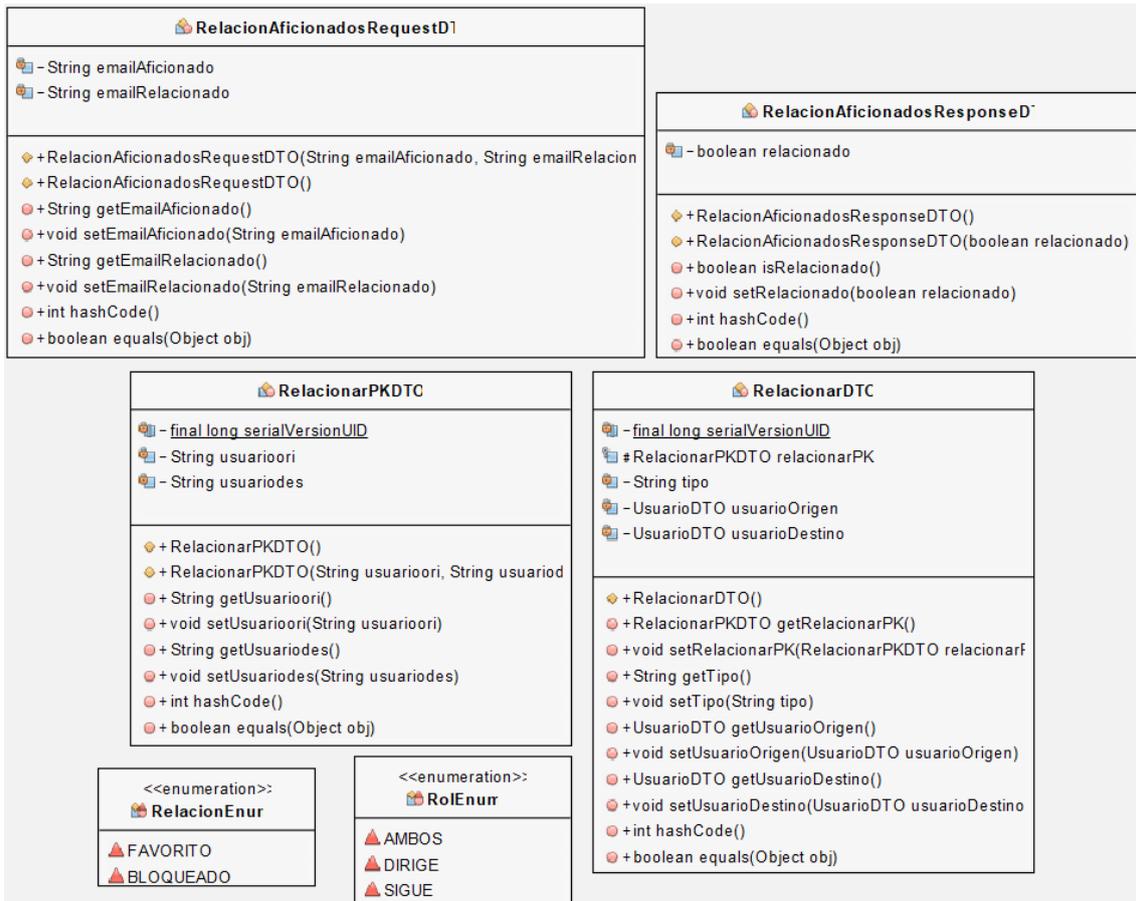


Figura 41. Clases del paquete webmvc.clientes.dto 9.

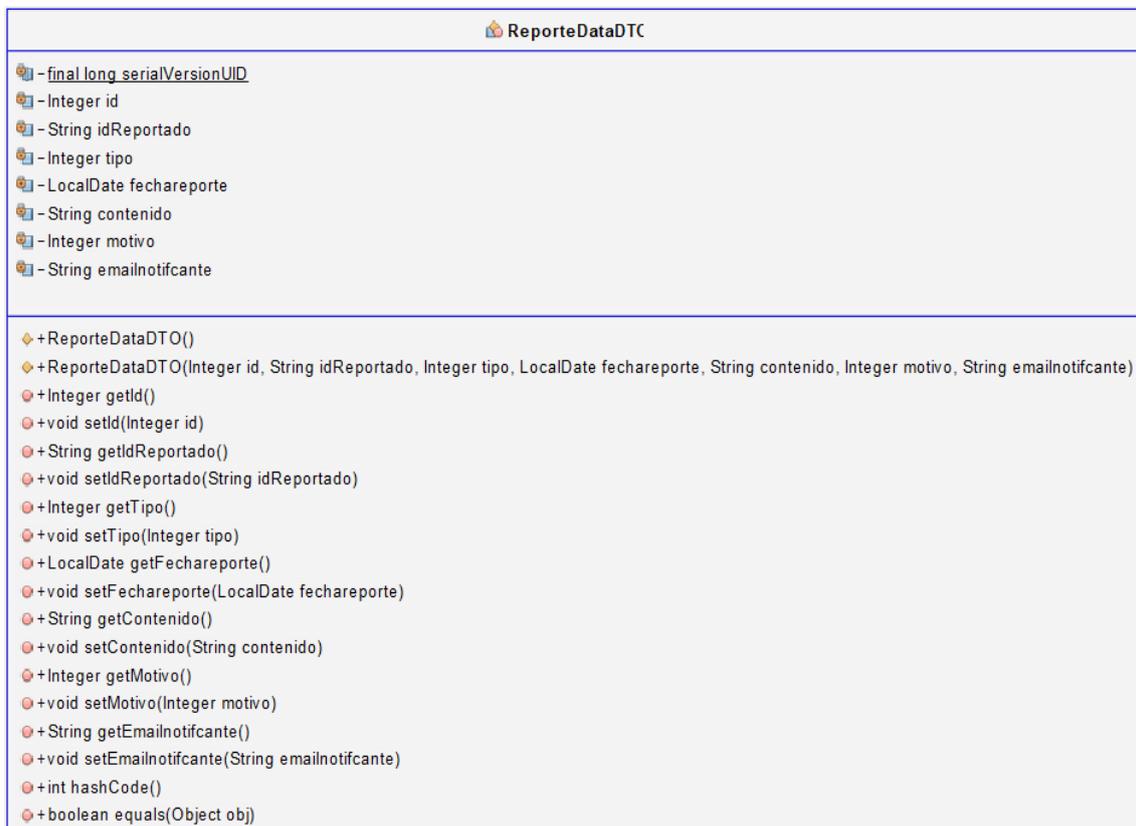


Figura 42. Clases del paquete webmvc.clientes.dto 10.

ReporteDTC
<ul style="list-style-type: none"> - final long serialVersionUID - Integer id - Integer tipo - String idReportado - LocalDate fechareporte - String contenido - MotivoReporteDTO motivo - UsuarioDTO emailnotificante - String aliasReportado
<ul style="list-style-type: none"> + ReporteDTO() + ReporteDTO(Integer id, Integer tipo, String idReportado, LocalDate fechareporte, String contenido, MotivoReporteDTO motivo, UsuarioDTO emailnotificante, String aliasReportado) + Integer getId() + void setId(Integer id) + Integer getTipo() + void setTipo(Integer tipo) + String getIdReportado() + void setIdReportado(String idReportado) + LocalDate getFechaReporte() + void setFechaReporte(LocalDate fechareporte) + String getContenido() + void setContenido(String contenido) + MotivoReporteDTO getMotivo() + void setMotivo(MotivoReporteDTO motivo) + UsuarioDTO getEmailnotificante() + void setEmailnotificante(UsuarioDTO emailnotificante) + String getAliasReportado() + void setAliasReportado(String aliasReportado) + int hashCode() + boolean equals(Object obj)

Figura 43. Clases del paquete webmvc.clientes.dto 11.

UsuarioDTO
<ul style="list-style-type: none"> - final long serialVersionUID - String email - String password - LocalDate fechacrea - LocalDate fechamodi - int estado - Collection<RolAppDTO> rolAppCollection - Collection<ConfiguracionDTO> configuracionCollection - DetalleUsuarioDTO detalleUsuario
<ul style="list-style-type: none"> + UsuarioDTO() + UsuarioDTO(String email, String password, LocalDate fechacrea, LocalDate fechamodi, int estado, Collection<RolAppDTO> rolAppCollection, Collection<ConfiguracionDTO> configuracionCollection, DetalleUsuarioDTO detalleUsuario) + String getEmail() + void setEmail(String email) + String getPassword() + void setPassword(String password) + LocalDate getFechaCrea() + void setFechaCrea(LocalDate fechacrea) + LocalDate getFechaModi() + void setFechaModi(LocalDate fechamodi) + int getEstado() + void setEstado(int estado) + Collection<RolAppDTO> getRolAppCollection() + void setRolAppCollection(Collection<RolAppDTO> rolAppCollection) + Collection<ConfiguracionDTO> getConfiguracionCollection() + void setConfiguracionCollection(Collection<ConfiguracionDTO> configuracionCollection) + DetalleUsuarioDTO getDetalleUsuario() + void setDetalleUsuario(DetalleUsuarioDTO detalleUsuario) + int hashCode() + boolean equals(Object obj) + boolean isActive()

Figura 44. Clases del paquete webmvc.clientes.dto 12.



Figura 45. Clases del paquete webmvc.clientes.dto 13.

ServiciosREST: clases controladoras de los servicios

Clases para manejar la ApiREST de los servicios.

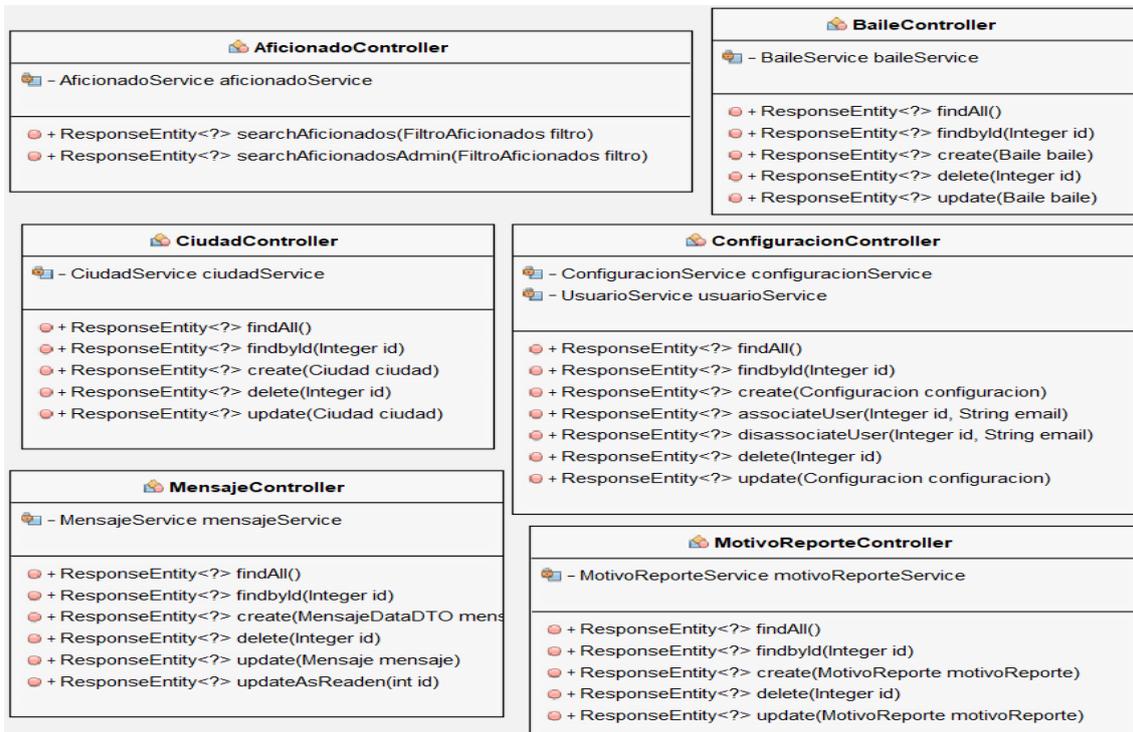


Figura 46. Clases del paquete serviciosrest.controller 1.

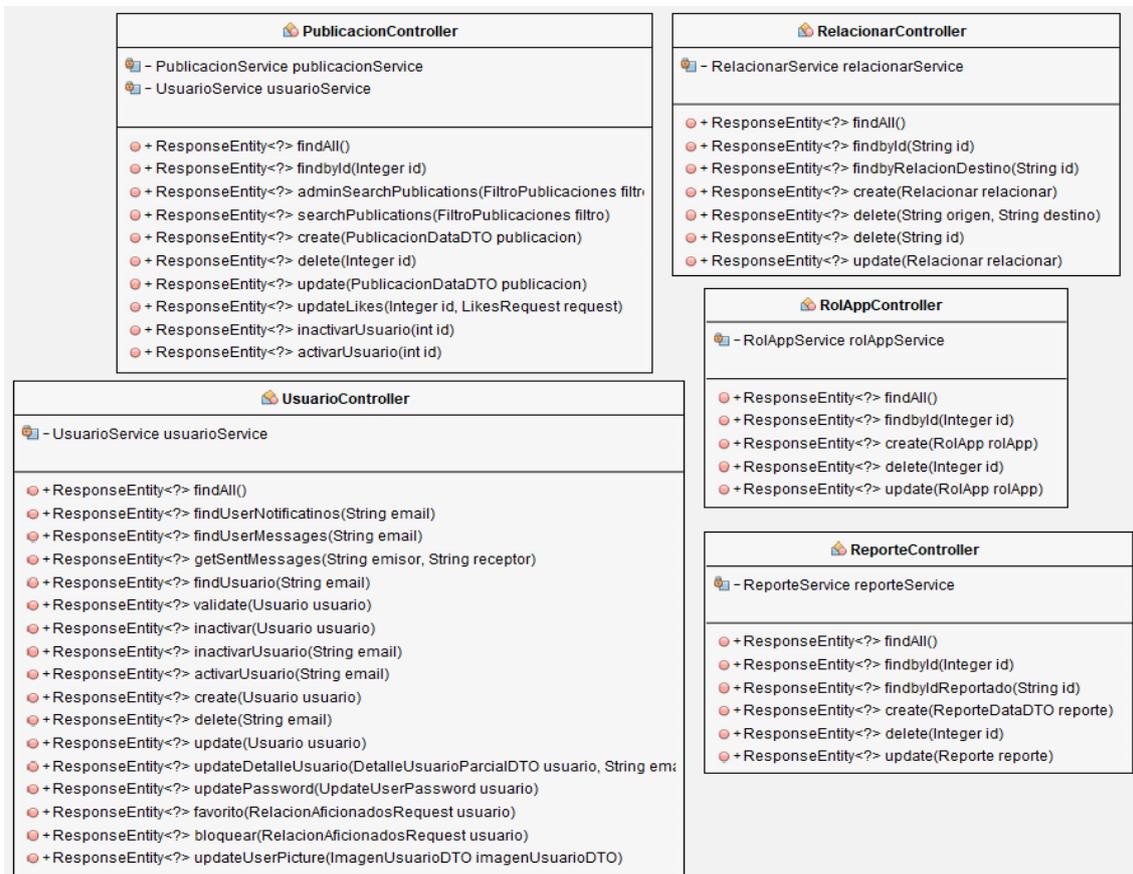


Figura 47. Clases del paquete serviciosrest.controller 2.

Constantes
final String ERROR_CREAR CONFIGURATION
final String ERROR_CREAR_USUARIO
final String ERROR_UPDATE_CONFIGURATION
final String ERROR_UPDATE_USER
final String ERROR_CONFIGURATIONS NOT FOUND
final String ERROR_CONFIGURATION_NOT_FOUND
final String CONFIGURATION_ALREADY_EXIST
final String CONFIGURATION NOT EXIST
final String ERROR_DELETING CONFIGURATION
final Object BAILES_NOT_FOUND
final Object BAILE_NOT_FOUND
final Object BAILE_ALREADY_EXIST
final Object USER_ALREADY_EXIST
final Object BAILE_NOT_EXIST
final Object USER NOT EXIST
final Object ERROR_DELETING DANCE
final Object ERROR_DELETING_USER
final Object CITIES_NOT_FOUND
final Object CITY NOT FOUND
final Object CITY_ALREADY_EXIST
final Object CITY_NOT_EXIST
final Object ERROR_DELETING_CITY
final Object MESSAGES_NOT_FOUND
final Object MESSAGE_NOT_FOUND
final Object MESSAGE_ALREADY_EXIST
final Object MESSAGE_NOT_EXIST
final Object REPORT_REASONS NOT FOUND
final Object REPORT_REASON NOT FOUND
final Object REPORT_REASON_ALREADY_EXIST
final Object REPORT_REASON_NOT_EXIST
final Object ERROR_DELETING_REPORT_REASON
final Object PUBLICATIONS NOT FOUND
final Object PUBLICATION_NOT_FOUND
final Object PUBLICATION_ALREADY_EXIST
final Object PUBLICATION NOT EXIST
final Object ERROR_DELETING PUBLICATION
final Object USERS_NOT_FOUND
final Object REPORTS_NOT_FOUND
final Object REPORT NOT FOUND
final Object REPORT_ALREADY_EXIST
final Object ERROR_DELETING_REPORT
final Object APP_ROLES_NOT_FOUND
final Object APP_ROLE NOT FOUND
final Object APP_ROLE_ALREADY_EXIST
final Object ERROR_DELETING_APP_ROLE
final Object REALTIONS_NOT_FOUND
final Object REALTION NOT FOUND
final Object RELATION NOT EXIST
final Object ERROR_DELETING_RELATION
final Object REALTION_NOT_EXIST
final Object RELATION_ALREADY_EXIST

Figura 48. Clases del paquete serviciosrest.controller 3.

🏠 FiltroAficionados
<ul style="list-style-type: none"> 📄 - Integer cityId 📄 - Integer generold 📄 - Integer rolld 📄 - List<Integer> dancesId 📄 - String texto 📄 - boolean isReportado 📄 - boolean isInactivo 📄 - String motivoReporte
<ul style="list-style-type: none"> 🔹 + FiltroAficionados() 🔹 + FiltroAficionados(Integer cityId, Integer generold, Integer rolld, List<Integer> dancesId, String texto, boolean isReportado, boolean isInactivo, String motivoReporte) 🔴 + Integer getCityId() 🔴 + void setCityId(Integer cityId) 🔴 + Integer getGenerold() 🔴 + void setGenerold(Integer generold) 🔴 + Integer getRolld() 🔴 + void setRolld(Integer rolld) 🔴 + List<Integer> getDancesId() 🔴 + void setDancesId(List<Integer> dancesId) 🔴 + String getTexto() 🔴 + void setTexto(String texto) 🔴 + boolean isReportado() 🔴 + void setReportado(boolean isReportado) 🔴 + boolean isInactivo() 🔴 + void setInactivo(boolean isInactivo) 🔴 + String getMotivoReporte() 🔴 + void setMotivoReporte(String motivoReporte) 🔴 + int hashCode() 🔴 + boolean equals(Object obj)

Figura 49. Clases del paquete serviciosrest.controller.bean 1.

🏠 FiltroPublicaciones
<ul style="list-style-type: none"> 📄 - Integer cityId 📄 - Integer typeld 📄 - Integer[] dancesId 📄 - String startDate 📄 - String endDate 📄 - String email 📄 - String texto 📄 - boolean isReportado 📄 - boolean isInactivo 📄 - String motivoReporte
<ul style="list-style-type: none"> 🔹 + FiltroPublicaciones() 🔹 + FiltroPublicaciones(Integer cityId, Integer typeld, Integer[] dancesId, String startDate, String endDate, String email, String texto, boolean isReportado, boolean isInactivo, String motivoReporte) 🔴 + Integer getCityId() 🔴 + void setCityId(Integer cityId) 🔴 + Integer getTypeld() 🔴 + void setTypeld(Integer typeld) 🔴 + Integer[] getDancesId() 🔴 + void setDancesId(Integer[] dancesId) 🔴 + String getStartDate() 🔴 + void setStartDate(String startDate) 🔴 + String getEndDate() 🔴 + void setEndDate(String endDate) 🔴 + String getEmail() 🔴 + void setEmail(String email) 🔴 + String getTexto() 🔴 + void setTexto(String texto) 🔴 + boolean isReportado() 🔴 + void setReportado(boolean isReportado) 🔴 + boolean isInactivo() 🔴 + void setInactivo(boolean isInactivo) 🔴 + String getMotivoReporte() 🔴 + void setMotivoReporte(String motivoReporte) 🔴 + int hashCode() 🔴 + boolean equals(Object obj)

Figura 50. Clases del paquete serviciosrest.controller.bean 2.

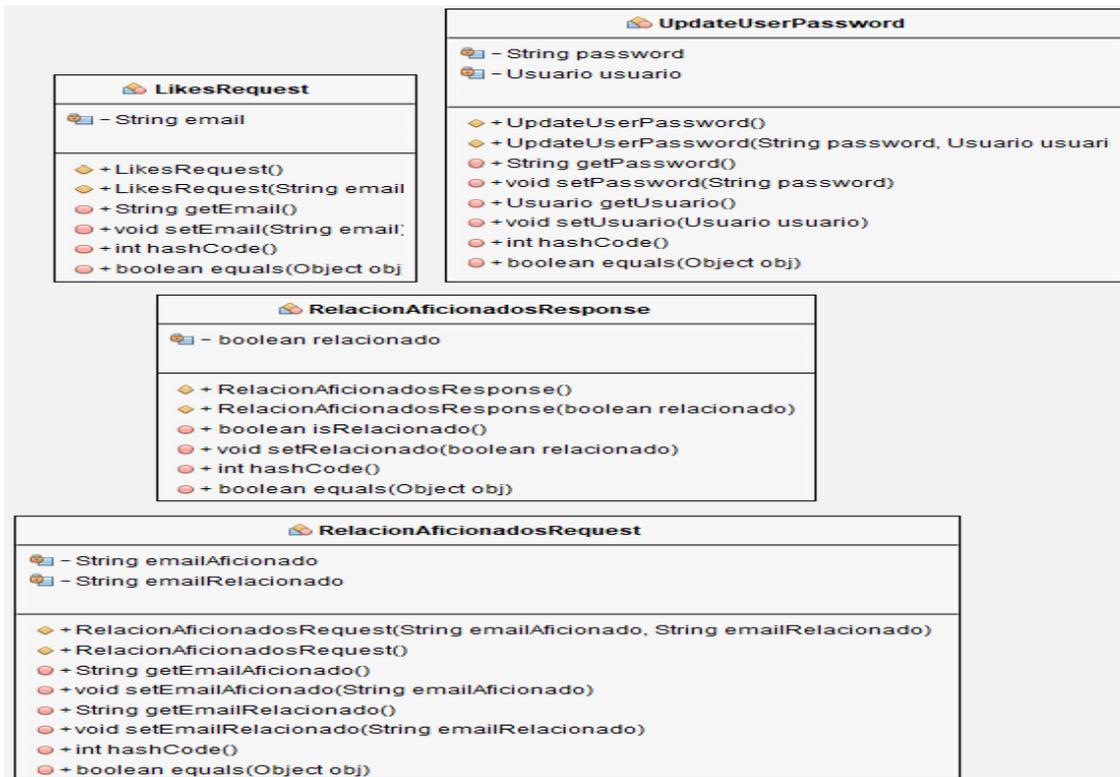


Figura 51. Clases del paquete serviciosrest.controller.bean 3.



Figura 52. Clases del paquete serviciosrest.controller.dto 1.

DetalleUsuarioDTO

- final long serialVersionUID
- String usuarioEmail
- int genero
- int rol
- String descripinte
- LocalDate fechanac
- String foto
- String nick
- Collection<BaileDTO> baileCollection
- CiudadDTO ciudadId

- + DetalleUsuarioDTO()
- + DetalleUsuarioDTO(String usuarioEmail, int genero, int rol, String descripinte, LocalDate fechanac, String foto, String nick, Collection<BaileDTO> baileCollecti
- + String getUsuarioEmail()
- + void setUsuarioEmail(String usuarioEmail)
- + int getGenero()
- + void setGenero(int genero)
- + int getRol()
- + void setRol(int rol)
- + String getDescripinte()
- + void setDescripinte(String descripinte)
- + LocalDate getFechanac()
- + void setFechanac(LocalDate fechanac)
- + String getFoto()
- + void setFoto(String foto)
- + String getNick()
- + void setNick(String nick)
- + Collection<BaileDTO> getBaileCollection()
- + void setBaileCollection(Collection<BaileDTO> baileCollection)
- + CiudadDTO getCiudadId()
- + void setCiudadId(CiudadDTO ciudadId)
- + int hashCode()
- + boolean equals(Object obj)
- + String getEdad()

Figura 53. Clases del paquete serviciosrest.controller.dto 2.

DetalleUsuarioParcialDTO

- final long serialVersionUID
- String usuarioEmail
- int genero
- int rol
- String descripinte
- String fechanac
- String nick
- Collection<String> bailes
- Integer ciudad

- + DetalleUsuarioParcialDTO()
- + DetalleUsuarioParcialDTO(String usuarioEmail, int genero, int rol, String descripinte, String fechanac, String nick, Collection<String> bailes, Integer ciudad)
- + String getUsuarioEmail()
- + void setUsuarioEmail(String usuarioEmail)
- + int getGenero()
- + void setGenero(int genero)
- + int getRol()
- + void setRol(int rol)
- + String getDescripinte()
- + void setDescripinte(String descripinte)
- + String getFechanac()
- + void setFechanac(String fechanac)
- + String getNick()
- + void setNick(String nick)
- + Collection<String> getBailes()
- + void setBailes(Collection<String> bailes)
- + Integer getCiudad()
- + void setCiudad(Integer ciudad)
- + int hashCode()
- + boolean equals(Object obj)

Figura 54. Clases del paquete serviciosrest.controller.dto 3.

 MensajeDataDTO
<ul style="list-style-type: none">  - <u>final long serialVersionUID</u>  - Integer id  - String contenido  - Integer tipo  - LocalDateTime fecha  - Integer leido  - String emisor  - String receptor
<ul style="list-style-type: none">  + MensajeDataDTO()  + MensajeDataDTO(Integer id, String contenido, Integer tipo, LocalDateTime fecha, Integer leido, String emisor, String receptor)  + Integer getId()  + void setId(Integer id)  + String getContenido()  + void setContenido(String contenido)  + Integer getTipo()  + void setTipo(Integer tipo)  + LocalDateTime getFecha()  + void setFecha(LocalDateTime fecha)  + Integer getLeido()  + void setLeido(Integer leido)  + String getEmisor()  + void setEmisor(String emisor)  + String getReceptor()  + void setReceptor(String receptor)  + int hashCode()  + boolean equals(Object obj)

Figura 55. Clases del paquete serviciosrest.controller.dto 4.

 MensajeDTO
<ul style="list-style-type: none">  - <u>final long serialVersionUID</u>  - Integer id  - String contenido  - Integer tipo  - LocalDateTime fecha  - Integer leido  - UsuarioDTO emisor  - UsuarioDTO receptor
<ul style="list-style-type: none">  + MensajeDTO()  + MensajeDTO(Integer id, String contenido, Integer tipo, LocalDateTime fecha, Integer leido, UsuarioDTO emisor, UsuarioDTO receptor)  + Integer getId()  + void setId(Integer id)  + String getContenido()  + void setContenido(String contenido)  + Integer getTipo()  + void setTipo(Integer tipo)  + LocalDateTime getFecha()  + void setFecha(LocalDateTime fecha)  + Integer getLeido()  + void setLeido(Integer leido)  + UsuarioDTO getEmisor()  + void setEmisor(UsuarioDTO emisor)  + UsuarioDTO getReceptor()  + void setReceptor(UsuarioDTO receptor)  + int hashCode()  + boolean equals(Object obj)

Figura 56. Clases del paquete serviciosrest.controller.dto 5.

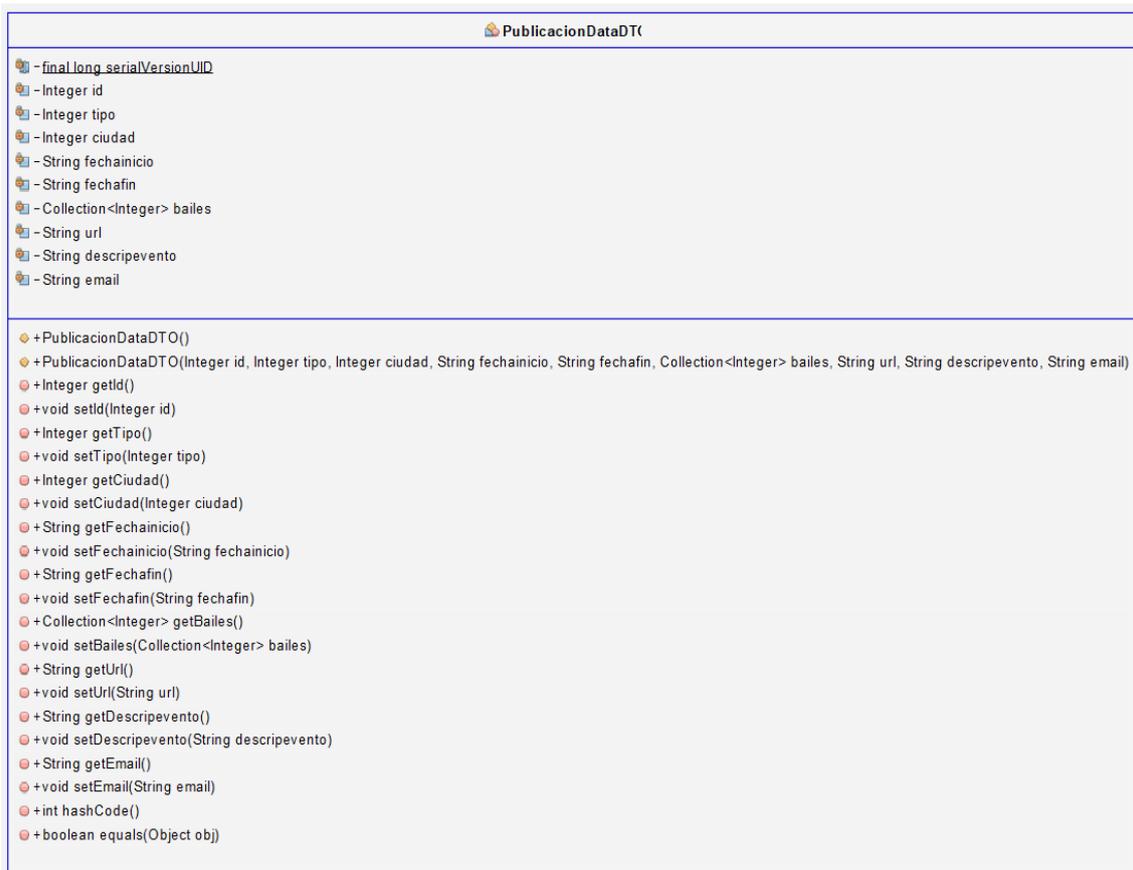


Figura 57. Clases del paquete serviciosrest.controller.dto 6.

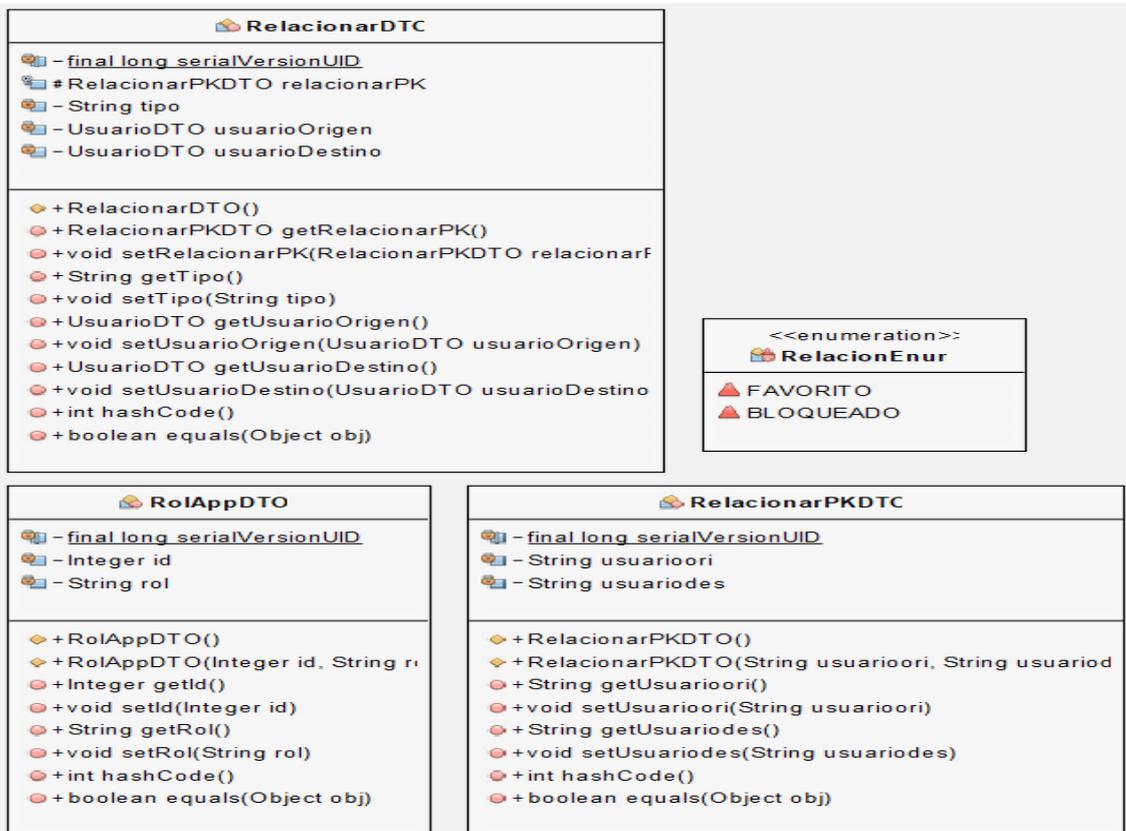


Figura 58. Clases del paquete serviciosrest.controller.dto 7.

 **PublicacionDTO**

-  - final long serialVersionUID
-  - Integer id
-  - String tipo
-  - LocalDate fechaini
-  - LocalDate fechafin
-  - String describevento
-  - String url
-  - Integer estado
-  - Collection<BaileDTO> baileCollection
-  - Collection<UsuarioDTO> usuarioCollection
-  - CiudadDTO ciudad
-  - UsuarioDTO usuario

-  + PublicacionDTO()
-  + PublicacionDTO(Integer id, String tipo, LocalDate fechaini, LocalDate fechafin, String describevento, String url, Integer estado, Collection<BaileDTO> baileCollection, Collection<UsuarioDTO> usuarioCollection, CiudadDTO ciudad, UsuarioDTO usuario)
-  + Integer getId()
-  + void setId(Integer id)
-  + String getTipo()
-  + void setTipo(String tipo)
-  + LocalDate getFechaini()
-  + void setFechaini(LocalDate fechaini)
-  + LocalDate getFechafin()
-  + void setFechafin(LocalDate fechafin)
-  + String getDescribevento()
-  + void setDescribevento(String describevento)
-  + String getUrl()
-  + void setUrl(String url)
-  + Integer getEstado()
-  + void setEstado(Integer estado)
-  + Collection<BaileDTO> getBaileCollection()
-  + void setBaileCollection(Collection<BaileDTO> baileCollection)
-  + Collection<UsuarioDTO> getUsuarioCollection()
-  + void setUsuarioCollection(Collection<UsuarioDTO> usuarioCollection)
-  + CiudadDTO getCiudad()
-  + void setCiudad(CiudadDTO ciudad)
-  + UsuarioDTO getUsuario()
-  + void setUsuario(UsuarioDTO usuario)
-  + int hashCode()
-  + boolean equals(Object obj)

Figura 59. Clases del paquete serviciosrest.controller.dto 8.

 **ReporteDataDTO**

-  - final long serialVersionUID
-  - Integer id
-  - String idReportado
-  - Integer tipo
-  - LocalDate fechareporte
-  - String contenido
-  - Integer motivo
-  - String emailnotificante

-  + ReporteDataDTO()
-  + ReporteDataDTO(Integer id, String idReportado, Integer tipo, LocalDate fechareporte, String contenido, Integer motivo, String emailnotificante)
-  + Integer getId()
-  + void setId(Integer id)
-  + String getIdReportado()
-  + void setIdReportado(String idReportado)
-  + Integer getTipo()
-  + void setTipo(Integer tipo)
-  + LocalDate getFechareporte()
-  + void setFechareporte(LocalDate fechareporte)
-  + String getContenido()
-  + void setContenido(String contenido)
-  + Integer getMotivo()
-  + void setMotivo(Integer motivo)
-  + String getEmailnotificante()
-  + void setEmailnotificante(String emailnotificante)
-  + int hashCode()
-  + boolean equals(Object obj)

Figura 60. Clases del paquete serviciosrest.controller.dto 9.

ReporteDTO
<ul style="list-style-type: none"> 📄 - final long serialVersionUID 📄 - Integer id 📄 - Integer tipo 📄 - String idReportado 📄 - LocalDate fechareporte 📄 - String contenido 📄 - MotivoReporteDTO motivo 📄 - UsuarioDTO emailnotificante 📄 - String aliasReportado
<ul style="list-style-type: none"> 🔹 + ReporteDTO() 🔹 + ReporteDTO(Integer id, Integer tipo, String idReportado, LocalDate fechareporte, String contenido, MotivoReporteDTO motivo, UsuarioDTO emailnotificante, String aliasReportado) 🔸 + Integer getId() 🔸 + void setId(Integer id) 🔸 + Integer getTipo() 🔸 + void setTipo(Integer tipo) 🔸 + String getIdReportado() 🔸 + void setIdReportado(String idReportado) 🔸 + LocalDate getFechareporte() 🔸 + void setFechareporte(LocalDate fechareporte) 🔸 + String getContenido() 🔸 + void setContenido(String contenido) 🔸 + MotivoReporteDTO getMotivo() 🔸 + void setMotivo(MotivoReporteDTO motivo) 🔸 + UsuarioDTO getEmailnotificante() 🔸 + void setEmailnotificante(UsuarioDTO emailnotificante) 🔸 + String getAliasReportado() 🔸 + void setAliasReportado(String aliasReportado) 🔸 + int hashCode() 🔸 + boolean equals(Object obj)

Figura 61. Clases del paquete serviciosrest.controller.dto 10.

UsuarioDTO

- final long serialVersionUID
- String email
- String password
- LocalDate fechacrea
- LocalDate fechamodi
- int estado
- Collection<RolAppDTO> rolAppCollection
- Collection<ConfiguracionDTO> configuracionCollection
- DetalleUsuarioDTO detalleUsuario

- + UsuarioDTO()
- + UsuarioDTO(String email, String password, LocalDate fechacrea, LocalDate fechamodi, int estado, Collection<RolAppDTO> rolAppCollection, Collection<ConfiguracionDTO> configuracionCollection, DetalleUsuarioDTO detalleUsuario)
- + String getEmail()
- + void setEmail(String email)
- + String getPassword()
- + void setPassword(String password)
- + LocalDate getFechacrea()
- + void setFechacrea(LocalDate fechacrea)
- + LocalDate getFechamodi()
- + void setFechamodi(LocalDate fechamodi)
- + int getEstado()
- + void setEstado(int estado)
- + Collection<RolAppDTO> getRolAppCollection()
- + void setRolAppCollection(Collection<RolAppDTO> rolAppCollection)
- + Collection<ConfiguracionDTO> getConfiguracionCollection()
- + void setConfiguracionCollection(Collection<ConfiguracionDTO> configuracionCollection)
- + DetalleUsuarioDTO getDetalleUsuario()
- + void setDetalleUsuario(DetalleUsuarioDTO detalleUsuario)
- + int hashCode()
- + boolean equals(Object obj)

Figura 62. Clases del paquete serviciosrest.controller.dto 11.

<p style="text-align: center;">LocalDateDeserializer</p> <hr/> <ul style="list-style-type: none"> - final long serialVersionUID <hr/> <ul style="list-style-type: none"> # LocalDateDeserializer() + LocalDate deserialize(JsonParser jp, DeserializationContext cbxt) 	<p style="text-align: center;">LocalDateSerializer</p> <hr/> <ul style="list-style-type: none"> - final long serialVersionUID <hr/> <ul style="list-style-type: none"> + LocalDateSerializer() + void serialize(LocalDate value, JsonGenerator gen, SerializerProvider sp)
<p style="text-align: center;">LocalDate TimeDeserializer</p> <hr/> <ul style="list-style-type: none"> - final long serialVersionUID <hr/> <ul style="list-style-type: none"> # LocalDate TimeDeserializer() + LocalDate Time deserialize(JsonParser jp, DeserializationContext cbxt) 	<p style="text-align: center;">LocalDate Time Serializer</p> <hr/> <ul style="list-style-type: none"> - final long serialVersionUID <hr/> <ul style="list-style-type: none"> + LocalDate TimeSerializer() + void serialize(LocalDate Time value, JsonGenerator gen, SerializerProvider sp)

Figura 63. Clases del paquete serviciosrest.controller.formatter.

ServiciosREST: clases servicio de los servicios

Clases para manejar la lógica de los datos.



Figura 64. Clases del paquete service 1.

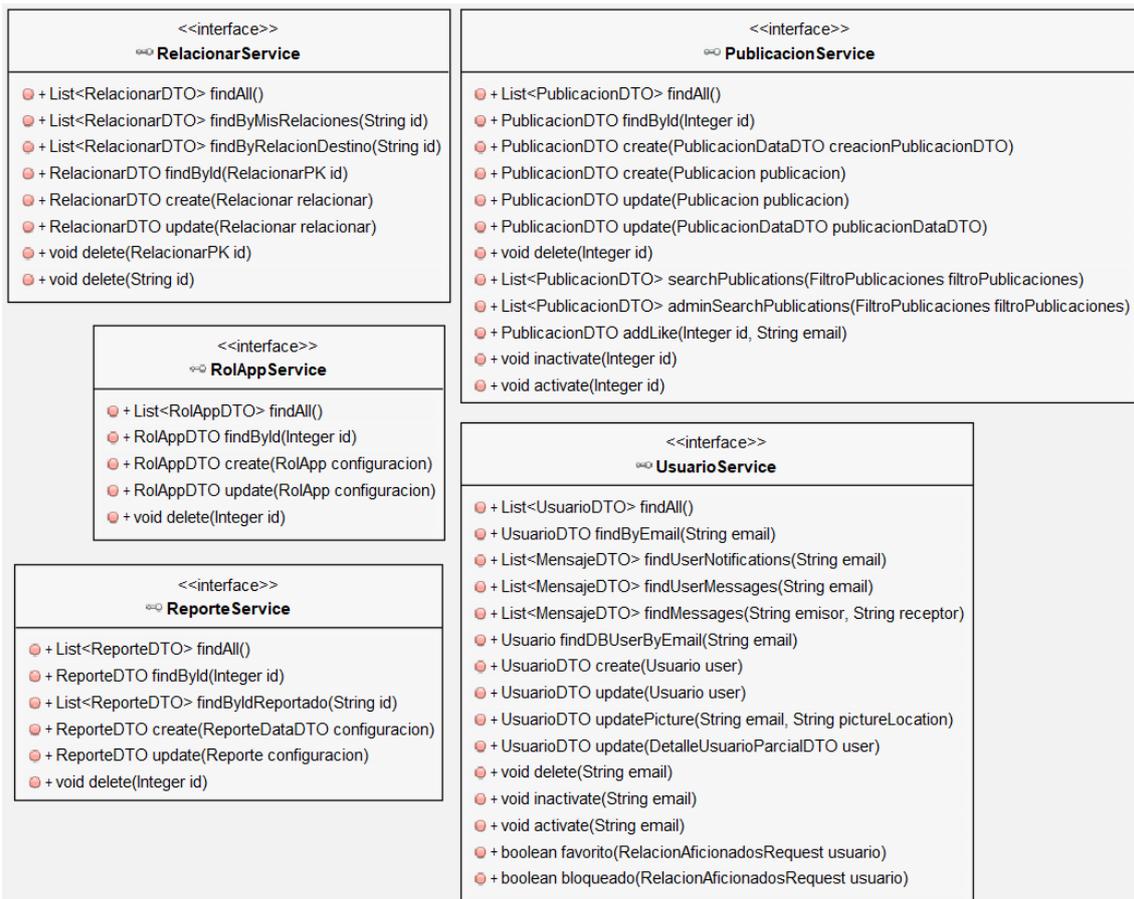


Figura 65. Clases del paquete service 2.



Figura 66. Clases del paquete service.converter 1.

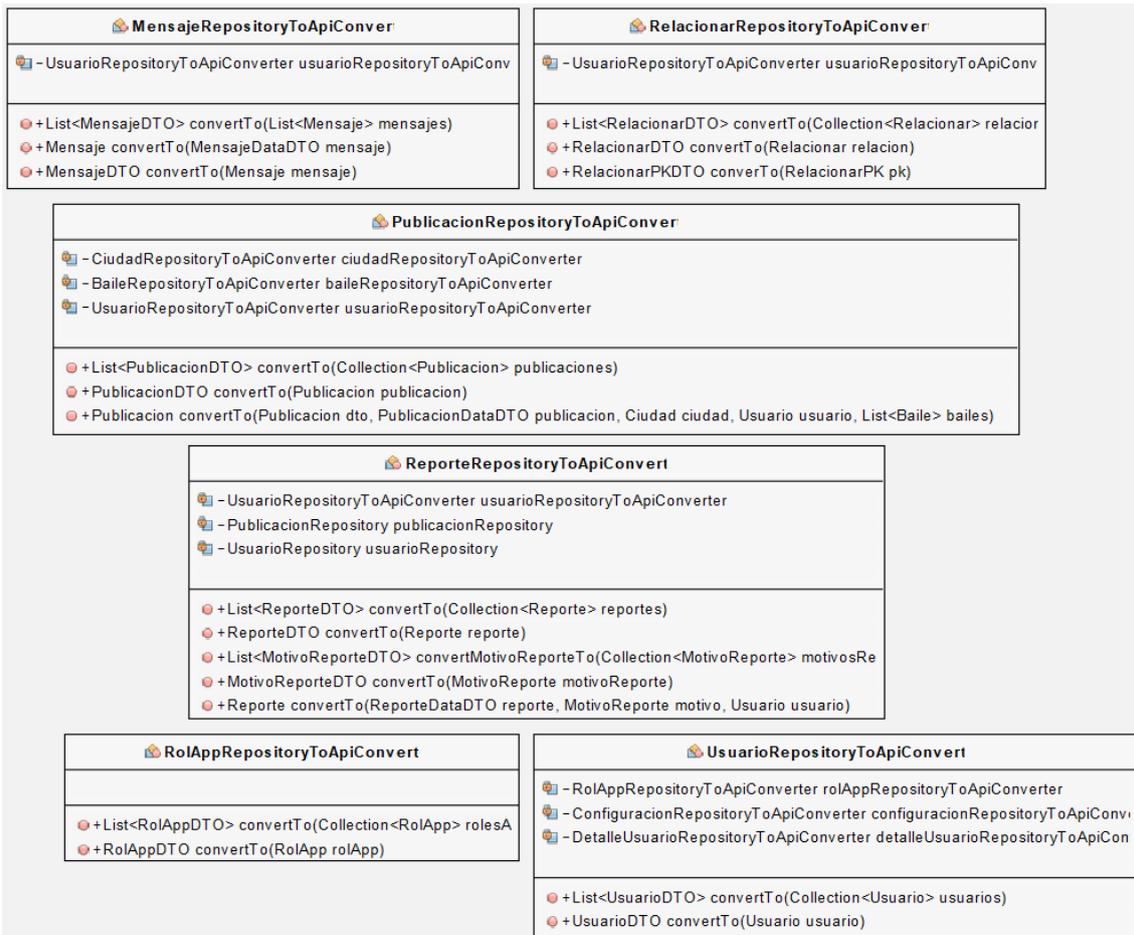


Figura 67. Clases del paquete service.converter 2.

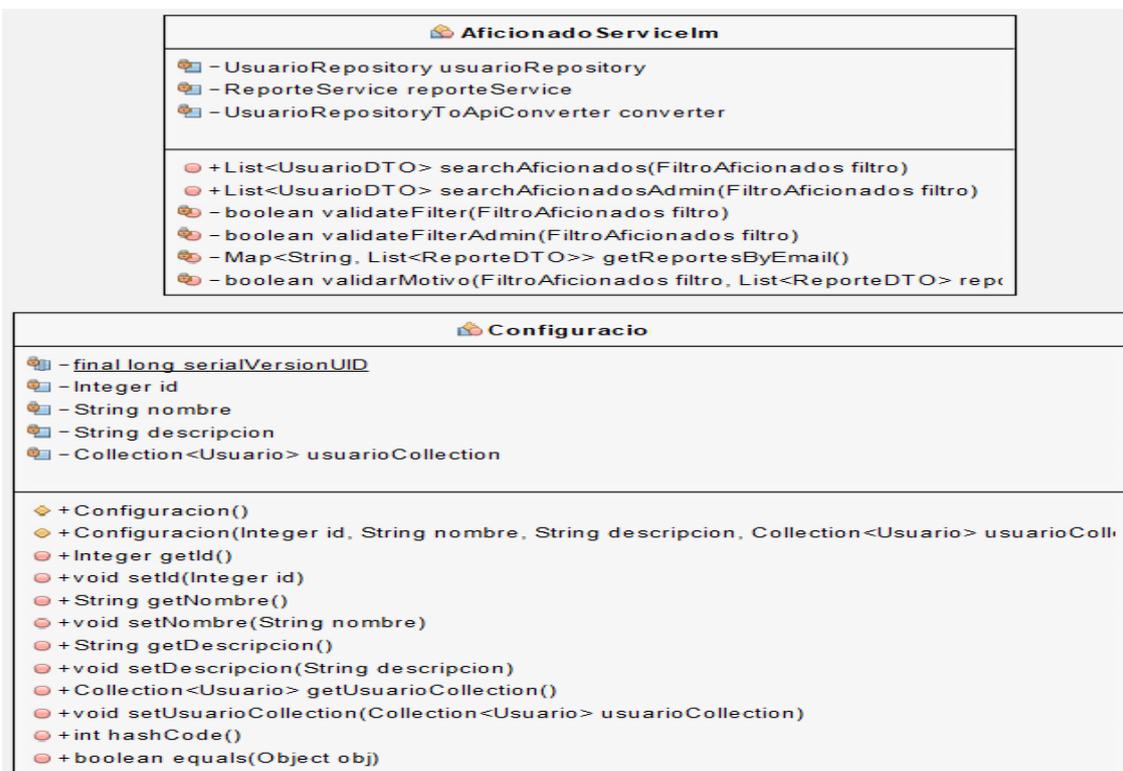


Figura 68. Clases del paquete service.impl 1.

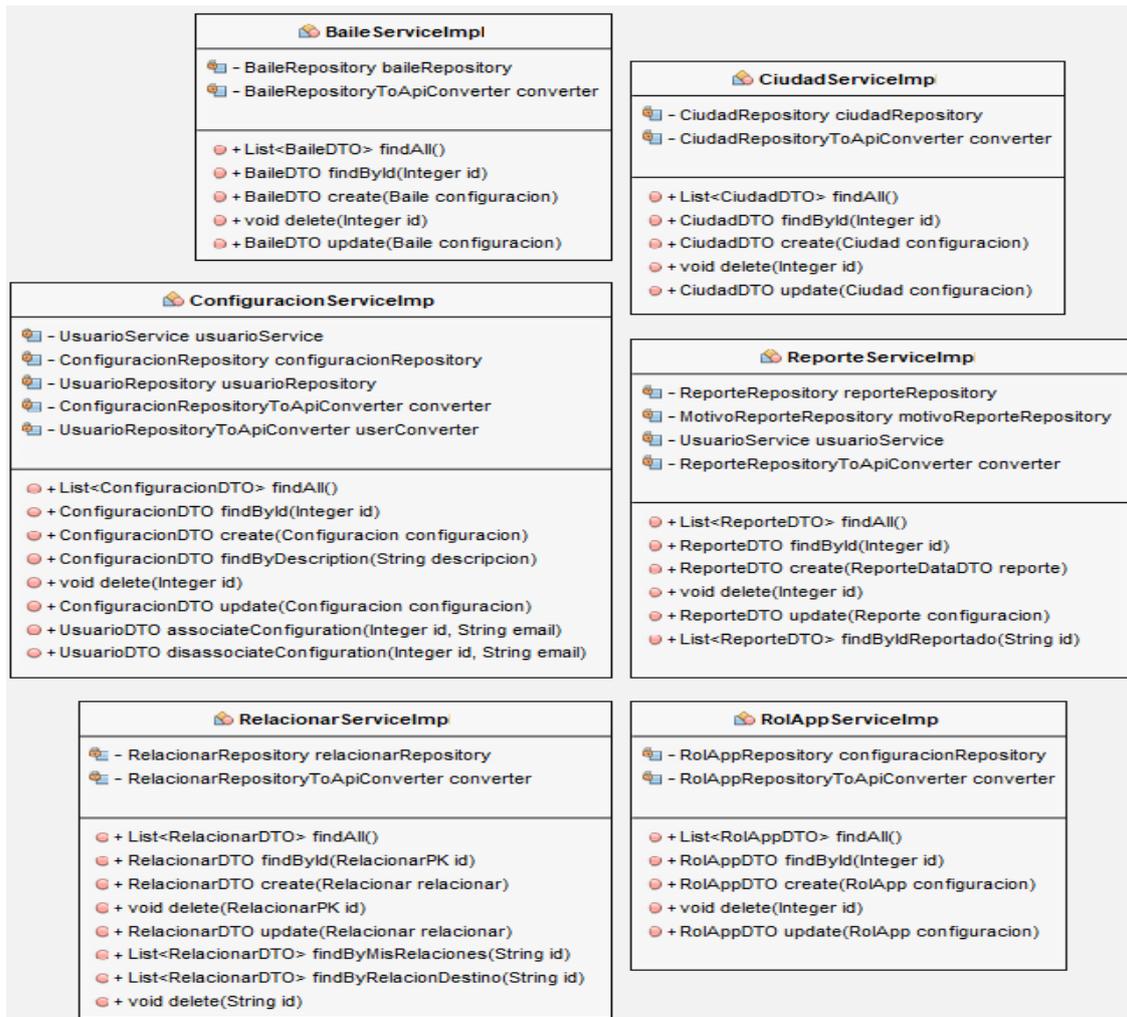


Figura 69. Clases del paquete service.impl 2.

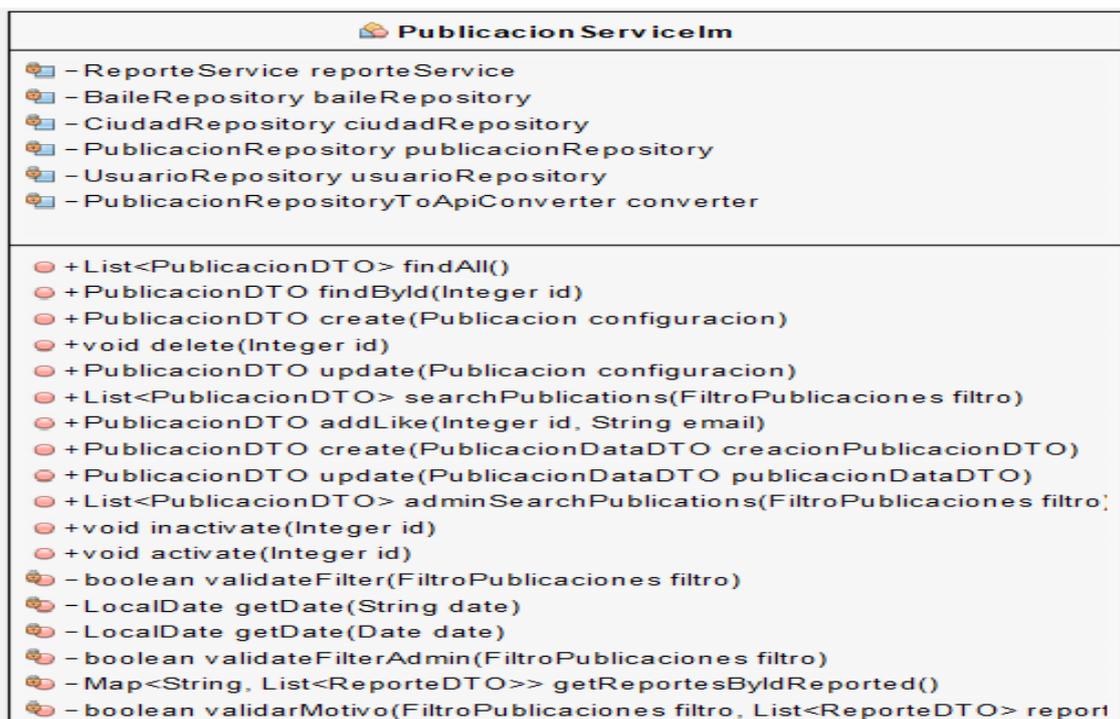


Figura 70. Clases del paquete service.impl 3.

📦 UsuarioServiceImpl
<ul style="list-style-type: none"> 📦 - UsuarioRepository usuarioRepository 📦 - BaileRepository baileRepository 📦 - CiudadRepository ciudadRepository 📦 - RelacionarRepository relacionarRepository 📦 - MensajeRepository mensajeRepository 📦 - UsuarioRepositoryToApiConverter converter 📦 - MensajeRepositoryToApiConverter mensajeConverter
<ul style="list-style-type: none"> 🔴 + List<UsuarioDTO> findAll() 🔴 + UsuarioDTO findByEmail(String email) 🔴 + Usuario findDBUserByEmail(String email) 🔴 + UsuarioDTO create(Usuario user) 🔴 + UsuarioDTO update(Usuario user) 🔴 + void delete(String email) 🔴 + void inactivate(String email) 🔴 + void activate(String email) 🔴 + UsuarioDTO update(DetalleUsuarioParcialDTO user) 🔴 + boolean favorito(RelacionAficionadosRequest usuario) 🔴 + boolean bloqueo(RelacionAficionadosRequest usuario) 🔴 - boolean relacionarUsuario(RelacionAficionadosRequest usuario, String relacion) 🔴 - void addRelation(RelacionAficionadosRequest usuario, Usuario user, String relacion) 🔴 + List<MensajeDTO> findUserNotifications(String email) 🔴 + //busca todos los mensajes donde ese email es un emisor o un receptor List<MensajeDTO> findUserMessages(String email) 🔴 + List<MensajeDTO> findMessages(String emisor, String receptor) 🔴 + UsuarioDTO updatePicture(String email, String pictureLocation)

Figura 71. Clases del paquete service.impl 4.

ServiciosREST: clases entidad de los servicios

Clases para manejar el acceso a los datos.

📦 Baile
<ul style="list-style-type: none"> 📦 - final long serialVersionUID 📦 - Integer id 📦 - String nombre 📦 - Collection<DetalleUsuario> detalleUsuarioCollection 📦 - Collection<Publicacion> publicacionCollection
<ul style="list-style-type: none"> 🔴 + Baile() 🔴 + Baile(Integer id) 🔴 + Baile(Integer id, String nombre) 🔴 + Integer getId() 🔴 + void setId(Integer id) 🔴 + String getNombre() 🔴 + void setNombre(String nombre) 🔴 + Collection<DetalleUsuario> getDetalleUsuarioCollection() 🔴 + void setDetalleUsuarioCollection(Collection<DetalleUsuario> detalleUsuarioCollection) 🔴 + Collection<Publicacion> getPublicacionCollection() 🔴 + void setPublicacionCollection(Collection<Publicacion> publicacionCollection) 🔴 + int hashCode() 🔴 + boolean equals(Object object)

📦 Ciudad
<ul style="list-style-type: none"> 📦 - final long serialVersionUID 📦 - Integer id 📦 - String ciudad 📦 - Collection<DetalleUsuario> detalleUsuarioCollection 📦 - Collection<Publicacion> publicacionCollection
<ul style="list-style-type: none"> 🔴 + Ciudad() 🔴 + Ciudad(Integer id) 🔴 + Integer getId() 🔴 + void setId(Integer id) 🔴 + String getCiudad() 🔴 + void setCiudad(String ciudad) 🔴 + Collection<DetalleUsuario> getDetalleUsuarioCollection() 🔴 + void setDetalleUsuarioCollection(Collection<DetalleUsuario> detalleUsuarioCollection) 🔴 + Collection<Publicacion> getPublicacionCollection() 🔴 + void setPublicacionCollection(Collection<Publicacion> publicacionCollection) 🔴 + int hashCode() 🔴 + boolean equals(Object object)

Figura 72. Clases del paquete entity 1.

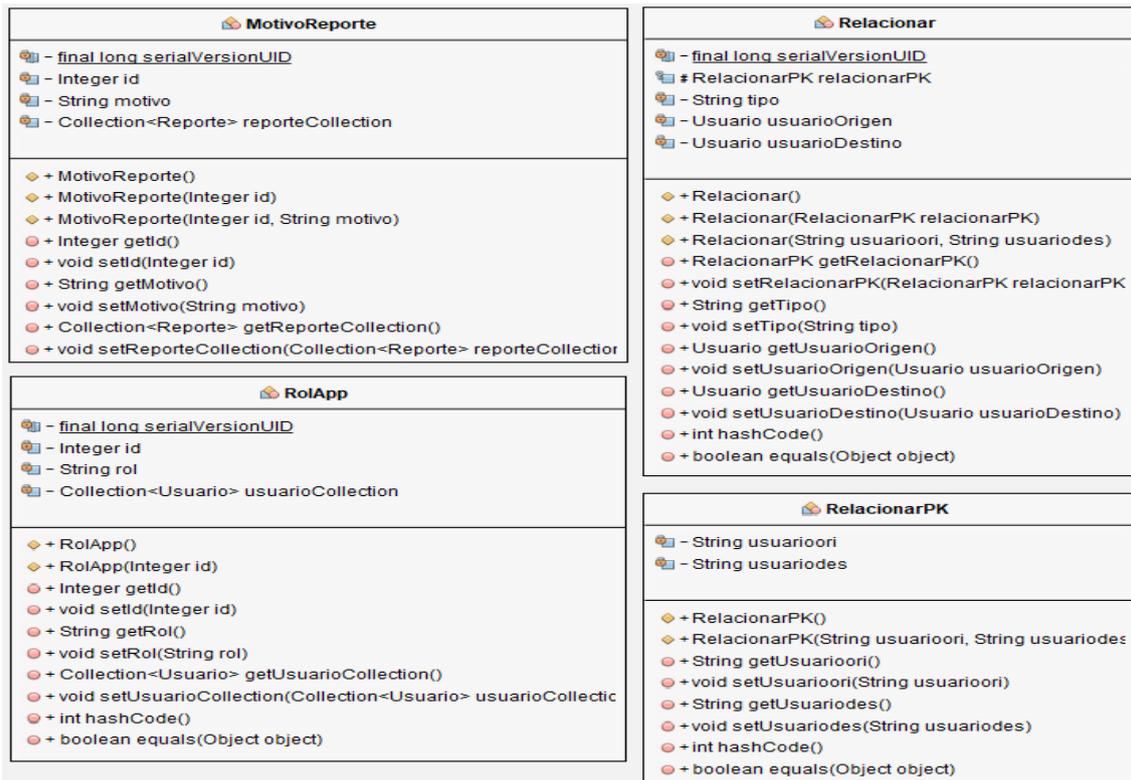


Figura 73. Clases del paquete entity 2.

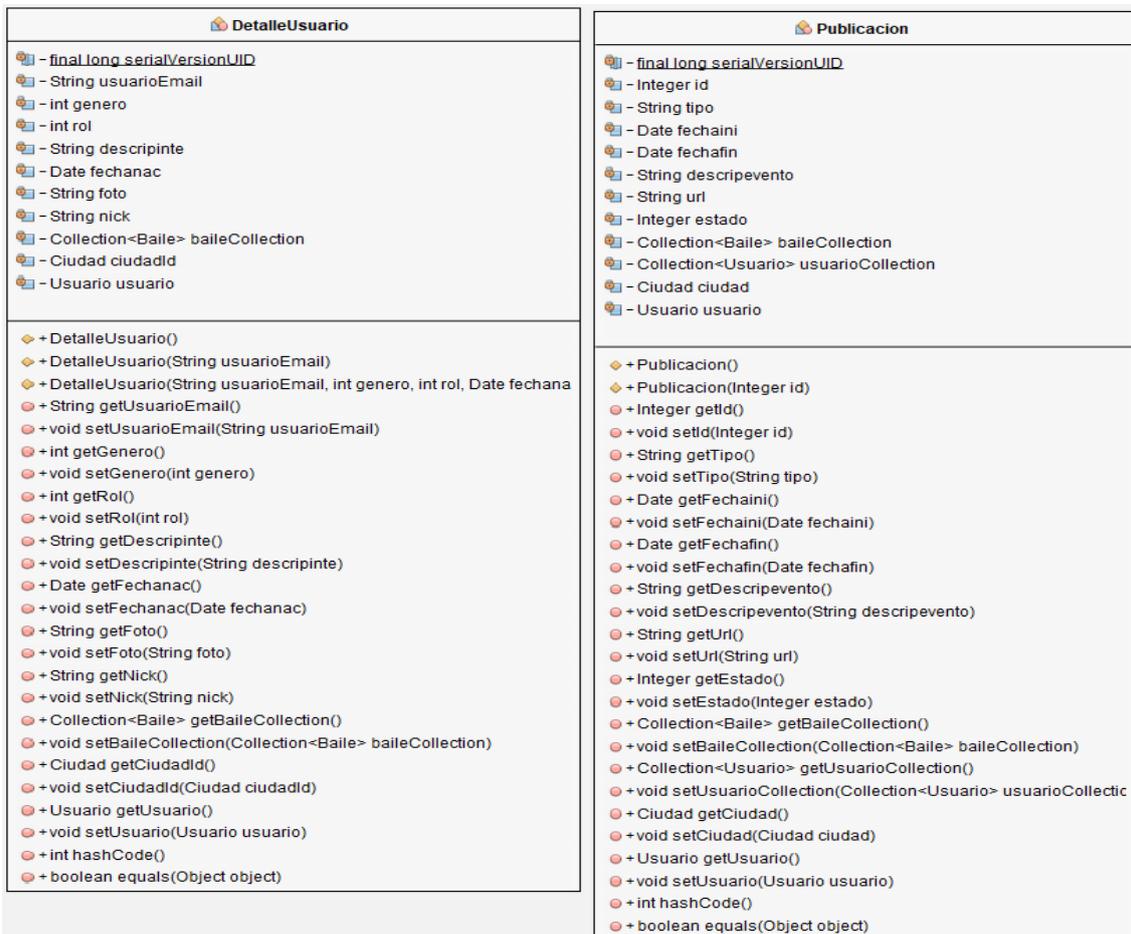


Figura 74. Clases del paquete entity 3.

Reporte	Mensaje
<ul style="list-style-type: none"> - final long serialVersionUID - Integer id - String idreportado - Integer tipo - Date fechareporte - String contenido - MotivoReporte motivo - Usuario emailnotificante 	<ul style="list-style-type: none"> - final long serialVersionUID - Integer id - String contenido - Integer tipo - Date fecha - Integer leido - Usuario emisor - Usuario receptor
<ul style="list-style-type: none"> + Reporte() + Reporte(Integer id) + Integer getId() + void setId(Integer id) + String getIdreportado() + void setIdreportado(String idreportado) + Integer getTipo() + void setTipo(Integer tipo) + Date getFechareporte() + void setFechareporte(Date fechareporte) + String getContenido() + void setContenido(String contenido) + MotivoReporte getMotivo() + void setMotivo(MotivoReporte motivo) + Usuario getEmailnotificante() + void setEmailnotificante(Usuario emailnotifican) + int hashCode() + boolean equals(Object object) 	<ul style="list-style-type: none"> + Mensaje() + Mensaje(Integer id) + Integer getId() + void setId(Integer id) + String getContenido() + void setContenido(String contenic) + Integer getTipo() + void setTipo(Integer tipo) + Date getFecha() + void setFecha(Date fecha) + Integer getLeido() + void setLeido(Integer leido) + Usuario getEmisor() + void setEmisor(Usuario emisor) + Usuario getReceptor() + void setReceptor(Usuario recepto) + int hashCode() + boolean equals(Object object)

Figura 75. Clases del paquete entity 4.

Usuario
<ul style="list-style-type: none"> - final long serialVersionUID - String email - String password - Date fechaarea - Date fechamodi - int estado - Collection<RolApp> rolAppCollection - Collection<Configuracion> configuracionCollection - Collection<Publicacion> publicacionCollection - Collection<Relacionar> relacionarCollection - Collection<Relacionar> relacionarCollection1 - DetalleUsuario detalleUsuario - Collection<Mensaje> emisorMessages - Collection<Mensaje> receptorMessages - Collection<Publicacion> publicacionCollection1 - Collection<Reporte> reporteCollection
<ul style="list-style-type: none"> + Usuario() + Usuario(String email) + Usuario(String email, String password, Date fechaarea, int estado) + String getEmail() + void setEmail(String email) + String getPassword() + void setPassword(String password) + Date getFechaarea() + void setFechaarea(Date fechaarea) + Date getFechamodi() + void setFechamodi(Date fechamodi) + int getEstado() + void setEstado(int estado) + Collection<RolApp> getRolAppCollection() + void setRolAppCollection(Collection<RolApp> rolAppCollection) + Collection<Configuracion> getConfiguracionCollection() + void setConfiguracionCollection(Collection<Configuracion> configuracionCollecti) + Collection<Publicacion> getPublicacionCollection() + void setPublicacionCollection(Collection<Publicacion> publicacionCollection) + Collection<Relacionar> getRelacionarCollection() + void setRelacionarCollection(Collection<Relacionar> relacionarCollection) + Collection<Relacionar> getRelacionarCollection1() + void setRelacionarCollection1(Collection<Relacionar> relacionarCollection1) + DetalleUsuario getDetalleUsuario() + void setDetalleUsuario(DetalleUsuario detalleUsuario) + Collection<Mensaje> getEmisorMessages() + void setEmisorMessagesn(Collection<Mensaje> emisorMessages) + Collection<Mensaje> getReceptorMessages() + void setReceptorMessages(Collection<Mensaje> receptorMessages) + Collection<Publicacion> getPublicacionCollection1() + void setPublicacionCollection1(Collection<Publicacion> publicacionCollection1) + Collection<Reporte> getReporteCollection() + void setReporteCollection(Collection<Reporte> reporteCollection) + int hashCode() + boolean equals(Object object) + String toString()

Figura 76. Clases del paquete entity 5.

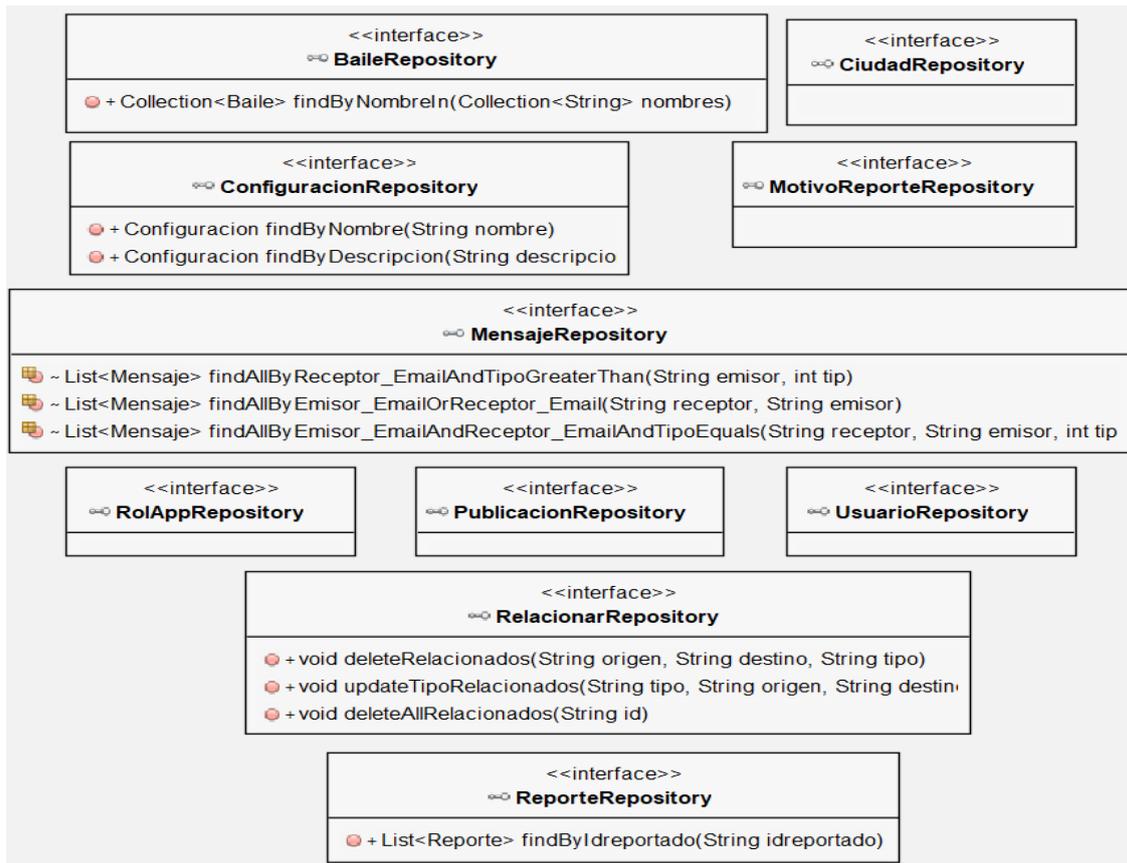


Figura 77. Clases del paquete entity.repository.

5.5. Diagramas de secuencia

Los diagramas de secuencia aportan información de como los mensajes interactúan entre los componentes de un sistema al realizar una determinada funcionalidad a lo largo del tiempo. Por tanto, estos contribuyen a facilitar la comprensión de los caso de uso.

Seguidamente se van a presentar los diagramas de la aplicación elaborados en la etapa de diseño.

Registrarse

El diagrama de abajo (ver figura 78), se corresponde con el caso de uso 'UNR1 - Registrarse' describiendo las interacciones del registro en el sistema.

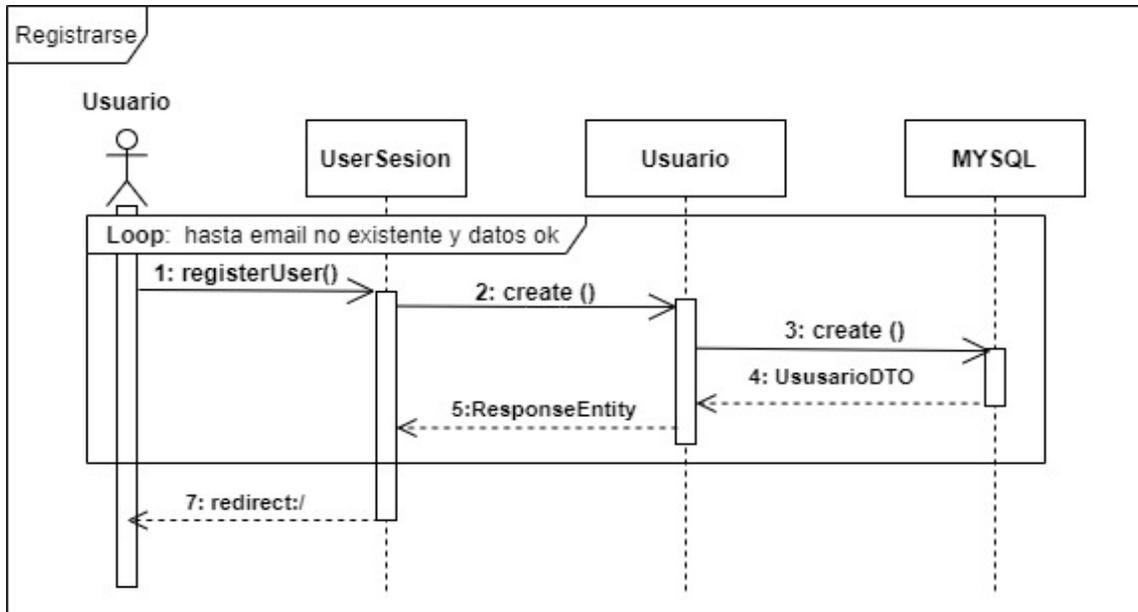


Figura 78. Diagrama de secuencia registrarse.

Login

El diagrama de abajo (ver figura 79), se corresponde con el caso de uso 'UR1 - Login' describiendo las interacciones del inicio de sesión en el sistema.

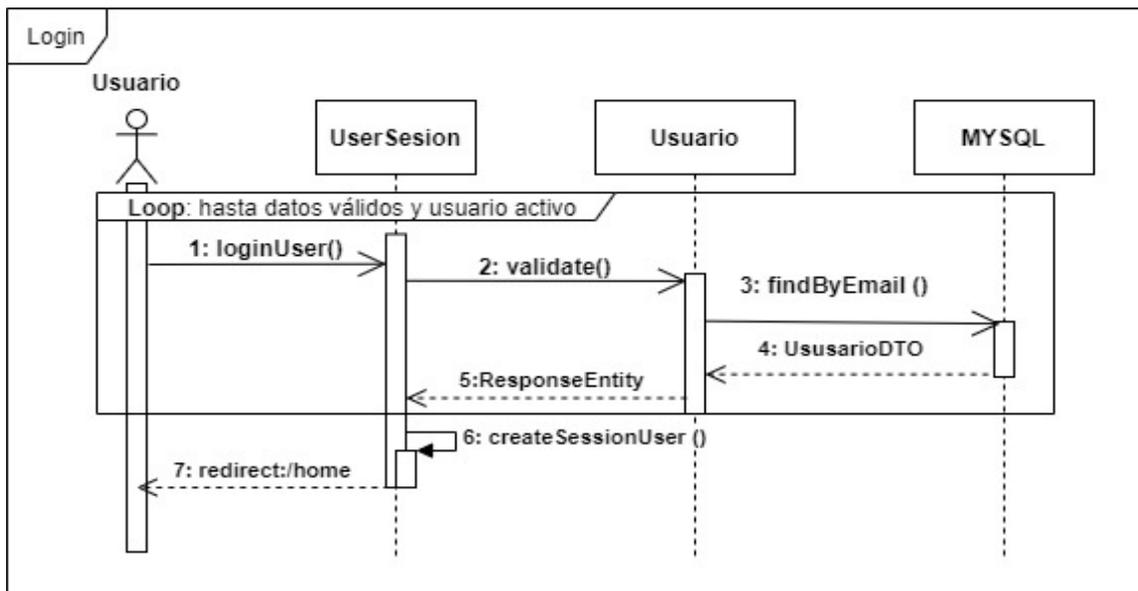


Figura 79. Diagrama de secuencia login.

Recuperar contraseña

El diagrama de abajo (ver figura 80), se corresponde con el caso de uso 'UR2 - Recuperar contraseña por email' describiendo las interacciones de la recuperación de la clave de acceso mediante el envío de un correo electrónico.

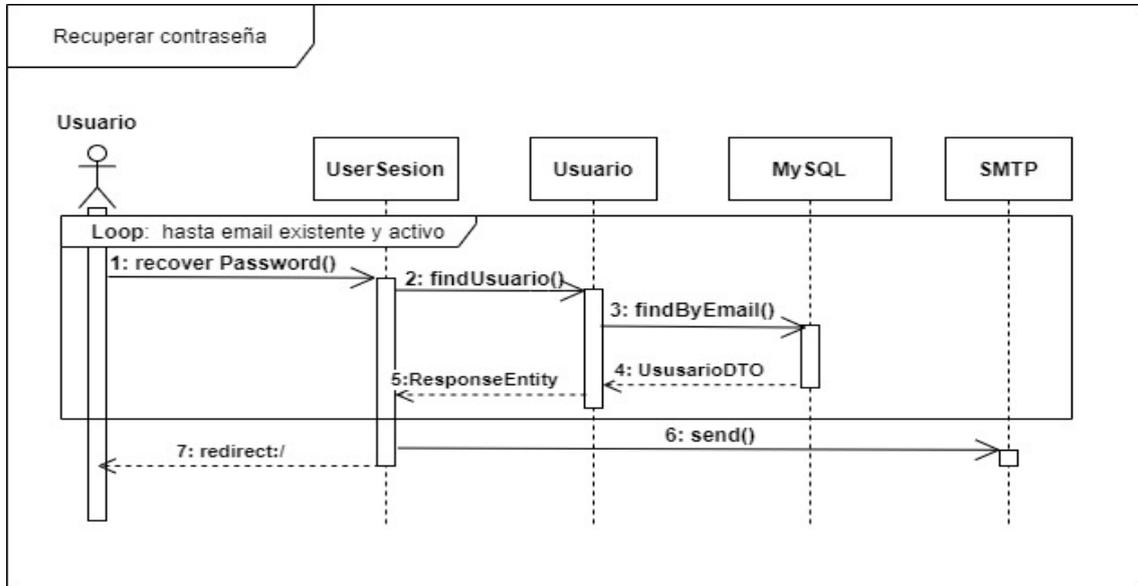


Figura 80. Diagrama de secuencia recuperar contraseña.

Logout

El diagrama de abajo (ver figura 81), se corresponde con el caso de uso 'UR3 - Logout' describiendo las interacciones del cierre de sesión en el sistema.

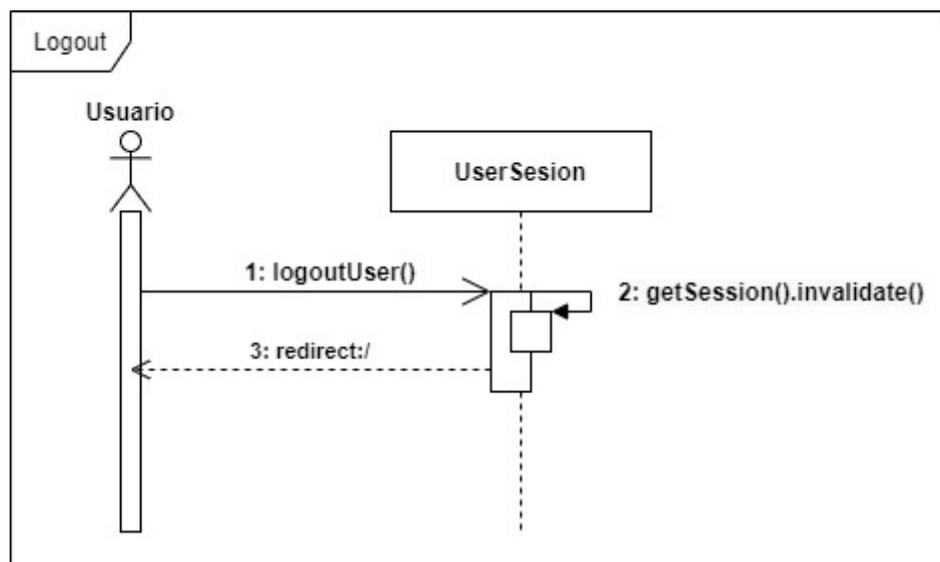


Figura 81. Diagrama de secuencia logout.

Ver perfil detallado

El diagrama de abajo (ver figura 82), se corresponde con los casos de uso ‘UG1 - Ver perfil propio’, ‘UG10 - Ver publicaciones propias’, ‘UG15 - Consultar la lista propia de favoritos’, así como también los casos de uso ‘UG17 - Ver perfiles’ y ‘UG22 - Ver publicaciones’, describiendo las interacciones al ver un perfil en detalle.

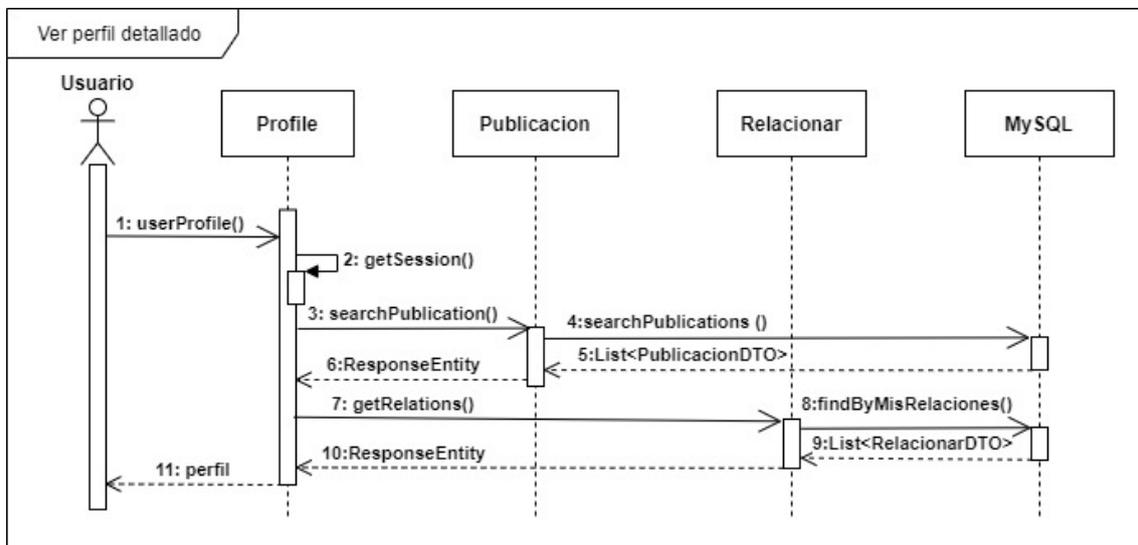


Figura 82. Diagrama de secuencia ver perfil detallado.

Consultar configuración propia

El diagrama de abajo (ver figura 83), se corresponde con el caso de uso ‘UG2 - Consultar configuración propia’ describiendo las interacciones al consultar el usuario su propia configuración.

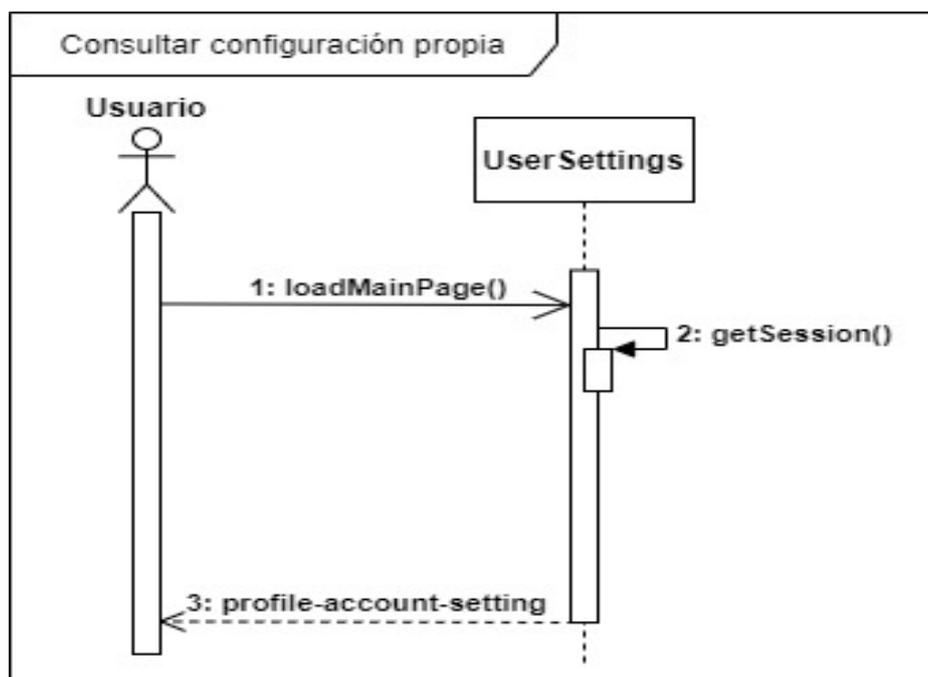


Figura 83. Diagrama de secuencia de consultar configuración propia.

Gestionar la configuración

El diagrama de abajo (ver figura 84), se corresponde con los casos de uso ‘UG3 - Gestionar las notificaciones por email de los mensajes recibidos’, ‘UG4 - Gestionar las notificaciones por email de las publicaciones de los favoritos’, ‘UG5 - Gestionar la visibilidad de la edad’ y ‘UG6 - Gestionar la visibilidad del email’, describiendo las interacciones al gestionar la configuración con cualquiera de las opciones configurables.

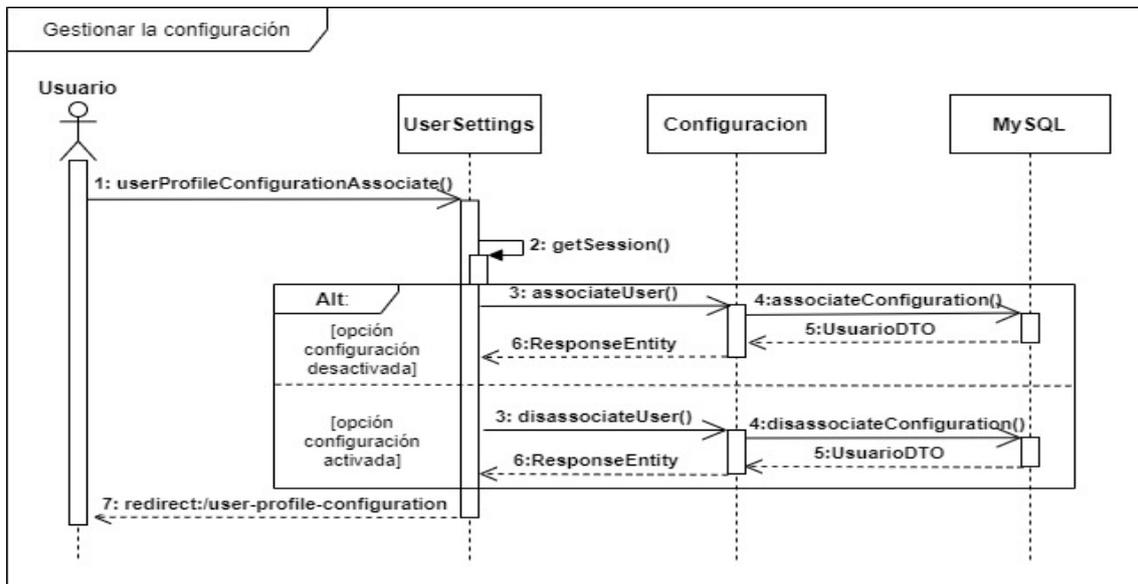


Figura 84. Diagrama de secuencia de gestionar la configuración.

Cambiar foto de perfil

El diagrama de abajo (ver figura 85), se corresponde con el caso de uso ‘UG7 - Editar perfil’, describiendo las interacciones al modificar el perfil, en concreto al subir una foto de perfil.

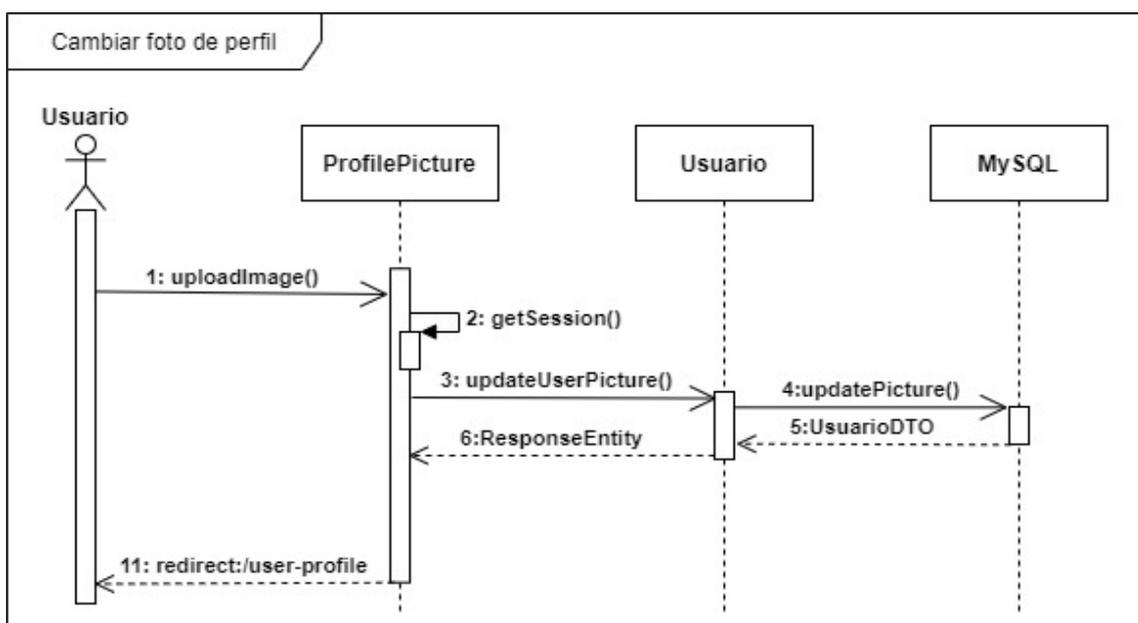


Figura 85. Diagrama de secuencia de cambiar foto de perfil.

Editar perfil

El diagrama de abajo (ver figura 86), se corresponde con el caso de uso 'UG7 - Editar perfil' describiendo las interacciones al modificar el perfil, en concreto de los campos modificables a excepción de la foto.

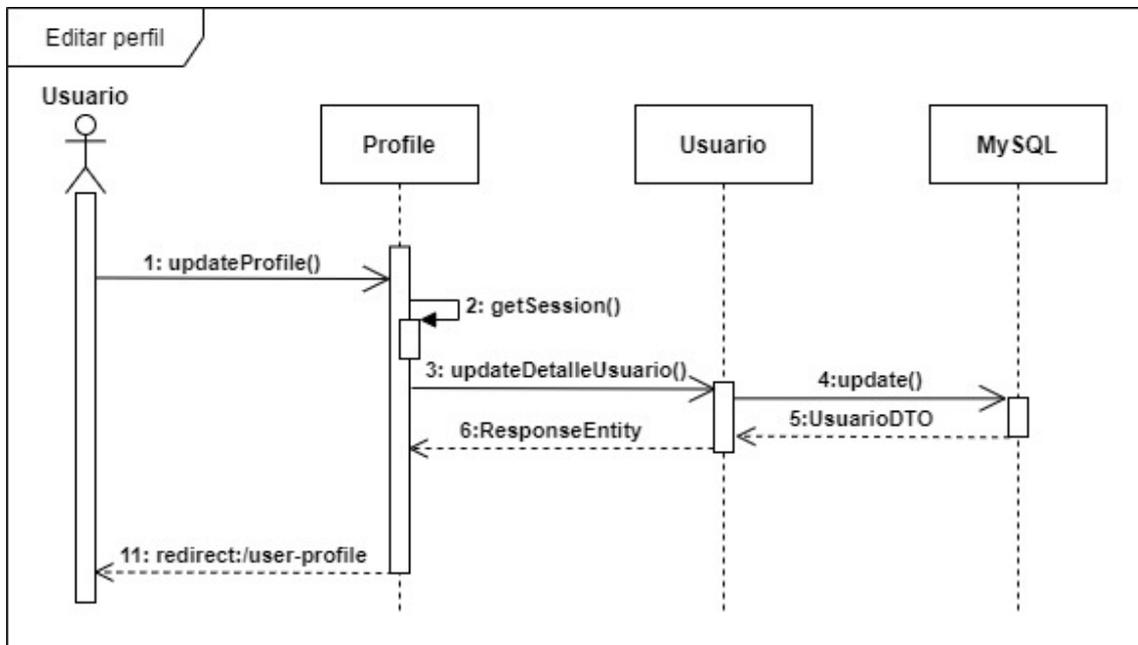


Figura 86. Diagrama de secuencia de editar perfil.

Eliminar perfil

El diagrama de abajo (ver figura 87), se corresponde con el caso de uso 'UG8 - Eliminar perfil' describiendo las interacciones al desactivar el propio perfil de usuario.

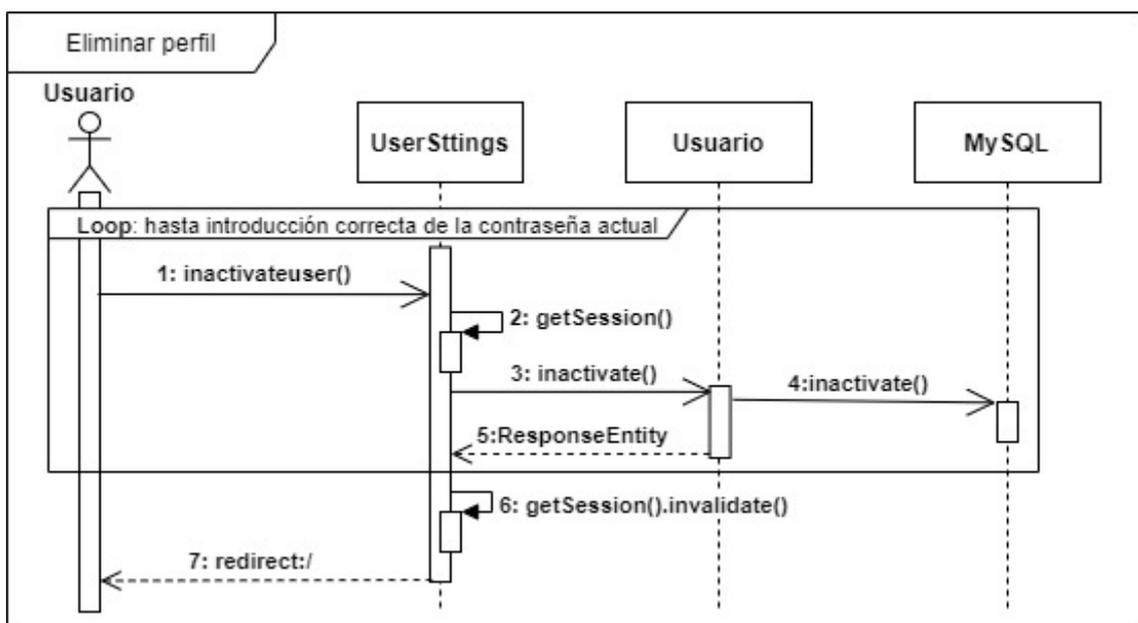


Figura 87. Diagrama de secuencia de eliminar perfil.

Cambiar contraseña

El diagrama de abajo (ver figura 88), se corresponde con el caso de uso ‘UG9 - Cambiar contraseña’ describiendo las interacciones al modificar la contraseña.

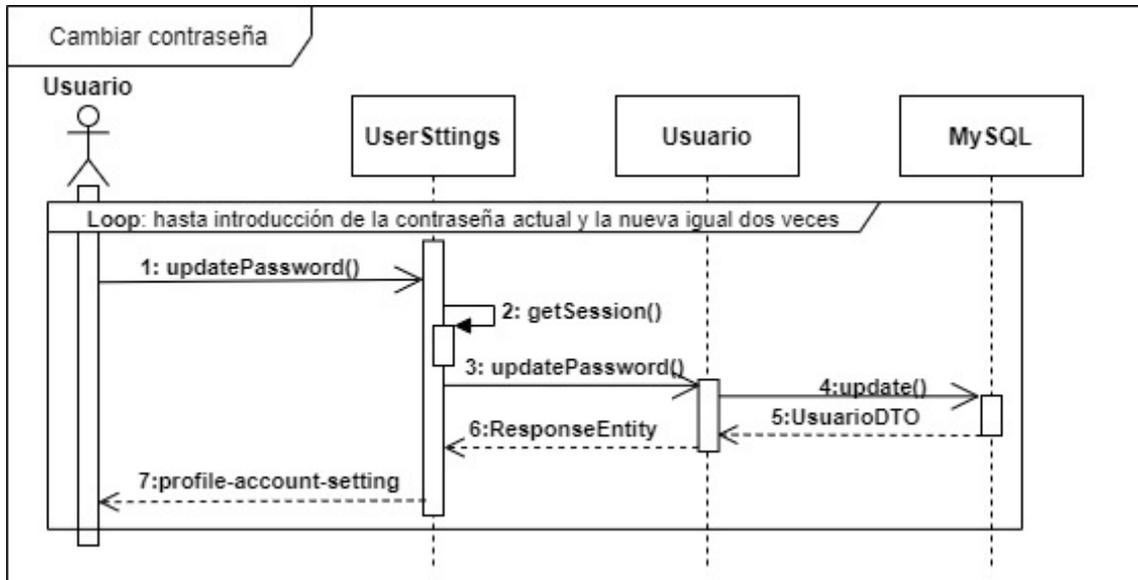


Figura 88. Diagrama de secuencia de cambiar contraseña.

Crear publicación

El diagrama de abajo (ver figura 89), se corresponde con el caso de uso ‘UG11 - Crear publicación’ describiendo las interacciones al generar una nueva publicación.

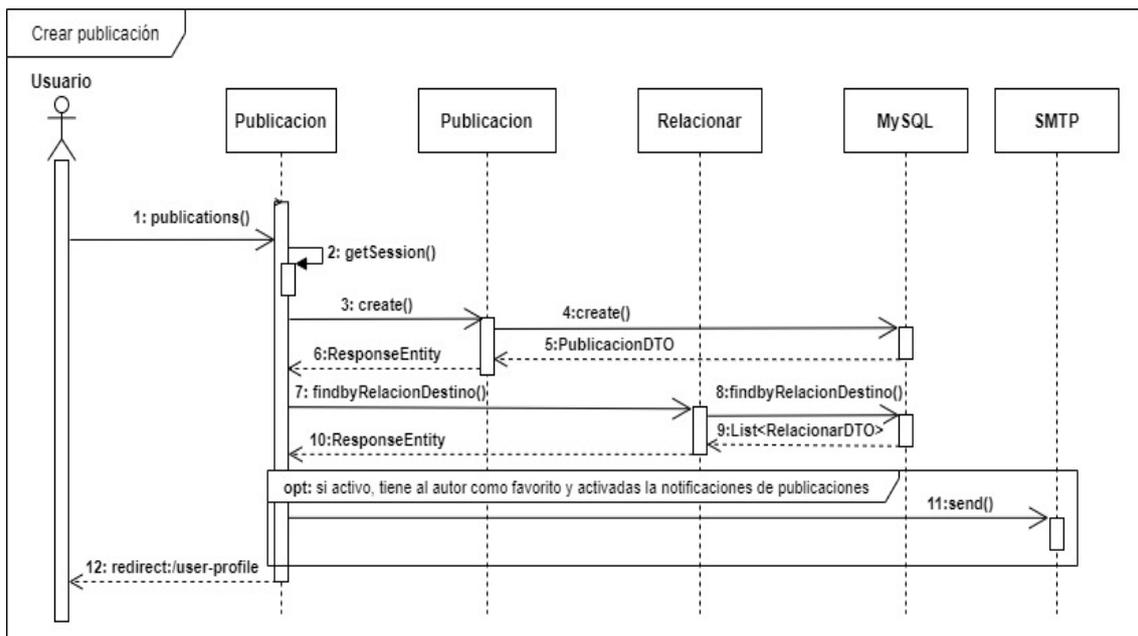


Figura 89. Diagrama de secuencia de crear publicación.

Editar publicación

El diagrama de abajo (ver figura 90), se corresponde con el caso de uso ‘UG12 - Editar publicación’ describiendo las interacciones al modificar una publicación previamente creada.

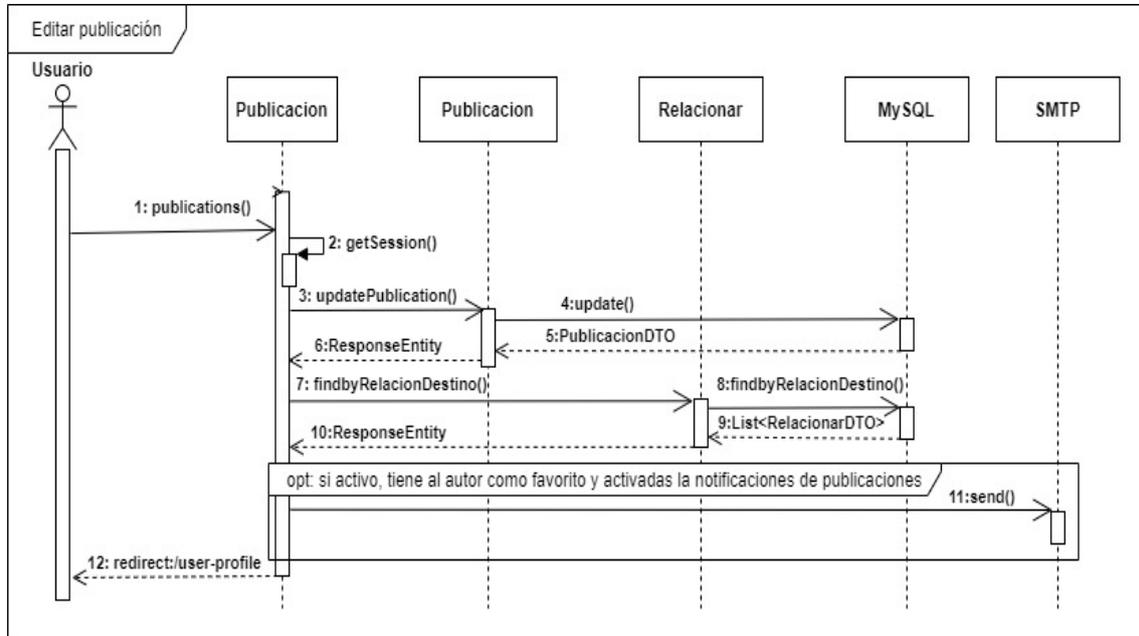


Figura 90. Diagrama de secuencia de editar publicación.

Dar me gusta

El diagrama de abajo (ver figura 91), se corresponde con el caso de uso ‘UG14 - Dar me gusta’ del ‘Usuario general’, que describe las interacciones al dar un like a una publicación.

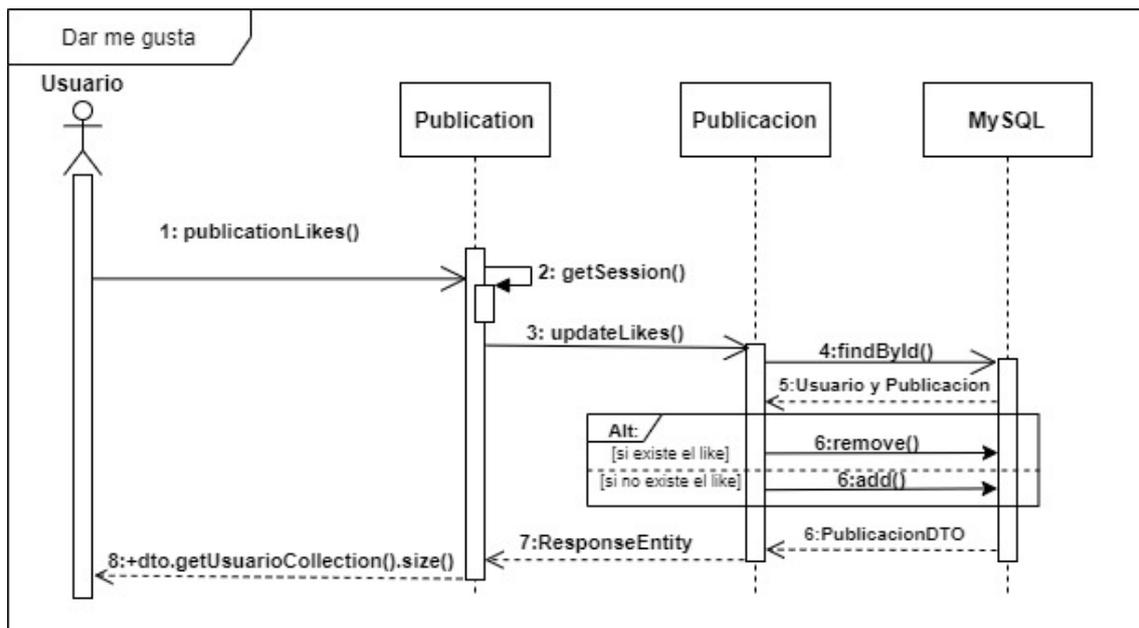


Figura 91. Diagrama de secuencia de dar me gusta.

Buscar perfiles

El diagrama de abajo (ver figura 92), se corresponde con los casos de uso ‘UG16 - Buscar perfiles’ y ‘UG17 - Ver perfiles’, describiendo las interacciones al buscar otros perfiles con un posible uso de filtros.

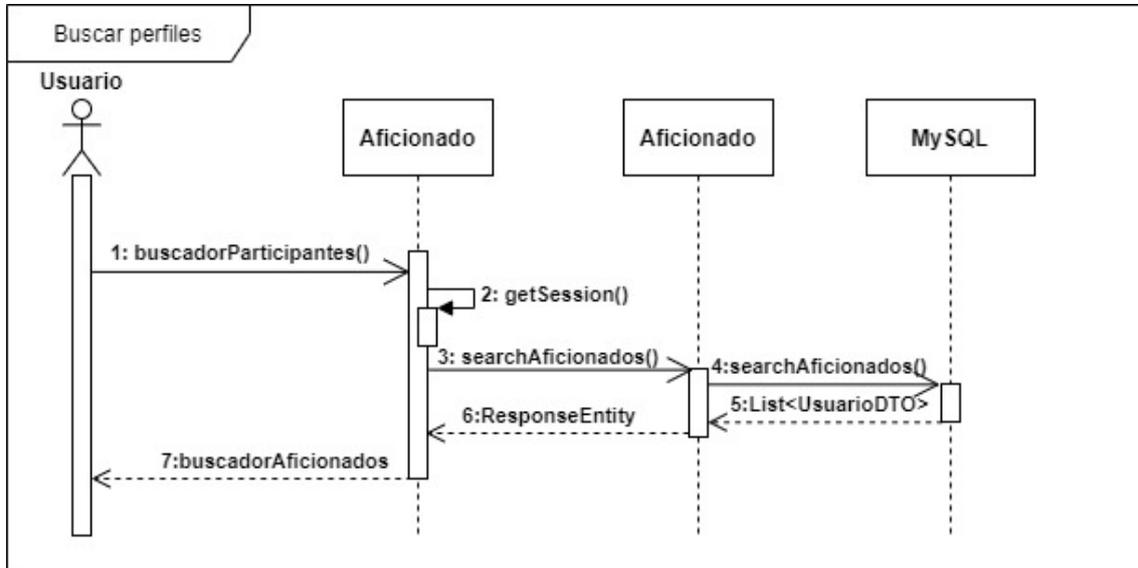


Figura 92. Diagrama de secuencia de buscar perfiles.

Gestionar la inclusión como favorito

El diagrama de abajo (ver figura 93), se corresponde con el caso de uso ‘UG18 - Gestionar la inclusión como favorito’, describiendo las interacciones al gestionar la incorporación como favorito.

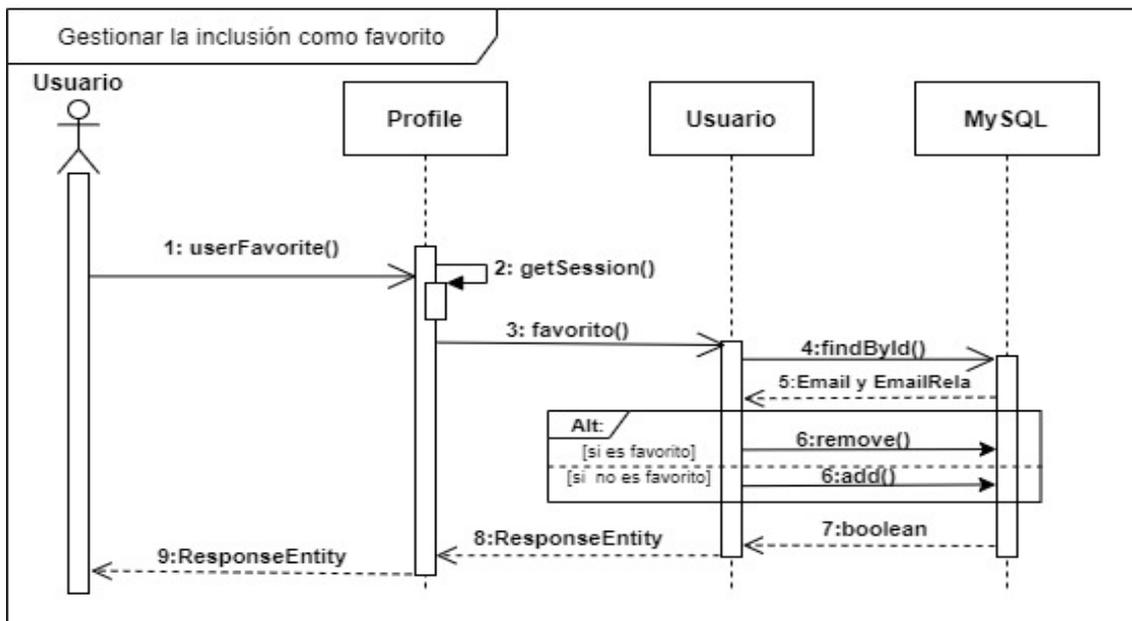


Figura 93. Diagrama de secuencia de gestionar la inclusión como favorito.

Buscar publicaciones

El diagrama de abajo (ver figura 94), se corresponde con los casos de uso 'UG21 - Buscar publicaciones' y 'UG22 - Ver publicaciones', describiendo las interacciones al buscar publicaciones con un posible uso de filtros.

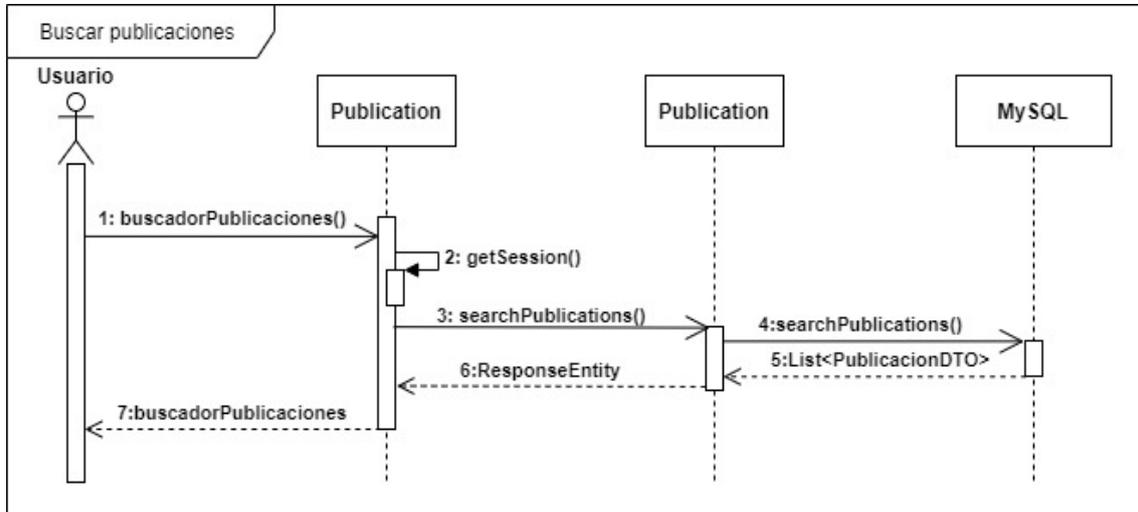


Figura 94. Diagrama de secuencia de buscar publicaciones.

Reportar

El diagrama de abajo (ver figura 95), se corresponde con los casos de uso 'UG19 - Reportar perfil' y 'UG23 - Reportar publicación', describiendo las interacciones al reportar.

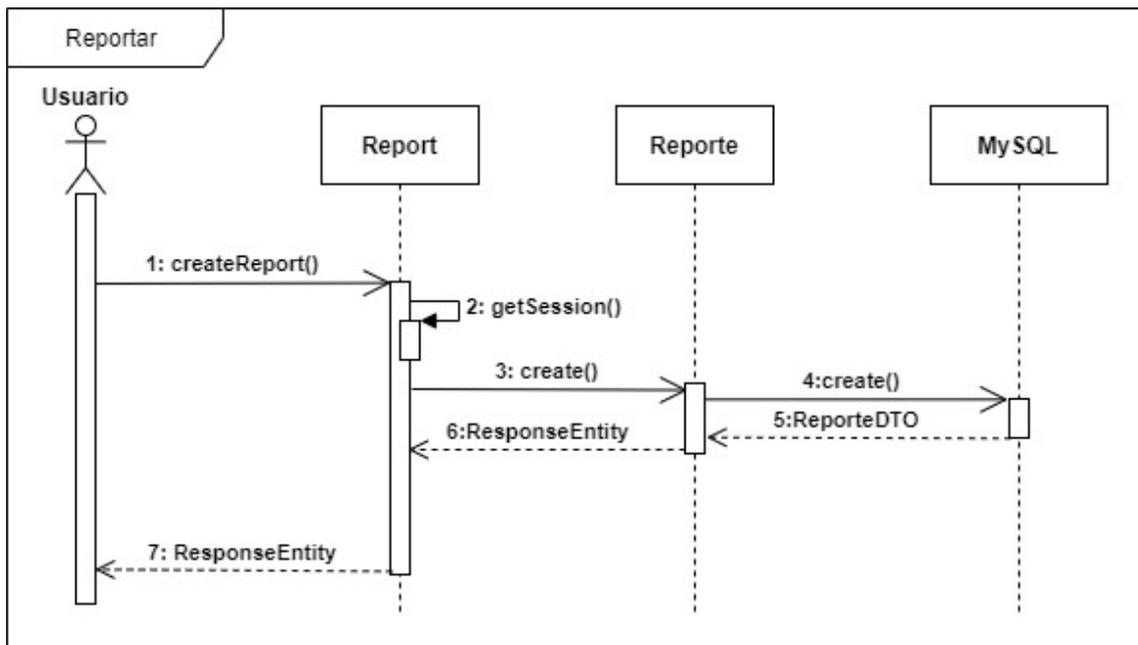


Figura 95. Diagrama de secuencia de reportar.

Acceder al chat

El diagrama de abajo (ver figura 96), se corresponde con los casos de uso ‘UG24 - Acceder al chat’, ‘UG25 - Visualizar mensajes’ y ‘UG26 - Marcar como leído’, describiendo las interacciones al acceder al chat, pudiendo mostrar los mensajes mantenidos con los perfiles, quedando como leídos los mostrados.

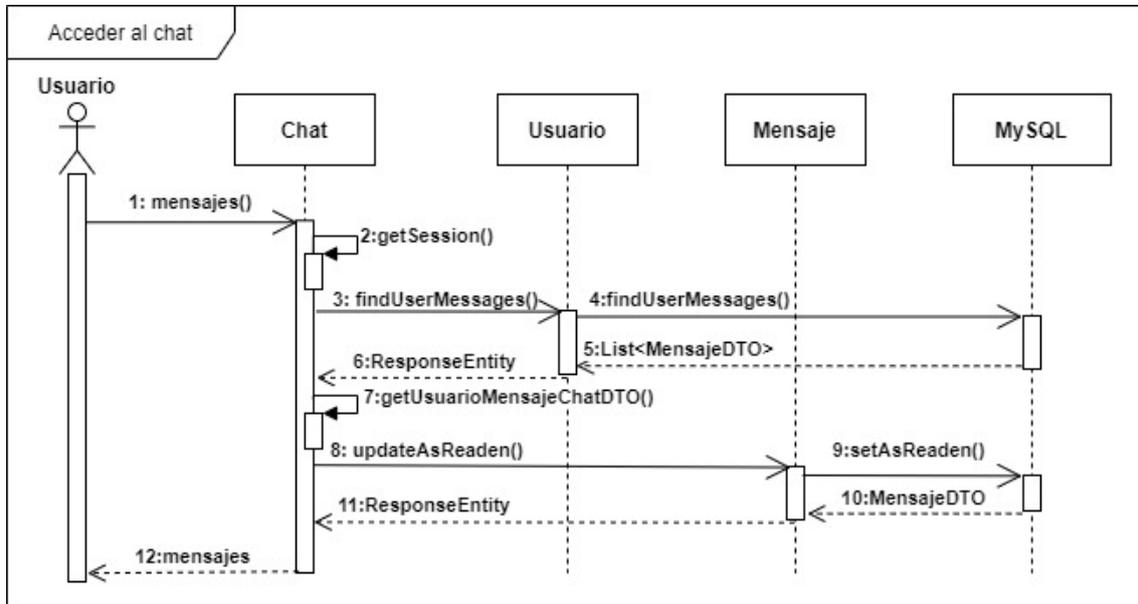


Figura 96. Diagrama de secuencia de acceder al chat.

Escribir mensaje privado al perfil

El diagrama de abajo (ver figura 97), se corresponde con el caso de uso ‘UG20 - Escribir mensaje privado al perfil’, describiendo las interacciones al enviar un mensaje privado al perfil.

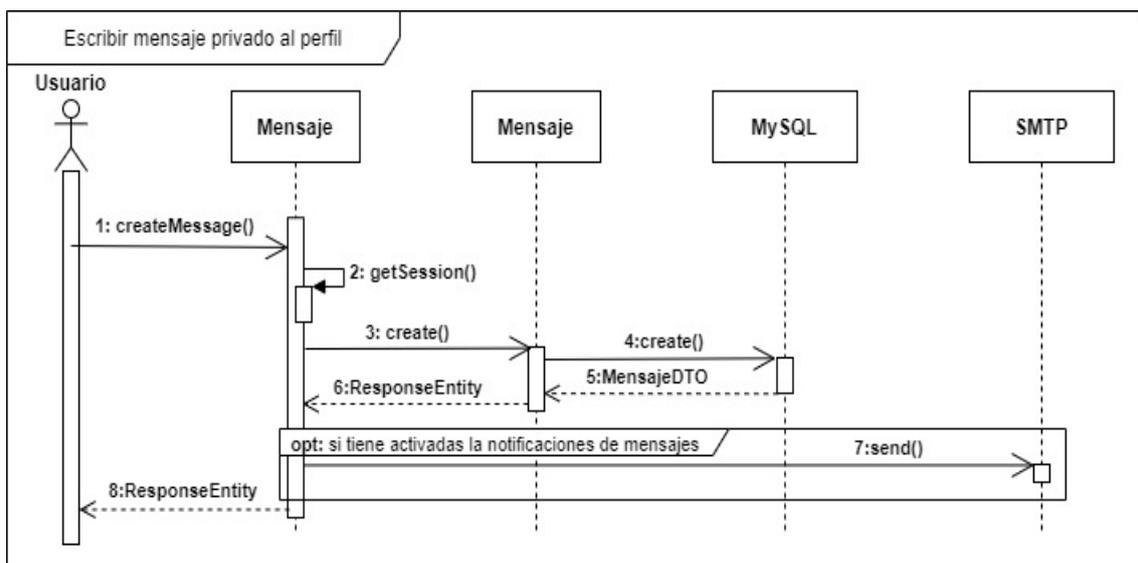


Figura 97. Diagrama de secuencia de escribir mensaje privado al perfil.

Buscar perfil reportado

El diagrama de abajo (ver figura 98), se corresponde con los casos de uso 'UA1 - Buscar perfil reportado' y 'UG17 - Ver perfiles', describiendo las interacciones a la búsqueda de perfiles reportados con el posible uso de los filtros para reportes.

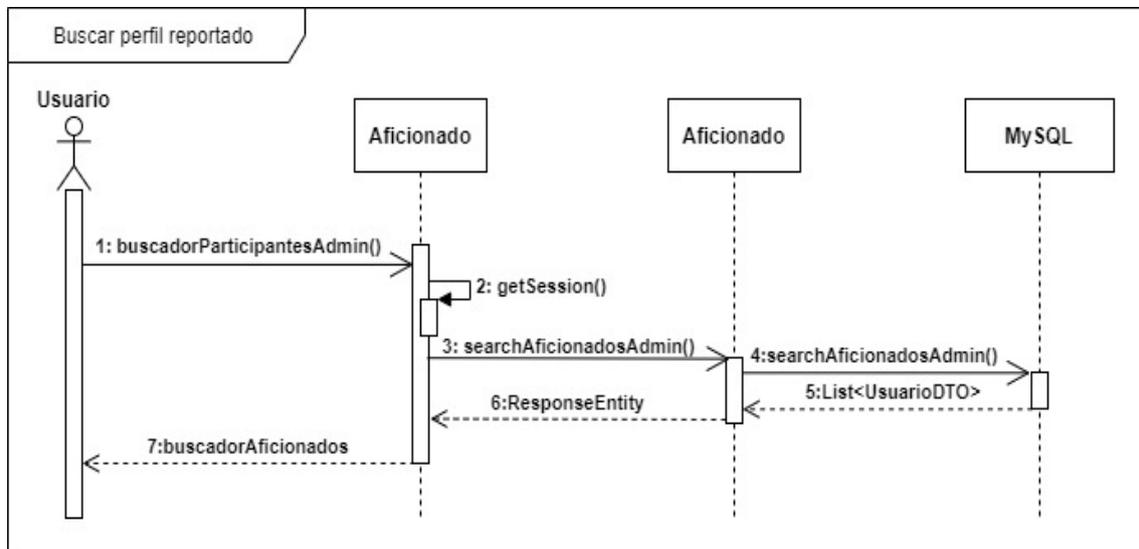


Figura 98. Diagrama de secuencia de buscar perfil reportado.

Buscar publicación reportada

El diagrama de abajo (ver figura 99), se corresponde con los casos de uso 'UA2 - Buscar publicación reportada' y 'UG22 - Ver publicaciones', describiendo las interacciones a la búsqueda de publicaciones reportadas con el posible uso de los filtros para reportes.

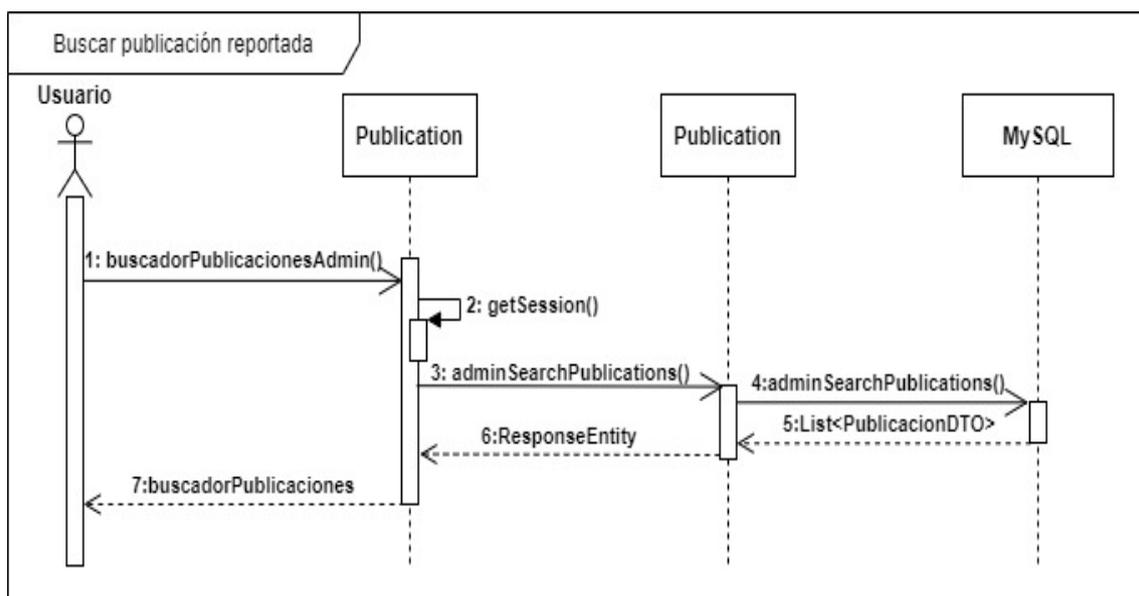


Figura 99. Diagrama de secuencia de buscar publicación reportada.

Visualizar reportes

El diagrama de abajo (ver figura 100), se corresponde con los casos de uso ‘UA3 - Visualizar reportes del perfil’ y ‘UA4 - Visualizar reportes de la publicación’, describiendo las interacciones al mostrar los reportes.

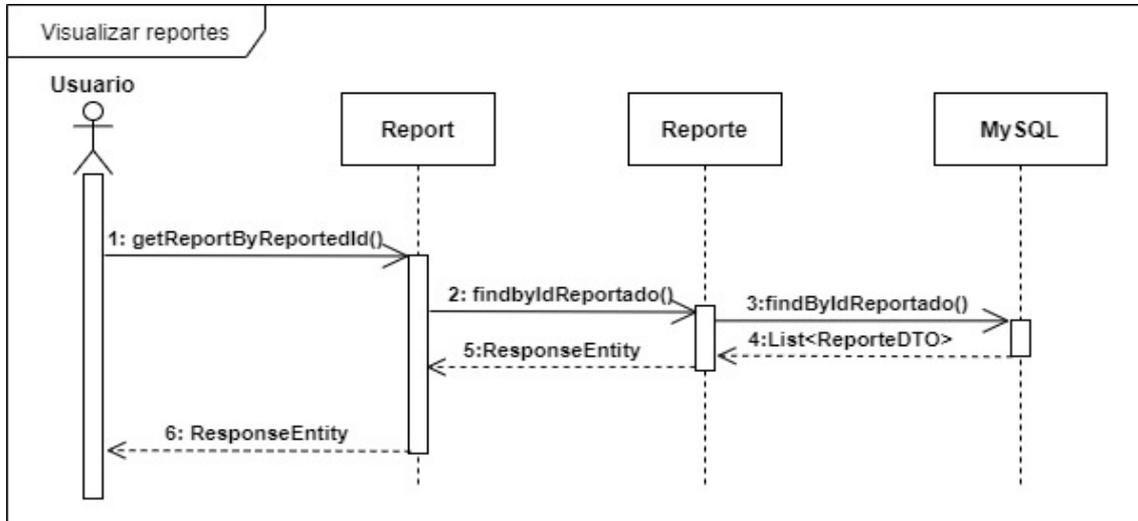


Figura 100. Diagrama de secuencia de visualizar reportes.

Gestionar la activación del perfil

El diagrama de abajo (ver figura 101), se corresponde con los casos de uso ‘UA5 - Gestionar la activación del perfil’ y ‘UA6 - Gestionar la activación de la publicación’, describiendo las interacciones al gestionar la activación determinado perfil de un tercero.

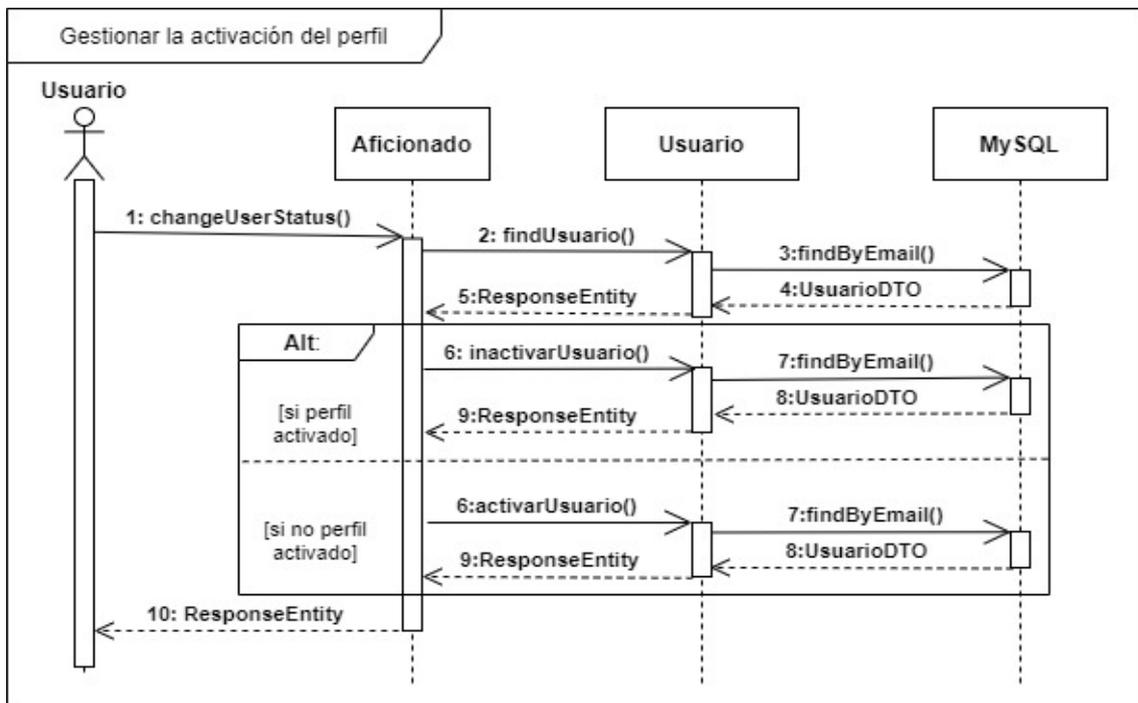


Figura 101. Diagrama de secuencia de gestionar la activación del perfil.

Gestionar la activación de la publicación

El diagrama de abajo (ver figura 102), se corresponde con el caso de uso 'UA6 - Gestionar la activación de la publicación', describiendo las interacciones al gestionar estado de una publicación.

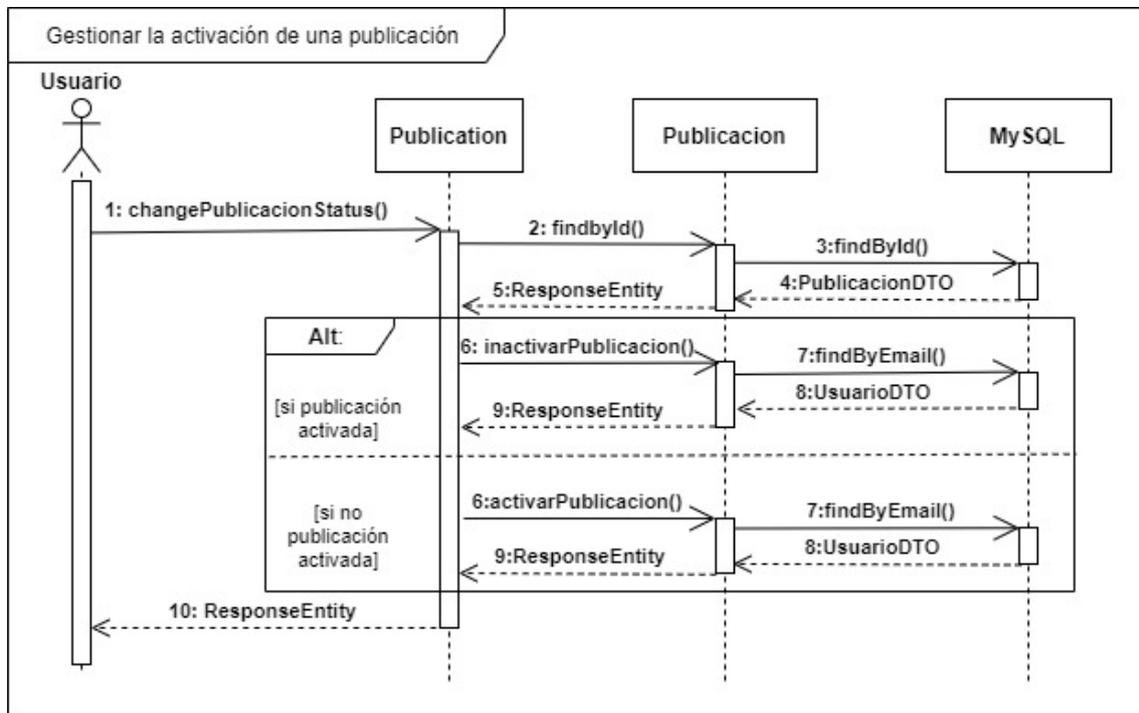


Figura 102. Diagrama de secuencia de gestionar la activación de la publicación.

5.6. Diseño de interfaces

El diseño de interfaces es la parte que abarca la creación y desarrollo de las presentaciones visuales de la aplicación que serán usadas para que los diferentes usuarios interactúen And Social con la aplicación web. Para ello se van a realizar esbozos de las diferentes páginas reflejando el diseño y estructuración partiendo del template HTML, Workwise - The Freelancer And Social Networking [3].

Pese que no haya un único esquema, sí que se puede distinguir tras inicio de sesión: un menú de cabecera, un menú desplegable a la derecha de la cabecera bajo la foto de perfil con el nombre de usuario, un menú central sobre la zona de contenido si se visualiza un perfil, un menú a la izquierda con diferentes opciones en función de la página, siendo el menú de búsqueda diferente para los usuarios administradores, puesto que permite una búsqueda con filtros para encontrar perfiles inactivos y con reportes.

Vista	Registro
Descripción	Permite el registro en el sistema a un usuario no registrado.
Activación	El usuario hace clic sobre 'Registrarme'.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Registrarse en el sistema.

Tabla 47. Esbozo interfaz de registro.

Vista	Login
Descripción	Permite el inicio de sesión al usuario registrado.
Activación	El usuario hace clic sobre 'Iniciar sesión'.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Iniciar sesión en el sistema. • Acceder a home.

Tabla 48. Esbozo interfaz de login.

Vista	Recuperación de contraseña por email
Descripción	Permite la recuperación de la contraseña con el email de registro.
Activación	El usuario hace clic sobre 'Iniciar sesión' y después '¿Olvido la contraseña?'
Esbozo	
Eventos	<ul style="list-style-type: none"> • Recuperar la contraseña mediante el email a través del email de registro. • Acceder a la página de inicio de sesión/registro.

Tabla 49. Esbozo interfaz de recuperación de contraseña por email.

Vista	Home
Descripción	Inicio de la aplicación web. Se corresponde con la opción 'Home' del menú de cabecera.
Activación	El usuario inicia sesión o clic sobre 'Home' del menú de cabecera.
Esbozo	
Eventos	Ninguno.

Tabla 50. Esbozo interfaz de home.

Vista	Info. de mi perfil
Descripción	Permite al usuario la visualización de sus propios datos del perfil.
Activación	El usuario clica 'Mi perfil' del menú desplegable y después 'Info' del menú central de la parte superior de la zona de contenido.
Esbozo	
Eventos	Ninguno.

Tabla 51. Esbozo interfaz de info. de mi perfil.

Vista	Publicaciones de mi perfil
Descripción	Permite al usuario la visualización de sus propias publicaciones.
Activación	El usuario clica 'Mi perfil' del menú desplegable y después 'Publicaciones' del menú central de la parte superior de la zona de contenido.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Editar publicaciones propias.

Tabla 52. Esbozo interfaz de publicaciones de mi perfil.

Vista	Añadidos a favoritos de mi perfil
Descripción	Permite al usuario la visualización de sus usuarios añadidos a favoritos.
Activación	El usuario clica 'Mi perfil' del menú desplegable y después 'Favoritos' del menú central de la parte superior de la zona de contenido.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Quitar un usuario de favoritos. • Escribir un mensaje. • Reportar un perfil.

Tabla 53. Esbozo interfaz de añadidos a favoritos mi perfil.

Vista	Configuración
Descripción	Permite configurar las notificaciones y la privacidad.
Activación	El usuario clica 'Configuración' desde el menú desplegable o pulsa 'Configuración' del menú izquierdo, previo clicado de 'Mi perfil' del menú desplegable.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Gestionar la activación de alguna o todas las opciones configurables.

Tabla 54. Esbozo interfaz configuración.

Vista	Cambio de contraseña
Descripción	Permite cambiar la contraseña actual por otra segura.
Activación	El usuario pulsa ‘Cambio de contraseña’ del menú de la izquierda, previo pulsado de ‘Configuración’ desde el menú desplegable o izquierdo por haber clicado anteriormente ‘Mi perfil’ del menú desplegable.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Cambiar la contraseña. • Limpiar los campos para las contraseñas.

Tabla 55. Esbozo interfaz de cambio de contraseña.

Vista	Eliminación del perfil
Descripción	Permite la desactivación del perfil.
Activación	El usuario pulsa ‘Eliminar perfil’ del menú de la izquierda, previo clicado de ‘Configuración’ desde el menú desplegable o izquierdo por haber clicado anteriormente ‘Mi perfil’ del menú desplegable.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Limpiar las contraseña introducida. • Desactivar el perfil.

Tabla 56. Esbozo interfaz de eliminación del perfil.

Vista	Búsqueda de perfiles
Descripción	Permite la búsqueda de perfiles de usuarios activos mediante filtros.
Activación	El usuario clicla la opción 'Aficionados' del menú de cabecera.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Limpiar los filtros. • Configurar filtros. • Buscar en función de los filtros. • Añadir a favoritos. • Enviar un mensaje. • Reportar un perfil. • Ver perfil con más detalle.

Tabla 57. Esbozo interfaz de búsqueda de perfiles.

Vista	Búsqueda de publicaciones
Descripción	Permite la búsqueda de publicaciones de usuarios mediante filtros.
Activación	El usuario pulsa la opción 'Publicaciones' del menú de cabecera.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Limpiar los filtros. • Configurar filtros. • Buscar en función de los filtros. • Enviar un mensaje al perfil creador de la publicación. • Dar al like. • Reportar una publicación. • Ver perfil con más detalle. (Al clicar la foto y el nombre).

Tabla 58. Esbozo interfaz de búsqueda de publicaciones.

Vista	Mensajes
Descripción	Permite chat de mensajes privados mantenidos con usuarios.
Activación	El usuario pulsa la opción 'Mensajes' del menú de cabecera.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Ver el chat mantenido con un usuario. • Enviar mensajes. • Ver perfil aficionado con detalle. (Al clicar la foto y el nombre).

Tabla 59. Esbozo interfaz de mensajes.

Vista	Info. ver perfil ajeno con detalle
Descripción	Permite visualizar la información detallada de un perfil ajeno.
Activación	El usuario clica primeramente 'Ver perfil' de un perfil poco detalle, el nombre o foto de perfil de una publicación o de su conversación de chat, y después sobre 'Info' del menú central superior de la zona de contenido.
Esbozo	
Eventos	Ninguno.

Tabla 60. Esbozo interfaz de info. de perfil ajeno.

Vista	Publicaciones de un perfil ajeno
Descripción	Permite visualizar las publicaciones de un perfil ajeno.
Activación	El usuario clica primeramente la opción ‘Ver perfil’ de un perfil poco detalle, el nombre o foto de perfil de una publicación o de su conversación de chat, y después sobre ‘Publicaciones’.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Reportar una publicación. (En caso de que tenga).

Tabla 61. Esbozo interfaz de publicaciones de perfil ajeno.

Vista	Creación y edición de una publicación
Descripción	Ventana modal que permite la creación y edición de publicaciones.
Activación	El usuario clica ‘Crear publicación’ del menú izquierdo, previo clic en ‘Mi perfil’ del menú desplegable, u otra posibilidad es que clique ‘Edición’ sobre una publicación, después de haber clicado ‘Mi perfil’ y ‘Publicaciones’.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Seleccionar opciones sobre la publicación y añadir descripción. • Cancelar la posibilidad de publicar o editar. • Publicar la publicación creada o editada.

Tabla 62. Esbozo interfaz de creación y edición de publicación.

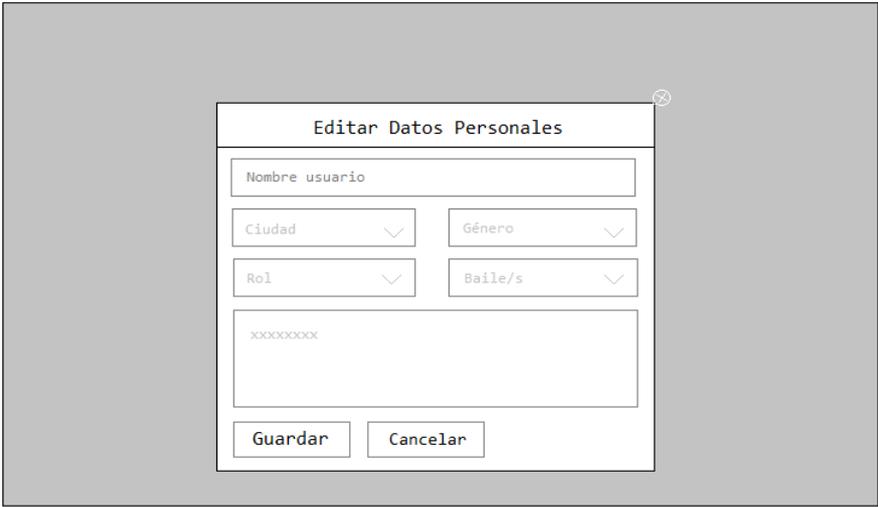
Vista	Edición de perfil
Descripción	Ventana modal que permite la edición del propio perfil.
Activación	El usuario clica ‘Modificar perfil’ del menú izquierdo, previo clic en ‘Mi perfil’ del menú desplegable.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Seleccionar opciones sobre el perfil y modificar la descripción. • Cancelar la posibilidad de editar el perfil. • Guardar lo editado del propio perfil.

Tabla 63. Esbozo interfaz de edición de perfil.

Vista	Envío de mensaje
Descripción	Ventana modal que permite el envío de mensajes.
Activación	El usuario clica sobre el sobre del perfil poco detallado o de su publicación, o clica viendo el perfil con detalle sobre ‘Enviar mensaje’ del menú de la izquierda.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Cancelar el mandar un mensaje. • Enviar el mensaje escrito.

Tabla 64. Esbozo interfaz de envío de mensaje.

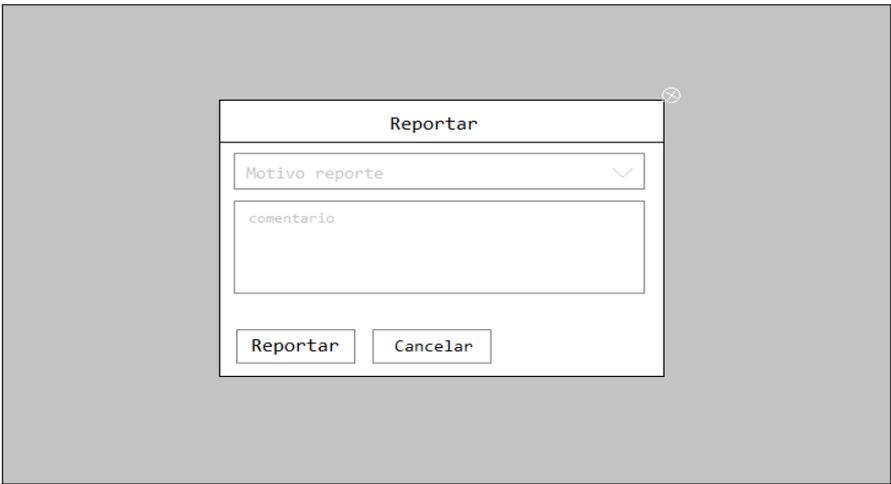
Vista	Creación de reporte
Descripción	Ventana modal que permite reportar un perfil o una publicación.
Activación	El usuario clica sobre el triángulo con exclamación, ya sea de un perfil o una publicación. En caso ver un perfil ajeno con detalle se puede reportar desde el menú izquierdo de perfil o desde la publicación.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Seleccionar opciones sobre sobre reportes y añadir comentario. • Cancelar la posibilidad de reportar. • Reportar el perfil o la publicación.

Tabla 65. Esbozo interfaz de creación de reporte.

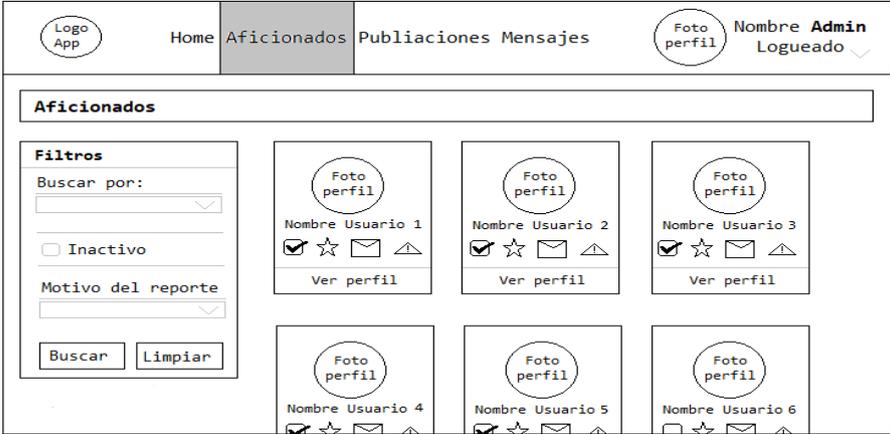
Vista	Búsqueda de aficionados para administradores
Descripción	Permite la búsqueda de perfiles para usuarios administradores.
Activación	El administrador clica la opción 'Aficionados' del menú de cabecera.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Limpiar los filtros. • Configurar filtros. • Buscar en función de los filtros. • Añadir a favoritos. • Enviar un mensaje. • Ver los reportes del perfil. • Ver perfil con más detalle. • Gestionar la activación de un perfil.

Tabla 66. Esbozo interfaz de búsqueda de aficionados para administradores.

Vista	Búsqueda de publicaciones para administradores
Descripción	Permite la búsqueda de publicaciones para usuarios administradores.
Activación	El administrador clicla la opción ‘Publicaciones’ del menú de cabecera.
Esbozo	
Eventos	<ul style="list-style-type: none"> • Limpiar los filtros. • Configurar filtros. • Buscar en función de los filtros. • Enviar un mensaje al perfil creador de la publicación. • Dar al like. • Ver los reportes de una publicación. • Ver perfil con más detalle. (Al clicar la foto y el nombre).

Tabla 67. Esbozo interfaz de búsqueda de publicaciones para administradores.

Vista	Visualización de reportes asociados para administradores
Descripción	Ventana modal que permite la visualización de reportes.
Activación	El administrador clicla sobre el triángulo de un perfil poco detallado o una publicación correspondiente para visualizar los reportes asociados. (Esto no se puede hacer desde la visualización detallada).
Esbozo	
Eventos	<ul style="list-style-type: none"> • Cancelar la visualización.

Tabla 68. Esbozo interfaz de visualización de reportes para administradores.

6. Implementación de la aplicación

En esta sección se presentan las tecnologías y herramientas empleadas en la totalidad de la aplicación *Bailando en pareja*. Sin pasar por alto las motivación de las decisiones y las complicaciones más relevantes encontradas en la conversión del proyecto diseñado al producto final.

6.1. Implementación de WebMVC y ServiciosREST

WebMVC es una aplicación web implementada bajo el patrón MVC (Modelo-Vista-Controlador). *ServiciosRest* es un servicio con interfaz basado en el protocolo HTTP, desarrollado para brindar a la aplicación las cuatro operaciones básicas de consulta, creación, edición y eliminación de información de los aficionados, en la base de datos.

Para la implementación de ambos proyectos se hace uso de Spring Initializr por la simplificación del proceso, aportando una API extensible capaz de generar proyectos basados en JVM, inspeccionando los metadatos para generar proyectos, listando las dependencias y las versiones disponibles.

La inicialización de ambos proyectos se ha hecho a través de la siguiente URL: <https://start.spring.io> [4], puesto que el servicio extrae las dependencias que se necesitan para la aplicación haciendo la mayor parte de la configuración. Se ha seleccionado Maven, el lenguaje Java sumado a las dependencias Spring Web, Thymeleaf (solo para *WebMVC*) y Spring Boot DevTools. Generando así un archivo de aplicación web configurado con las elecciones, el cual se descarga como un zip, para después integrarle desde el IDE.

La motivación de usar Maven proviene por el fácil acceso que da la información esencial, permite la incorporación de nuevas dependencias mediante la introducción de código de dependencias nuevo en el archivo POM, así como que el proceso de construcción del proceso es simple y bien organizado.

6.2. Implementación de la bases de datos

La base de datos es fundamental para el almacenamiento de información de manera organizada y accesible. Para ello, se hace uso de MySQL Workbench en la versión 8.0.30 por ser una herramienta visual que permite modelar, diseñar e incluso administrar bases de datos MySQL. Siendo una herramienta muy completa, además de ser multiplataforma.

MySQL es un sistema gestor de bases de datos, el cual permite su utilización con diferentes motores de almacenamiento. Los motores de almacenamiento son relevantes de cara al diseño de las transacciones, afectando a la integridad y a las características del

sistema. Por lo que se ha elegido motor de almacenamiento InnoDB debido a la preferencia por una base de datos relacional, a la posibilidad de poder revertir los cambios en caso de error garantizando así la persistencia de los datos y a la garantía de la integridad de las tablas por las características ACID (que en español significa Atomicidad, Consistencia, Aislamiento y Durabilidad en español).

6.3. Principales tecnologías usadas

Las principales tecnologías usadas para la implementación del diseño lógico se exponen a continuación:

Java JDK 17

Java es un lenguaje de programación orientado a objetos y de alto nivel. Java SE Development Kit [5] (java edición estándar con un conjunto de productos de desarrollo) es una herramienta esencial para el desarrollo de Java, con un entorno para crear y ejecutar aplicaciones de java en diversas plataformas.

Spring Boot

Spring Boot [6] es una tecnología de código abierto para la creación aplicaciones Java empresariales de una manera más veloz y simple, basándose en el marco de trabajo o framework Spring. Spring boot, destaca por: integrarse con Spring framework, facilitar la configuración, embeber servidores, soportar crear microservicios y múltiples fuentes de datos, además de un ecosistema entre otras características.

Thymeleaf

Thymeleaf [7] es un motor de plantillas usado en el desarrollo web, integrable en Java. Su cometido es hacer más sencilla la creación de vistas dinámicas, posibilitando que la presentación y la lógica de negocio se combinen de forma eficiente y clara, permitiendo la integración dentro del patrón MVC.

Bootstrap

Bootstrap [8] es un marco de trabajo de código abierto para crear sitios web de diseño adaptable a la diferentes pantallas como pueden ser las de los dispositivos móviles. Para ofrecer funciones y componentes relacionados con el diseño web haciendo uso de JavaScript, HTML y CSS.

JavaScript, HTML 5 y CSS 3

JavaScript [9], lenguaje de programación de alto nivel muy empleado en el desarrollo web aportando a los sitios web interactividad y dinamismo. Lenguaje de scripting que está orientado a objetos y se ejecuta navegador web del usuario, siendo fundamental para la interactividad de las aplicaciones web.

HTML 5 [9], versión quinta del lenguaje de marcadores de hipertexto usado para la definición de la estructura, el diseño y contenido de páginas web definiendo los contenidos de todo tipo de material multimedia.

CSS 3 [9], versión última de la tecnología de hojas de estilo en cascada, usada para la definición del diseño y del aspecto de los elementos de una página web aportando un mejor control sobre la presentación del contenido.

Maven

Maven [10] es proyecto de código abierto que ofrece un framework para la gestión y construcción de proyectos software, siendo algo más que una herramienta de compilación. Cada proyecto tiene su POM, es decir, su modelo de objetos del proyecto con información relevante sobre este como es su configuración, dependencias, etc. Siendo útil una herramienta útil para creación de un proceso de construcción simple y bien organizado.

REST

REST o Representational State Transfer [11], (transferencia de estado representacional) es una arquitectura de software empleada para el intercambio de información en la Worl Wide Web y otros entornos. En la arquitectura REST los recursos se identifican mediante URLs, permitiendo las operaciones CRUD, el cliente puede hacer solicitudes de peticiones en la que cada recurso posee una representación ya sea en forma de texto, XML, JSON, HTML, etc. Al no tener estado cada solicitud debe contener toda la información para su proceso.

Una forma de implementar este tipo de arquitectura mediante el lenguaje de programación Java es a través del uso de la librería Jax-RS, la cual está disponible de forma pública y es fácilmente aplicable mediante Maven y da soporte el diseño y programación de servicios web de forma estándar y sencilla.

JSON

JSON o JavaScript Object Notation [12], (notación de objeto JavaScript) es un formato estructurado y liviano para intercambiar datos, el cual es tanto fácil de leer y escribir para las personas como para los ordenadores. Posee una combinación de nombre y valor que da lugar a una lista ordenada de valores.

XML

XML o Extensible Markup Language [13], (lenguaje de marcas extensible) es un lenguaje marcado que aporta reglas para la definición de los datos, así como el almacenaje de datos de para compartir con otros posibles sistemas informáticos.

MySQL

MySQL [14] es una aplicación de código abierto para definir, crear y mantener una base de datos, es decir, un sistema de gestión de bases de datos. InnoDB es el nombre de la motor de almacenamiento elegido característico por su integridad referencial soporte de transacciones de tipo ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad en español).

Tomcat

Tomcat o Apache Tomcat [15] es un servidor web de código abierto útil para la ejecución de aplicaciones Java. Permite la recepción de peticiones HTTP generando contenido dinámico como respuesta. Además, permite configurar y gestionar el servidor mediante una consola de administración web.

7.4. Herramientas utilizadas

Las herramientas más relevantes se recogen seguidamente por haber sido utilizadas en el desarrollo del proyecto y/o en la elaboración de la memoria:

- **Draw.io**: la herramienta para la realización de diagramas se ha empleado para la realización del modelo entidad-relación, arquitectura lógica, arquitectura física, diagramas de casos de uso y diagramas de secuencia.
- **Microsoft Office 365 Word**: el software procesador de textos se ha usado para creación de la memoria del trabajo de fin de grado.
- **Microsoft Office 365 PowerPoint**: el software creador de diapositivas digitales presentaciones se ha empleado para la presentación del trabajo de fin de grado.
- **Google Chrome/Microsoft Edge/Opera**: los navegadores web, de los cuales se ha hecho uso a la hora de ejecutar y probar la aplicación web.
- **Eclipse IDE for Enterprise Java and Web Developers Version: 2022-09**: es un entorno de desarrollo integrado con herramientas para el desarrollo de software con el que se ha desarrollado los dos proyectos de la aplicación web.

- **Apache NetBeans IDE 15:** otro entorno de desarrollo integrado para el desarrollo de software empleado para la creación de las clases entidad y cliente, además de para crear los diagramas de clases.
- **MySQL WorkBench 8.0 CE:** herramienta usada para el desarrollo de y administración de la base de bases de datos, también usada para la realización del modelo entidad-relación.
- **Postman:** herramienta para el desarrollo de APIs, se ha empleado para el probar los servicios REST del proyecto de servicios.
- **Tomsplanner:** aplicación web para la planificación de proyectos, se ha manejado para la elaboración un diagrama de Gantt.
- **Microsoft Paint:** aplicación con herramientas para el dibujo y edición de imágenes con la que se han diseñado las interfaces.

7. Pruebas

Las pruebas de caja negra se realizan para determinar si la aplicación *Bailando en pareja* cumple con la funciones que debe realizar, sin tener en cuenta que hace el sistema por dentro. Es por ello, que en los siguientes apartados se van a reflejar las pruebas más relevantes, llevadas a cabo después de cada iteración (ver apartado 3.1.).

7.1. Pruebas de iteración I

Las siguientes pruebas han sido realizadas en la primera iteración:

ID	P1.1
Descripción	Un usuario se registra en el sistema.
Prerrequisito	Ese usuario no puede haberse registrado previamente.
Entrada	Nombre completo, email, contraseña segura, contraseña repetida y fecha de nacimiento y clic sobre 'Registrarme'.
Salida	Se muestra por pantalla 'Usuario registrado correctamente', quedando el usuario registrado en la base de datos, de lo contrario muestra un error en el campo correspondiente, posibilitando el volver a empezar de nuevo.
Test	Superado.

Tabla 69. Prueba P1.1.

ID	P1.2
Descripción	Un usuario inicia sesión en el sistema.
Prerrequisito	Ese usuario tiene que haberse registrado previamente y estar activo.
Entrada	Email y contraseña, más el clic sobre 'Iniciar sesión'
Salida	En caso de credenciales correctos redirigirá a la página home, de lo contrario indicará 'Usuario o contraseña no válida', siendo posible volver a empezar.
Test	Superado.

Tabla 70. Prueba P1.2.

ID	P1.3
Descripción	Un usuario cierra sesión en el sistema.
Prerrequisito	Ese usuario tiene que haberse registrado previamente y haber iniciado sesión.
Entrada	Clic sobre 'Salir' del menú desplegable.
Salida	Cierre de sesión y redirección a la página de inicio de sesión.
Test	Superado.

Tabla 71. Prueba P1.3.

ID	P1.4
Descripción	Un usuario se quiere registrarse estando ya registrado.
Prerrequisito	Ese usuario ya tiene que estar registrado.
Entrada	Nombre completo, email ya introducido en el registro anterior, contraseña segura, contraseña repetida y fecha de nacimiento de un mayor de edad, más el clic sobre 'Registrarme'.
Salida	Se muestra por pantalla 'Error registrando usuario'.
Test	Superado.

Tabla 72. Prueba P1.4.

ID	P1.5
Descripción	Un usuario inicia sesión en el sistema con credenciales incorrectas.
Prerrequisito	Ese usuario tiene que haberse registrado previamente y estar activo.
Entrada	Email y contraseña, más el clic sobre 'Iniciar sesión'.
Salida	Se muestra por pantalla 'Usuario o contraseña no válida'.
Test	No superado. No indica el error, pese a no crear un nuevo registro.

Tabla 73. Prueba P1.5.

ID	P1.5.1
Descripción	Un usuario inicia sesión en el sistema con credenciales incorrectas.
Corrección	Fallo en poner el mensaje en la excepción del método 'loginUser' del controlador de <i>WebMVC.1</i>
Prerrequisito	Ese usuario tiene que haberse registrado previamente y estar activo.
Entrada	Email y contraseña, más el clic sobre 'Iniciar sesión'.
Salida	Se muestra por pantalla 'Usuario o contraseña no válida'.
Test	Superado.

Tabla 74. Prueba P1.5.1.

7.2. Pruebas de iteración II

Las siguientes pruebas han sido realizadas en la segunda iteración:

ID	P2.1
Descripción	Un usuario ve su propio perfil.
Prerrequisito	Estar registrado, iniciar sesión y estar activo.
Entrada	Clic sobre 'Mi perfil'.
Salida	Muestra el perfil detallado con los datos del propio perfil.
Test	Superado.

Tabla 75. Prueba P2.1.

ID	P2.2
Descripción	Un usuario no administrador elimina su perfil.
Prerrequisito	Estar registrado, iniciar sesión, estar activo y no ser administrador.
Entrada	Clic sobre 'Eliminar perfil' e introducción de la contraseña, más el clic sobre 'Eliminar perfil'.
Salida	Solo aparece en caso no ser administrador. En caso de meter una contraseña incorrecta da error y en caso de meterla correctamente redirige a la página de inicio de sesión, cambiado el estado de uno a cero en la base de datos.
Test	Superado.

Tabla 76. Prueba P2.2.

ID	P2.3
Descripción	Un usuario cambia su contraseña.
Prerrequisito	Estar registrado, iniciar sesión y estar activo.
Entrada	Clic sobre 'Cambio de contraseña' e introducción de la contraseña antigua y la nueva repetida, más el clic sobre 'Cambiar contraseña'.
Salida	En caso de que alguna de las contraseñas no sea exactamente la antigua, la nueva no sea segura, la nueva y la repetición no sea coincidente lo indicará permitiendo volver a intentarlo. Pero en caso de que todo sea correcto cambiará la contraseña en la base de datos.
Test	Superado.

Tabla 77. Prueba P2.3.

ID	P2.4
Descripción	Un usuario administrador no tiene la opción eliminar perfil.
Prerrequisito	Estar registrado, iniciar sesión, estar activo y ser administrador.
Entrada	Hacer clic sobre la opción 'Configuración'.
Salida	No aparece la opción 'Eliminar perfil'.
Test	Superado.

Tabla 78. Prueba P2.4.

ID	P2.5
Descripción	Un usuario visualiza su configuración y la cambia.
Prerrequisito	Estar registrado, iniciar sesión, estar activo.
Entrada	Hacer clic sobre la opción 'Configuración' y mover el switch.
Salida	Visualización de la configuración, y movimiento del switch con el correspondiente registro creado o eliminado.
Test	No superado, no se mueve el <i>switch</i> , y no crea un registro con la configuración.

Tabla 79. Prueba P2.5.

ID	P2.6.1
Descripción	Un usuario visualiza su configuración y la cambia.
Corrección	Los métodos 'associateUser' y 'disassociateUser' del controlador de <i>ServiciosREST</i> .
Prerrequisito	Estar registrado, iniciar sesión, estar activo.
Entrada	Hacer clic sobre la opción 'Configuración' y mover el switch.
Salida	Visualización de la configuración, y movimiento del switch con el correspondiente registro creado o eliminado.
Test	Superado.

Tabla 80. Prueba P2.4.6.

ID	P2.7
Descripción	Un usuario cambia pone una foto de perfil nueva.
Prerrequisito	Estar registrado, iniciar sesión, estar activo.
Entrada	Hacer clic sobre la opción la cámara de la foto de perfil grande y subir una foto.
Salida	Aparece una nueva foto de perfil, a no ser que de muestre error por no ser '.png' o ser muy pesada.
Test	Superado.

Tabla 81. Prueba P2.7.

ID	P2.8
Descripción	Un usuario cambia su datos del perfil.
Prerrequisito	Estar registrado, iniciar sesión, estar activo.
Entrada	Hacer clic sobre la opción 'Modificar perfil' y se cambian datos, más clic sobre la opción 'Guardar'.
Salida	Se visualizan los datos modificados, que son los que se han registrado en la base de datos.
Test	Superado.

Tabla 82. Prueba P2.8.

ID	P2.9
Descripción	Un usuario recupera su contraseña por mail.
Prerrequisito	Estar registrado, iniciar sesión, estar activo.
Entrada	Hacer clic sobre la opción '¿Olvido contraseña?', introducción del correo de registro.
Salida	Recepción de un correo con la contraseña actual y la recomendación de cambio.
Test	Superado.

Tabla 83. Prueba P2.9.

7.3. Pruebas de iteración III

Las siguientes pruebas han sido realizadas en la tercera iteración:

ID	P3.1
Descripción	Un usuario crea una publicación.
Prerrequisito	Tener menos de 3 publicaciones vigentes por perfil creador.
Entrada	Clic sobre 'Crear publicación', selección de tipo, ciudad, fecha inicio, fecha fin, baile e introducción de posible URL y descripción, más el clic sobre 'Publicar'.
Salida	Visualización de la nueva publicación y recepción de un email en caso de los que han hecho favorito al autor, así lo tengan configurado, además de la creación en la base de datos.
Test	Superado.

Tabla 84. Prueba P3.1.

ID	P3.2
Descripción	Un usuario modifica su publicación.
Prerrequisito	Tener menos de 3 publicaciones vigentes por perfil creador.
Entrada	Clic sobre el icono de modificar la publicación, modificación de cualquier campo de la publicación 'Publicar'.
Salida	Visualización de la publicación modificada y recepción de un email en caso de los que han hecho favorito al autor, así lo tengan configurado, además de la modificación en la base de datos.
Test	Superado.

Tabla 85. Prueba P3.2.

ID	P3.3
Descripción	Un usuario intenta crear una publicación con fechas vencidas o con fecha fin anterior a la inicial.
Prerrequisito	Tener menos de 3 publicaciones vigentes por perfil creador.
Entrada	Clic sobre 'Crear publicación' o sobre el icono de modificar la publicación, selección de un tipo, ciudad, fecha inicio vencida, fecha fin vencida y anterior a la de inicio, baile e introducción de posible URL y descripción, más el clic sobre 'Publicar'.
Salida	Salta una alerta y no permite la creación o modificación así.
Test	No superado, las fechas tiene que ser vigentes, pero la final puede ser anterior a la de inicio.

Tabla 86. Prueba P3.3.

ID	P3.3.1
Descripción	Un usuario intenta crear una publicación con fechas vencidas o con fecha fin anterior a la inicial.
Corrección	Error solucionado en la comparación de fechas del JavaScript.
Prerrequisito	Tener menos de 3 publicaciones vigentes por perfil creador.
Entrada	Clic sobre 'Crear publicación' o sobre el icono de modificar la publicación, selección de un tipo, ciudad, fecha inicio vencida, fecha fin vencida y anterior a la de inicio, baile e introducción de posible URL y descripción, más el clic sobre 'Publicar'.
Salida	Salta una alerta y no permite la creación o modificación así.
Test	Superado.

Tabla 87. Prueba P3.3.1.

ID	P3.4
Descripción	Un usuario no puede tener más de 3 publicaciones activas.
Prerrequisito	Tener 3 publicaciones vigentes.
Entrada	Ninguna.
Salida	No aparece la opción de crear publicación y no deja editar una publicación caducada saltando una alerta.
Test	Superado.

Tabla 88. Prueba P3.4.

ID	P3.5
Descripción	Un usuario elimina su propia publicación.
Prerrequisito	Ninguno.
Entrada	Clic sobre el icono de eliminar de una publicación.
Salida	La publicación eliminada deja de ser visible.
Test	Superado.

Tabla 89. Prueba P3.5.

ID	P3.6
Descripción	Un usuario crea da al like a una publicación cualquiera.
Prerrequisito	Ninguno.
Entrada	Clic sobre el like de una población y vuelta a dar clic para su eliminación.
Salida	Tras el clic se incrementa en uno el contador de likes, y se crea un registro con el que ha dado al clic, pero al volver a clicar se decrementa dicho contador y se elimina el registro.
Test	Superado.

Tabla 90. Prueba P3.6.

ID	P3.7
Descripción	Un usuario añade a favoritos otro.
Prerrequisito	Ninguno
Entrada	Al hacer clic sobre la estrella de favoritos correspondiente a un perfil este se hace favorito y si se vuelve a pulsar, este deja de serlo.
Salida	Aparece una alerta indicando que se ha incluido o ha dejado de serlo, además en caso de serlo, aparece en el apartado de favoritos al estar registrado así en la base de datos, en caso contrario por dejarlo de ser no aparece en la lista, ni está en el registro.
Test	Superado.

Tabla 91. Prueba P3.7.

ID	P3.8
Descripción	Un usuario no administrador busca otros perfiles usando los filtros de la búsqueda de aficionados.
Prerrequisito	Tiene que haber más usuarios registrados.
Entrada	Se seleccionan filtros y se clica sobre 'Buscar'.
Salida	Los resultados se ajustan a los filtros seleccionados.
Test	Superado.

Tabla 92. Prueba P3.8.

ID	P3.9
Descripción	Un usuario no administrador busca publicaciones usando los filtros de la búsqueda de publicaciones.
Prerrequisito	Tiene que haber publicaciones creadas.
Entrada	Se seleccionan filtros y se clica sobre 'Buscar'.
Salida	Los resultados se ajustan a los filtros seleccionados.
Test	No superado, hay problemas no todos los filtros funcionan.

Tabla 93. Prueba P3.9.

ID	P3.9.1
Descripción	Un usuario no administrador busca publicaciones usando los filtros de la búsqueda de publicaciones.
Corrección	Corrección de la implementación de diferentes métodos implicados en el buen funcionamiento de los filtros.
Prerrequisito	Tiene que haber publicaciones creadas.
Entrada	Se seleccionan filtros y se clica sobre 'Buscar'.
Salida	Los resultados se ajustan a los filtros seleccionados.
Test	No superado, hay problemas no todos los filtros funcionan.

Tabla 94. Prueba P3.9.1.

ID	P3.10
Descripción	Un usuario escribe a otro perfil.
Prerrequisito	Tiene que haber perfiles creados.
Entrada	Se clica sobre el sobre de enviar mensaje, se introduce texto y se clica sobre 'Enviar'.
Salida	Se crea un nuevo mensaje y en caso de que el receptor así lo tenga configurado recibe un email.
Test	Superado.

Tabla 95. Prueba P3.10.

ID	P3.10
Descripción	Un usuario consulta los mensajes mantenidos.
Prerrequisito	El usuario tiene que haber intercambiado mensajes con algún perfil para poder ver algún mensaje.
Entrada	Clic sobre 'Mensajes' y sobre algún usuario.
Salida	Aparece una lista de los perfiles con los que se ha mantenido conversación, apareciendo más arriba los perfiles que ha enviado un mensaje más reciente mente y en caso de que no se hayan visto los mensajes, muestra el número. Se visualizan los mensajes del perfil superior, pero si se selecciona otro perfil, se muestran los mensajes de ese perfil, dejando de aparecer el número de mensajes no vistos, en caso de visualizarse.
Test	No superado, hay problemas con el ordenamiento de los usuarios.

Tabla 96. Prueba P3.10.

ID	P3.10.1
Descripción	Un usuario consulta los mensajes mantenidos.
Corrección	La implementación de la ordenación por fecha no era correcta.
Prerrequisito	El usuario tiene que haber intercambiado mensajes con algún perfil para poder ver algún mensaje.
Entrada	Clic sobre 'Mensajes' y sobre algún usuario.
Salida	Aparece una lista de los perfiles con los que se ha mantenido conversación, apareciendo más arriba los perfiles que ha enviado un mensaje más reciente mente y en caso de que no se hayan visto los mensajes, muestra el número. Se visualizan los mensajes del perfil superior, pero si se selecciona otro perfil, se muestran los mensajes de ese perfil, dejando de aparecer el número de mensajes no vistos, en caso de visualizarse.
Test	Superado.

Tabla 97. Prueba P3.10.1.

7.4. Pruebas de iteración IV

Las siguientes pruebas han sido realizadas en la cuarta iteración:

ID	P4.1
Descripción	Un usuario reporta el perfil de otro usuario.
Prerrequisito	Tiene que haber usuarios registrados.
Entrada	Clic sobre el triángulo asociado a un perfil, selección de un motivo, un posible comentario sobre el reporte, más el clic sobre 'Reportar'.
Salida	Creación de un reporte asociado al perfil en la base de datos.
Test	Superado

Tabla 98. Prueba P4.1.

ID	P4.2
Descripción	Un usuario reporta la población de otro usuario.
Prerrequisito	Tiene que haber usuarios registrados con publicaciones.
Entrada	Clic sobre el triángulo asociado a una publicación, selección de un motivo, un posible comentario sobre el reporte, más el clic sobre 'Reportar'.
Salida	Creación de un reporte asociado al perfil en la base de datos.
Test	Superado.

Tabla 99. Prueba P4.2.

ID	P4.3
Descripción	Un administrador activa y desactiva perfiles.
Prerrequisito	Inicio de sesión de un usuario administrador y tiene que haber usuarios registrados.
Entrada	Selección del cuadrado con 'v' y vuelve a seleccionar.
Salida	Tras la selección del cuadro con 'v', esta letra desaparecerá cambiando el estado del perfil a cero en la base de datos, y en caso de volver a seleccionar volverá aparecer la letra, cambiando estado del perfil en la base de datos a uno. Además, en caso de estar un perfil desactivado, sus publicaciones dejarán de ser visibles incluso para los administradores.
Test	Superado.

Tabla 100. Prueba P4.3.

ID	P4.4
Descripción	Un administrador activa y desactiva publicaciones.
Prerrequisito	Inicio de sesión de un usuario administrador y tiene que haber usuarios registrados con publicaciones.
Entrada	Selección del cuadrado con 'v' y vuelve a seleccionar.
Salida	Tras la selección del cuadro con 'v', esta letra desaparecerá cambiando el estado de la publicación a cero en la base de datos, y en caso de volver a seleccionar volverá aparecer la letra y el estado de la publicación cambiará en la base de datos a uno. Además, en caso de estar un perfil desactivado, sus publicaciones no serán visibles incluso para los administradores.
Test	Superado.

Tabla 101. Prueba P4.4.

ID	P4.5
Descripción	Un usuario administrador y otro que no lo es miran el chat, solo pudiendo ver las conversaciones de los usuarios desactivados el administrador.
Prerrequisito	Tiene que haber usuarios registrados, tienen que haber mantenido conversación con el que se inicia sesión para mirar el chat, y solo los administradores verán los perfiles y las conversaciones de los perfiles desactivados.
Entrada	Clic sobre 'Mensajes'.
Salida	Solo los administradores consiguen ver los perfiles y sus conversaciones.
Test	Superado.

Tabla 102. Prueba P4.5.

ID	P4.6
Descripción	Un administrador busca perfiles reportados.
Prerrequisito	Inicio de sesión de un usuario administrador y tiene que haber perfiles de usuarios reportados.
Entrada	Uso de los filtros y de motivos de reporte, más el clic sobre 'Buscar'.
Salida	Muestra perfiles según los filtros.
Test	Superado.

Tabla 103. Prueba P4.6.

ID	P4.7
Descripción	Un administrador busca publicaciones reportadas.
Prerrequisito	Inicio de sesión de un usuario administrador y tiene que haber publicaciones de usuarios reportadas.
Entrada	Uso de los filtros y de motivos de reporte, más el clic sobre 'Buscar'.
Salida	Muestra publicaciones según los filtros.
Test	Superado.

Tabla 104. Prueba P4.7.

ID	P4.8
Descripción	Un administrador visualiza los reportes de un perfil.
Prerrequisito	Inicio de sesión de un usuario administrador y tiene que haber perfiles de usuarios reportados.
Entrada	Clic sobre el triángulo del perfil poco detallado, según se muestra en la búsqueda, sin acceder a los detalles de un usuario.
Salida	Muestra todos los reportes asociados a ese perfil.
Test	Superado.

Tabla 105. Prueba P4.8.

ID	P4.9
Descripción	Un administrador visualiza los reportes de una publicación.
Prerrequisito	Inicio de sesión de un usuario administrador y tiene que haber publicaciones de usuarios reportadas.
Entrada	Clic sobre el triángulo de la publicación, según se presenta en la búsqueda sin acceder desde el perfil detallado de un usuario.
Salida	Muestra todos los reportes asociados a esa publicación.
Test	Superado.

Tabla 106. Prueba P4.9.

8. Manuales

Los manuales van a contar con la información necesaria para tanto poner en funcionamiento la aplicación *Bailando en pareja*, como poder manejar la misma por sus diferentes usuarios. Dichos manuales se presentan a continuación.

8.1. Manual de instalación y despliegue

El manual de instalación y despliegue contiene los pasos para que se pueda instalar y desplegar la aplicación. Sin embargo, previo a esto, es necesaria la preparación y configuración del servidor.

8.1.1 Preparación del servidor

La aplicación pese a haberse desarrollado sobre un Windows 10, también podría funcionar con otros sistemas operativos dado que Java es multiplataforma. Aunque, Linux es el sistema operativo más extendido en los servidores, por ello se va a detallar seguidamente como sería para Ubuntu Server 20.04.

Hay que instalar un servidor de bases de datos MySQL. Aunque, previamente a esta instalación se tienen que actualizar los paquetes del servidor mediante el comando ‘sudo apt update’. Después, se continúa con la instalación del paquete mysql-server con ‘sudo apt install mysql-server’. Posteriormente, para la protección y configuración de la instalación de MySQL se necesita introducir ‘sudo mysql_secure_installation’. Más adelante, ya habiendo accedido al terminal con el comando ‘mysql -u root -p’, hay que ejecutar el SQL (con la base de datos y sus datos, albergado en el contenido del repositorio, cuya información al respecto se encuentra en el ANEXO IV).

La aplicación requiere de una máquina virtual Java, puesto que Java es usado como lenguaje de programación. Lo que lleva a tener que poner los siguientes comandos para llevar a cabo la instalación: ‘sudo apt install software-properties-common’, ‘sudo add-apt-repository ppa:linuxuprising/java’, ‘sudo apt update’, ‘sudo apt install oracle-java17-installer’ y ‘java -version’.

La instalación de un servidor de aplicaciones no es necesaria debido a que Spring Boot cuenta con el servidor Apache Tomcat.

8.1.2 Configuración del puerto

La configuración del puerto de dominio es relevante en el caso de que donde se aloje el proyecto no use el puerto 8080 y tenga redirigida la salida al exterior.

Spring Boot proporciona muchas configuraciones por defecto, y podría ser que fuera necesario cambiar el puerto por defecto de la aplicación. Por defecto, usa el puerto 8080, aunque este se puede cambiar de varias maneras. Una de ellas es especificar la propiedad `'server.port=????'` en `application.properties`. Otra posibilidad es para cambiar el puerto de dominio es pasando el `server.port` por parámetro con la línea de comando: `'java -jar WebMVC-0.0.1-SNAPSHOT.jar --server.port=????'`.

8.1.3 Despliegue de la aplicación

El despliegue de la aplicación con empaquetado `'jar'` es muy sencillo, tan solo hay que situarse en el directorio en el que se quiere trabajar.

A pesar de la sencillez, debido a que son dos proyectos, se hace necesario el uso de dos terminales, y también la introducción del comando `'sudo apt-get install tmux'`. Ya instalado, se pueden tener dos terminales en una misma pantalla de manera vertical siendo necesaria la pulsación de las teclas `'ctrl'` más `'b'` y más `'%'` o `'ctrl'`, `'b'` y `'"'`. Tendiendo por presente que, para pasar de una terminal a otra, se requiere pulsar las teclas `'ctrl'` más `'b'` y la `'flecha correspondiente'`.

Finalmente, ya se puede introducir la sentencia `'java -jar ServicioRest-0.0.1-SNAPSHOT.jar'` en una terminal y la sentencia `'java -jar WebMVC-0.0.1-SNAPSHOT.jar'` en la otra sentencia y en caso de ser necesario otro puerto distinto al configurado, siempre se puede usar la sentencia `'java -jar WebMVC-0.0.1-SNAPSHOT.jar --server.port=????'`.

8.2. Manual de usuario

El manual de usuario se incluye para ofrecer a los usuarios de la aplicación información sobre el uso de las funcionalidades más relevantes.

Los usuarios, tras introducir la URL de la aplicación web, acceden a esta a través de un navegador web, dónde las dos únicas opciones posibles de interacción con la aplicación web son el registro o el inicio de sesión, concretamente: `'Registrarme'` o `'Iniciar Sesión'`. Dicha selección entre el registro y el inicio de sesión se selecciona en la parte superior derecha de la interfaz. También se encuentra un email en la parte inferior por si alguien necesitase ayuda, tuviera alguna duda o algún problema relacionado con la aplicación o el acceso a esta.

8.2.1 Manual de usuario no registrado

El usuario no registrado solo tiene dos posibilidades. Una es la de escribir un correo al email, que aparece debajo, al pie de página del registro e inicio de sesión, desde su propio correo. Y otra, es la registrarse, creándose así un perfil en la aplicación.

Para el registro, se pide un nombre completo, un email (posteriormente no se puede modificar), una contraseña y la correcta repetición de la misma (ambas, aparte de coincidir, deben contener al menos 8 caracteres, una mayúscula, una minúscula, un número y un carácter especial; la contraseña si se puede modificar posteriormente), además de una fecha de nacimiento (tampoco se puede modificar tras el registro) la cual debe ser correspondiente a la de un mayor de edad para que la aplicación permita el registro. Si tras la correcta introducción de los datos se ha producido el registro, se puede ver en un recuadro verde el texto ‘Usuario registrado correctamente’. Hay que tener presente, que los usuarios al registrarse toman ciertos apartados del perfil por defecto, los cuales son lo más ambiguos e inclusivos posibles, por lo que se recomienda hacer uso de la opción ‘Modificar perfil’ que se accede desde ‘Mi perfil’ explicada más adelante.

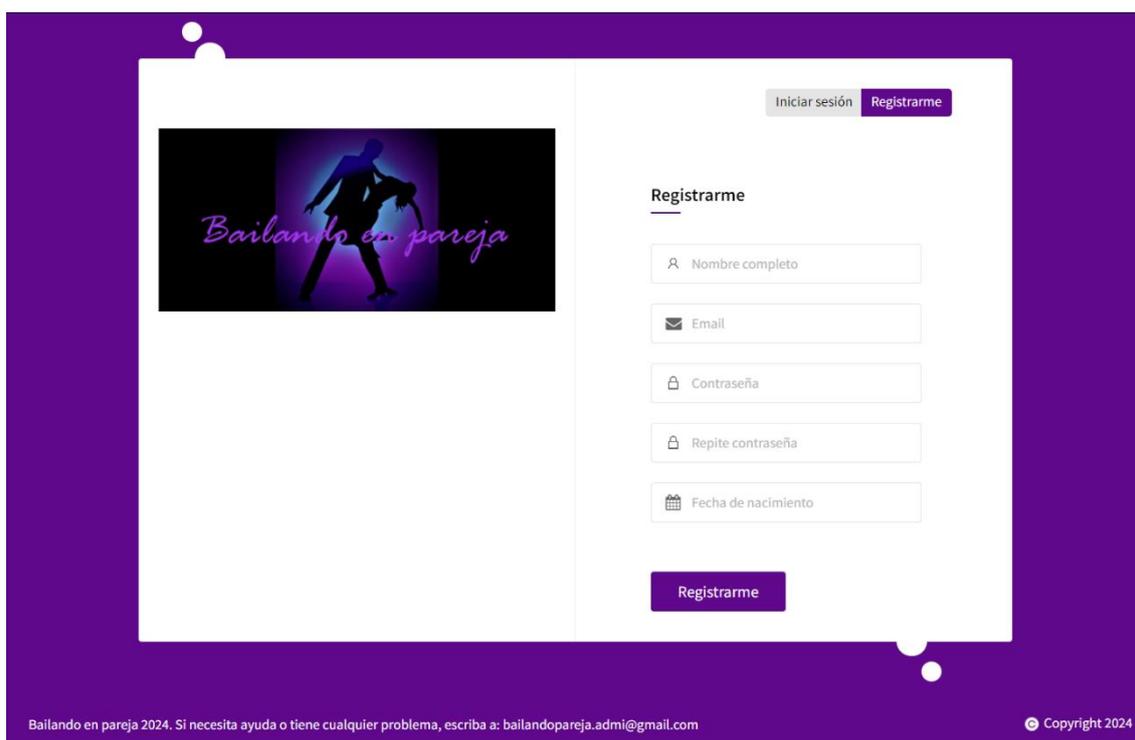


Figura 103. Pantalla registrarme.

8.2.2 Manual de usuario registrado

El usuario registrado, como bien indica su nombre, ya ha realizado el registro en la aplicación. Evidentemente, en caso de tener alguna incidencia, también puede hacer

uso del correo que aparece en el pie de página en el momento del inicio de sesión. Pero en caso de querer acceder a la aplicación, ha de cumplimentar correctamente los recuadros del inicio de sesión. No obstante, podría presentarse el texto ‘Usuario o contraseña no válido’ en un recuadro rojo, no permitiendo el acceso si no es correcto el email y/o la contraseña. Aunque podrían darse otras situaciones como es que el usuario haya sido desactivado por el administrador o que el mismo haya optado por eliminar el perfil.

De cualquier modo, para el inicio de sesión es necesaria la introducción de un email ya registrado, con su correspondiente contraseña. A pesar de ello, el inicio de sesión cuenta con la opción ‘¿Olvidó Contraseña?’ para que el usuario registrado que haya olvidado su contraseña, la reciba por email después de la introducción del email de registro.

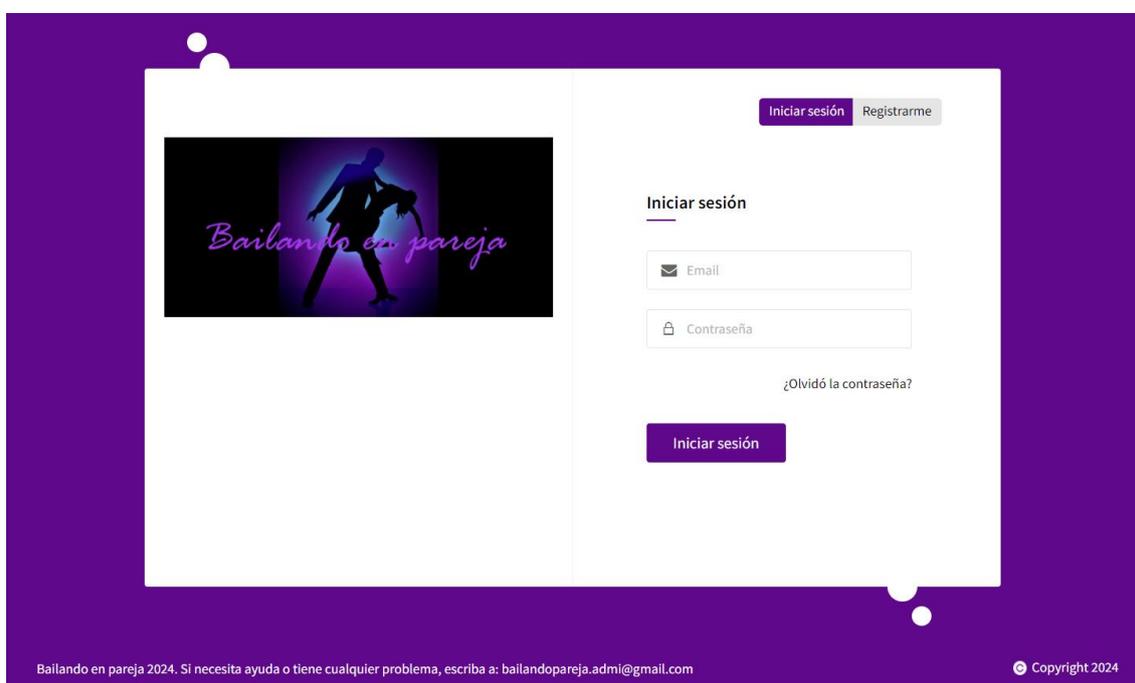


Figura 104. Pantalla iniciar sesión.

8.2.3 Manual de usuario general

El usuario general representa a todos los usuarios aficionado a los bailes de pareja que previamente han realizado el registro y han iniciado sesión en la aplicación y que les permite interactuar con la aplicación web y sus posibilidades. En caso de acceder, el usuario es dirigido a la primera opción del menú central de la cabecera ‘Home’.

El menú central de la cabecera cuenta con cuatro opciones, siendo de izquierda a derecha la primera ‘Home’, la segunda ‘Aficionados’, la tercera ‘Publicaciones’ y la cuarta ‘Mensajes’:

- ‘**Home**’ es donde son dirigidos todos los usuarios tras el inicio de sesión. Muestra en la zona de contenido de la página un texto el cometido de la aplicación, así

como un ruego de buen comportamiento y las posibles consecuencias de un mal comportamiento.



Figura 105. Pantalla home.

- ‘**Aficionados**’ muestra los aficionados activos en la aplicación en función de los filtros seleccionados del menú de búsqueda situado en la izquierda de la página. (El menú de filtros de aficionados puede quedar en la parte superior en caso de visualizarse en pantallas verticales como puedan ser las de dispositivo móvil). Dicho menú cuenta con cuatro filtros, las opciones ‘Buscar’ para iniciar la búsqueda y ‘Limpiar’ para limpiar el contenido de los filtros. Si no hubiera registrados y activos en la aplicación en base a los filtros de búsqueda, aparece el texto ‘No se han encontrado aficionados’. Los filtros de la búsqueda de aficionados son descritos de arriba a abajo: ‘Ciudad’, ‘Bailes’, ‘Géneros’ y ‘Rol’. Los cuales se explican a continuación:
 - ‘**Ciudad**’: para filtrar por la ciudad de los aficionados.
 - ‘**Bailes**’: para filtrar por el baile o bailes de los aficionados
 - ‘**Género**’: para filtrar por el género de los aficionados.
 - ‘**Rol**’: para filtrar por el rol de los aficionados.

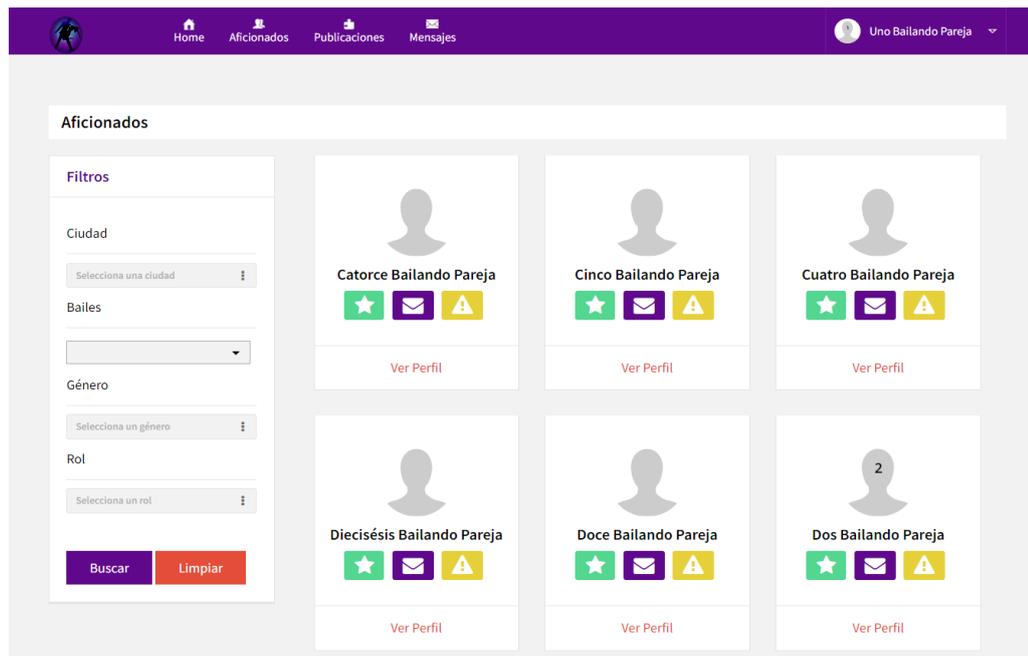


Figura 106. Pantalla aficionados.

- **‘Publicaciones’** es similar a la anterior opción explicada, solo que en este caso muestra las publicaciones de los usuarios activos quedando las más actuales arriba en función de los filtros seleccionados del menú de búsqueda situado también la izquierda de la página. (El menú de los filtros de publicaciones podría encontrarse en la zona de contenido superior si se visualiza desde una pantalla vertical). Hay que recalcar que solo muestra las publicaciones activas, es decir, aquellas que su fecha de finalización no ha vencido y además no han sido borradas por el propio usuario o desactivadas por un administrador. También cuenta con las opciones de ‘Buscar’ y ‘Limpiar’ funcionado exactamente, aunque los filtros son algo diferentes. Si por lo que fuera, no hubiera ninguna con los filtros seleccionados, entonces mostraría el mensaje de ‘No se han encontrado publicaciones’. Los filtros de la búsqueda de publicaciones descritos de arriba a abajo: ‘Ciudad’, ‘Bailes’, ‘Fechas’ y ‘Tipo’. Estos se detallan aquí debajo:
 - **‘Ciudad’**: filtra por la ciudad de la publicación.
 - **‘Bailes’**: filtra por el baile o bailes de la publicación.
 - **‘Fechas’**: filtra en función del intervalo comprendido entre dos fechas, una de inicio y otra de fin, incluyendo ambas. En caso de dar a buscar con una fecha final anterior a la inicial, salta una alerta con el texto ‘La fecha de fin debe ser mayor o igual a la fecha de inicio’ la cual hay que aceptar.
 - **‘Tipo’**: filtra en función del tipo de alusión a la que hace la publicación.

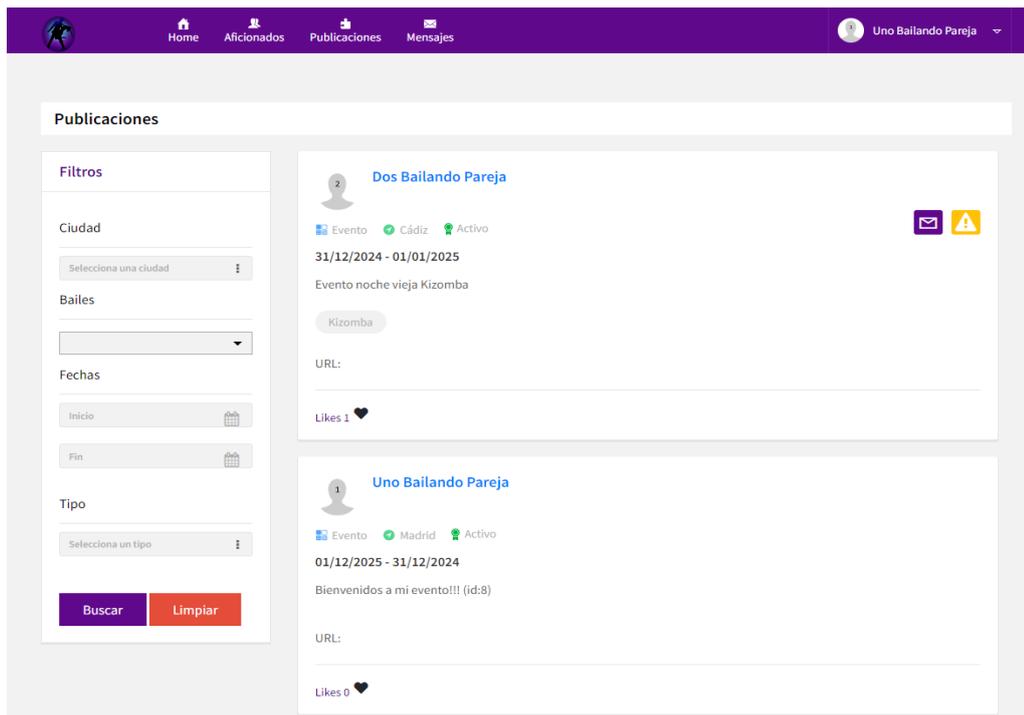


Figura 107. Pantalla publicaciones.

- **‘Mensajes’**, muestra las conversaciones mantenidas con los diferentes usuarios. Los usuarios se muestran a la izquierda organizados en función de la actualidad del último mensaje enviados por ellos, quedando arriba los usuarios cuyo mensaje enviado ha sido más reciente. En caso de que haya mensajes de un usuario sin leer, a la derecha del nombre del usuario aparece un círculo rojo y relleno un número en blanco indicando el número de mensajes sin leer. (Los usuarios con los que ha habido algún tipo de intercambio de mensajes, pueden estar en la parte superior y la conversación mostrada en la parte inferior, en caso de tratarse de pantallas verticales). Para ver las conversaciones es tan sencillo como seleccionar el aficionado en la zona de la izquierda del chat, entonces tras su selección se visualizará el usuario el usuario arriba y de debajo de este los mensajes ya enviados. Debajo de los mensajes enviados, se encuentra un cuadro de entrada de texto con el texto ‘Escribe un mensaje aquí’ y a la derecha de este cuadro se encuentra el botón de ‘Enviar’, útil para enviar mensajes al aficionado seleccionado una vez que se haya iniciado la comunicación.

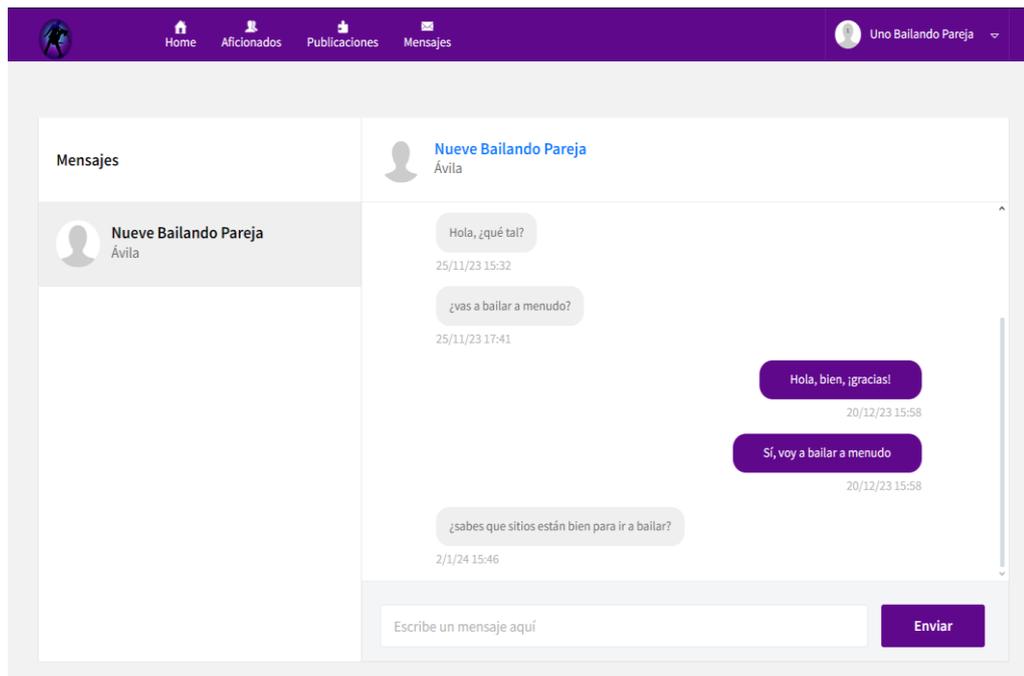


Figura 108. Pantalla mensajes.

En la cabecera casi a la derecha del todo, se encuentra la actual foto del perfil junto con el nombre y apellidos de la que sale un desplegable vertical con las siguientes tres opciones, enunciadas de abajo a arriba: ‘Salir’, ‘Configuración’ y ‘Mi perfil’. A continuación, se explican las tres opciones de selección del desplegable:



Figura 109. Pantalla con el menú desplegable.

- ‘**Salir**’, esta opción, tras su selección produce el cierre de la sesión del usuario.
- ‘**Configuración**’, la selección de esta opción da acceso a otra página con opciones sobre la configuración del perfil. Hay que recalcar que hay dos maneras de acceder a la configuración, desde el desplegable en el que aparece la foto y el nombre completo del aficionado de la sesión, o la opción del menú de la izquierda (pudiendo quedar en la parte superior en el caso de pantallas verticales, y aparecería tras seleccionar ‘Mi perfil’, que más adelante se va a explicar). Permite la configuración, donde la nueva página cuenta con un menú a la izquierda (también puede quedar arriba por la visualización vertical) con las siguientes

opciones presentadas de abajo a arriba ‘Eliminar perfil’, ‘Cambio de Contraseña’ y ‘Configuración’. Seguidamente se detallan estas opciones del menú de configuración:

- **‘Eliminar perfil’**: presenta en la zona de contenido la posibilidad de eliminar la cuenta, si se introduce la contraseña actual, los datos dejarán de ser visibles y no se podrá volver a acceder. Para ello es necesario introducir previamente la contraseña actual en cuadro en el que pone ‘Contraseña’. En caso de confusión con la contraseña el botón ‘Limpiar’ elimina los caracteres introducidos, en cambio el botón ‘Eliminar perfil’ procede a eliminar el perfil con el que se inició sesión en caso de ser la contraseña correcta. Detallar que, la cuenta realmente se está desactivando y solo un usuario que sea administrador podría activarla, volviendo a posibilitar el inicio de sesión.

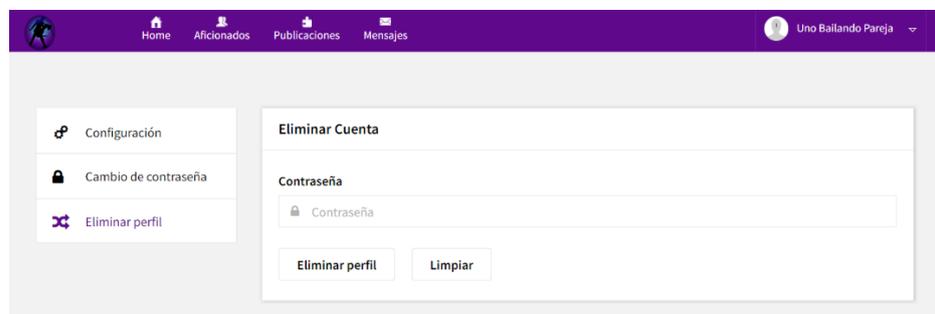


Figura 110. Pantalla eliminar contraseña.

- **‘Cambio de Contraseña’**: ofrece la posibilidad de modificar la contraseña del perfil. En el contenido se puede apreciar el título ‘Cambio De Contraseña’, abajo se encuentran con tres cuadros para la introducción de texto bajo ‘Antigua contraseña’, ‘Nueva contraseña’ y ‘Repite contraseña’, siendo necesario para el cambio la introducción correcta de las contraseñas correspondientes en cada cuadro de texto, de lo contrario saldrán alertas a aceptar o errores con caracteres en rojo sobre los cuadros de introducción de texto. La nueva contraseña tiene que ser coincidente, además de segura, de modo que tiene que contener al menos 8 caracteres, una mayúscula, una minúscula, un número y un carácter especial. Más abajo se encuentran los botones ‘Limpiar’ para eliminar los caracteres introducidos en caso de error en los tres cuadros de introducción de texto y ‘Cambiar contraseña’ para realizar el cambio. En caso de éxito aparecerá en verde bajo el título ‘Cambio de contraseña realizado de forma exitosa’, de lo contrario aparecerá en rojo ‘Cambio de contraseña no valido’ indicando en letras rojas el error sobre el cuadro de introducción de texto en el que se ha producido dicho error.

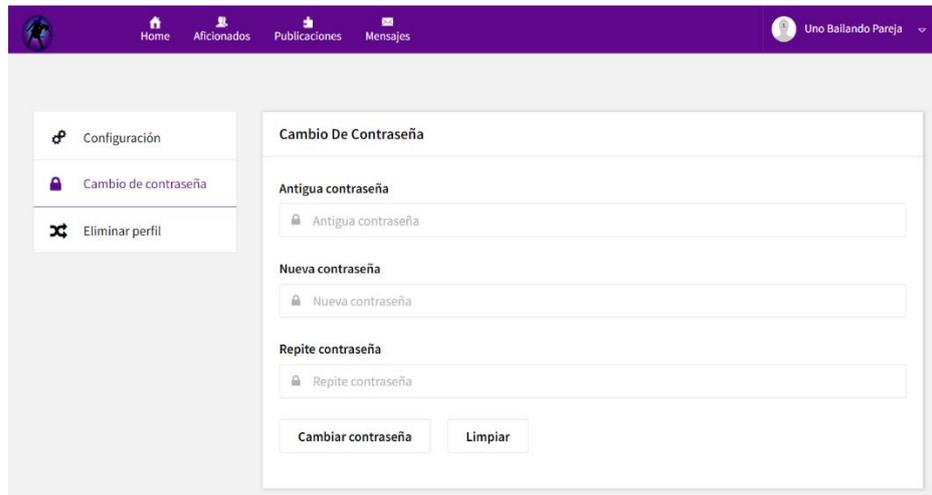


Figura 111. Pantalla cambio de contraseña.

- **‘Configuración’** presenta cuatro opciones en forma de interruptor. Si está desactivada la opción es poco visible, a diferencia de cuando está activada siendo visible el recorrido del interruptor de color morado junto con el interruptor como tal, representado por un punto blanco. Las opciones de configuración enunciadas de arriba abajo, que se detallan seguidamente:
 - **‘Notificar mensajes’**, para activar y desactivar la recepción de notificaciones al email de registro cada vez que se reciba un nuevo mensaje de cualquier aficionado.
 - **‘Notificar publicaciones’**, para activar y desactivar la recepción de notificaciones al email de registro cuando un usuario sea favorito, al crear o modificar una publicación.
 - **‘Ocultar edad’**, para activar y desactivar que no se vea la edad al ver el perfil.
 - **‘Ocultar email’**, para activar y desactivar que no se vea el email de registro y así nadie pueda ver el email, aunque los usuarios que sean administradores, sí que lo podrán seguir viendo.

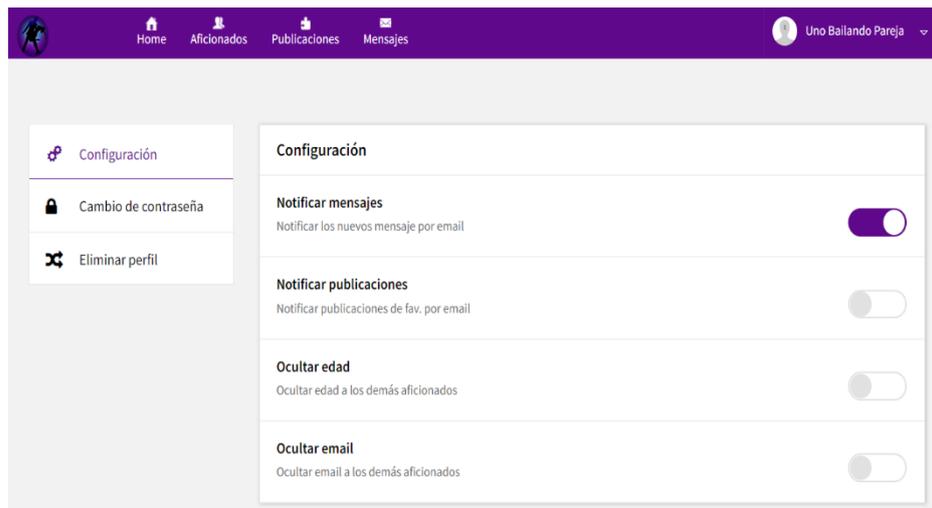


Figura 112. Pantalla configuración.

- ‘**Mi perfil**’ lleva al propio perfil del usuario que ha iniciado sesión. El propio perfil presenta la foto actual de relativo gran tamaño, viéndose parcialmente sobre la cabecera y sobre el menú de la izquierda (este menú, al igual que la foto pueden ubicarse en la parte superior en caso de pantallas verticales). Si se hace clic sobre la cámara se puede subir una foto almacenada en dispositivo con el que se ha iniciado sesión. Debajo de la foto ampliada del perfil están las opciones del menú indicadas de arriba a abajo: ‘Crear una publicación’ (podría no encontrarse, en caso de existir ya tres publicaciones vigentes), ‘Modificar Perfil’, ‘Configuración’ y ‘Eliminar Perfil’. Dichas opciones, se explican seguidamente:

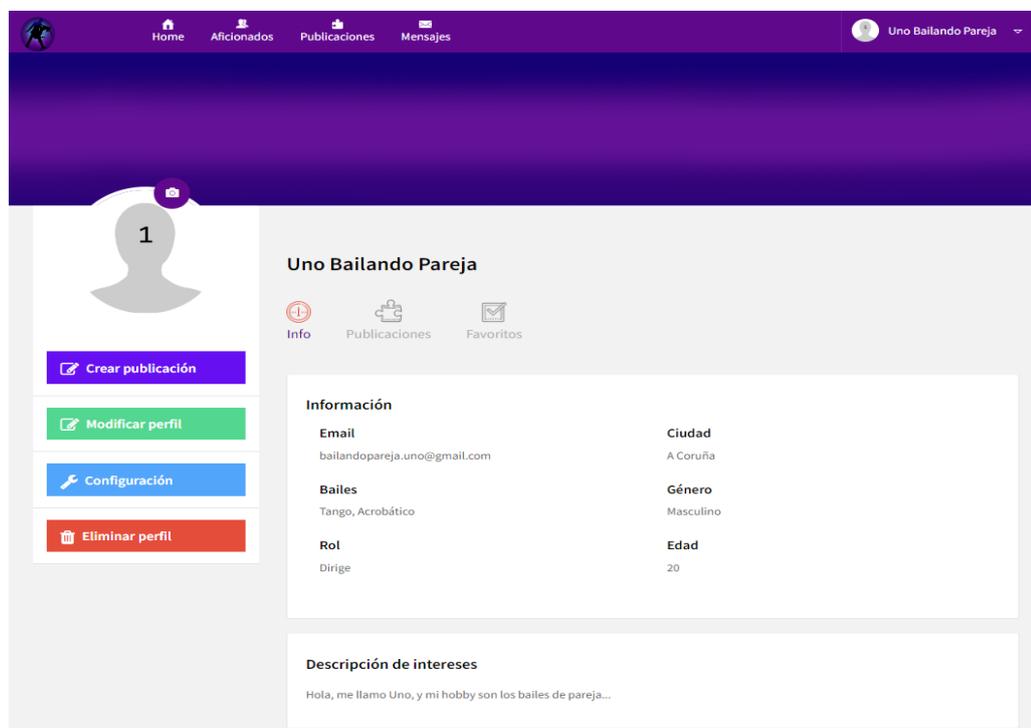


Figura 113. Pantalla mi perfil - info.

- **‘Crear Publicación’**, abre una ventana modal para la creación de nuevas publicaciones hasta un máximo de 3 publicaciones activas, evitando así el exceso de publicaciones por aficionado. La opción ‘Crear Publicación’ podría no encontrarse si ya hay 3 publicaciones activas, es decir, vigentes (que no han caducado). Las publicaciones caducadas, borradas por el creador y desactivadas por un administrador no cuentan como activas, siendo las caducadas activables cambiando la fecha siempre que no haya ya tres activas. Los campos de las publicaciones son: el tipo (quedada, clases, evento), la ciudad (ciudades de España ordenadas por orden alfabético), la fecha de inicio y de fin, el baile o bailes (salsa, bachata, kizomba, tango, swing, zouk, acrobático y otro), URL y una descripción sobre la publicación. Se pueden crear publicaciones con fechas que hagan al evento a estar vencido, eso sí, solo las verá propio creador. Para realizar la publicación en caso de que sea posible, teniendo los campos completados, es tan sencillo como clicar ‘Publicar’, de lo contrario ‘Cancelar’. En caso de activar alguna cambiando la fecha sale una alerta indicando ‘No se puede superar el límite de publicaciones vigentes permitidas’, la cual exige dar a aceptar.

Figura 114. Modal crear y modificar la publicación.

- **‘Modificar Perfil’**, abre otra ventana modal, la cual permite cambiar cierta información asociada el perfil como es el nombre y apellidos, la ciudad (ciudades de España están en orden alfabético), el género (masculino, femenino y otro), el rol (dirige, sigue y ambos), el baile (salsa,

bachata, kizomba, tango, swing, zouk, acrobático y otro), y descripción. En cambio, el email de registro y la edad no se pueden cambiar. Para que sea efectivo el cambio o los cambios hay que seleccionar ‘Guardar’, o ‘Cancelar’ para descartar los cambios. Es una opción que se recomienda usar de cara a las búsquedas y a tener el perfil correctamente, ya que por defecto los usuarios se crean con las siguientes asignaciones ‘Madrid’ (en ciudad), Otros (en bailes), Otro (en genero) y ‘Ambos’ (en rol).

Figura 115. Modal modificar perfil.

- **‘Configuración’**, lleva al mismo sitio, ya explicado anteriormente en el desplegable que sale de la foto de perfil junto con el nombre y apellidos tras la selección de la opción del medio ‘Configuración’. Del mismo modo se encuentran las opciones de configuración ya explicadas: ‘Eliminar perfil’, ‘Cambio de Contraseña’ y ‘Configuración’.
- **‘Eliminar Perfil’**, lleva a la misma opción ‘Eliminar perfil’ previamente explicada, con el desplegable tras la selección de la opción ‘Configuración’.

Permaneciendo en ‘Mi perfil’, además del menú de la izquierda (que puede quedarse arriba en formato vertical) coexiste un menú central en la zona superior del contenido, con las siguientes opciones de izquierda a derecha: ‘Info’, ‘Publicaciones’, y ‘Favoritos’, que aquí abajo se detallan:

- **'Info'**, muestra en la zona de contenido la propia información del usuario que ha iniciado sesión mostrando: el email, la ciudad, los bailes, el género, el rol y la edad, más la descripción de intereses. Esta información se puede modificar con la opción 'Modificar Perfil' situada en el menú de la izquierda (también podría encontrarse arriba), excepto el email y la edad vinculada a la edad de nacimiento quedando lo introducido en el registro. Recordar, que el email y la edad se puede ocultar al resto de aficionados haciendo uso de la configuración.
- **'Publicaciones'**, muestra en la zona de contenido las propias publicaciones quedando en la parte superior las más actuales. Aparecerán todas las publicaciones activas y vencidas, a no ser que se hayan borrado o desactivado por parte de los administradores. Aquí es el único sitio donde las publicaciones se pueden modificar o borrar sin exceder el límite de tres publicaciones activas. Aquí todas las publicaciones contarán en la parte superior derecha con dos botones pequeños y cuadrados, uno lápiz y otro con un cubo de basura.
 - El lápiz blanco sobre fondo verde, mediante una ventana modal emergente permite modificar la publicación de la misma manera, aunque no permitirá poner una fecha activa si ya tiene el usuario tres publicaciones activas mediante una alerta. Al presionar el botón con un lápiz blanco sobre fondo verde se pueden modificar o el botón del cubo sobre fondo rojo para eliminar la publicación.
 - El cubo sobre fondo rojo elimina la publicación, y creando una alerta que hay que aceptar, avisando que la publicación se va a borrar.

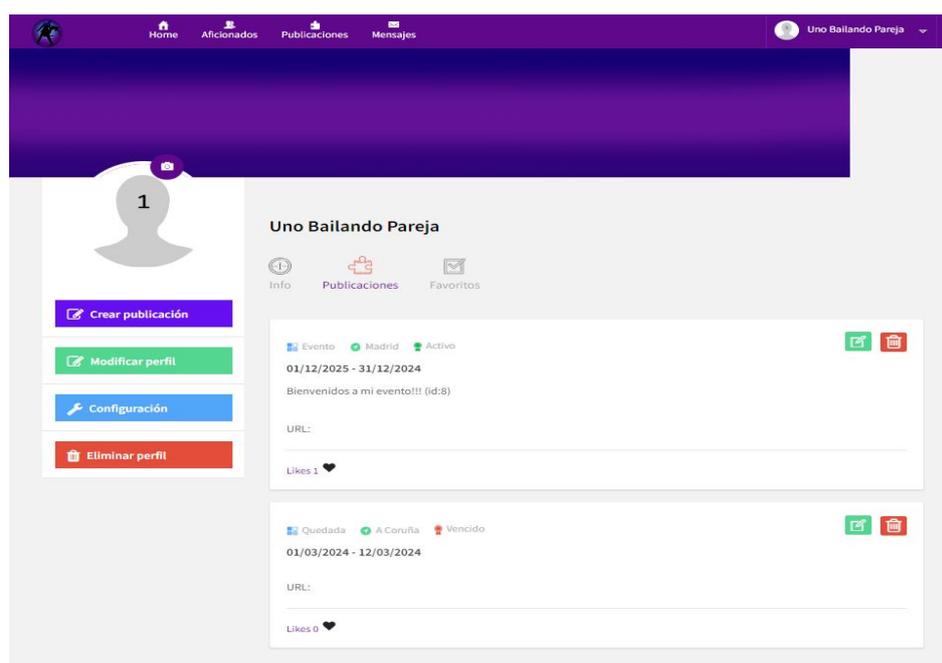


Figura 116. Pantalla mi perfil - publicaciones.

- **'Favoritos'**, muestra en la zona de contenido los aficionados añadidos a favoritos por el perfil que ha iniciado sesión. Al igual que se pueden añadir a favoritos, también se pueden quitar de ser favoritos, por lo que dejaran de aparecer en favoritos. Si hay aficionados asignados como favoritos, y la configuración 'Notificar publicaciones' está activada, en caso de un aficionado publique una publicación (cree o modifique), se recibirá un email al correo de registro.

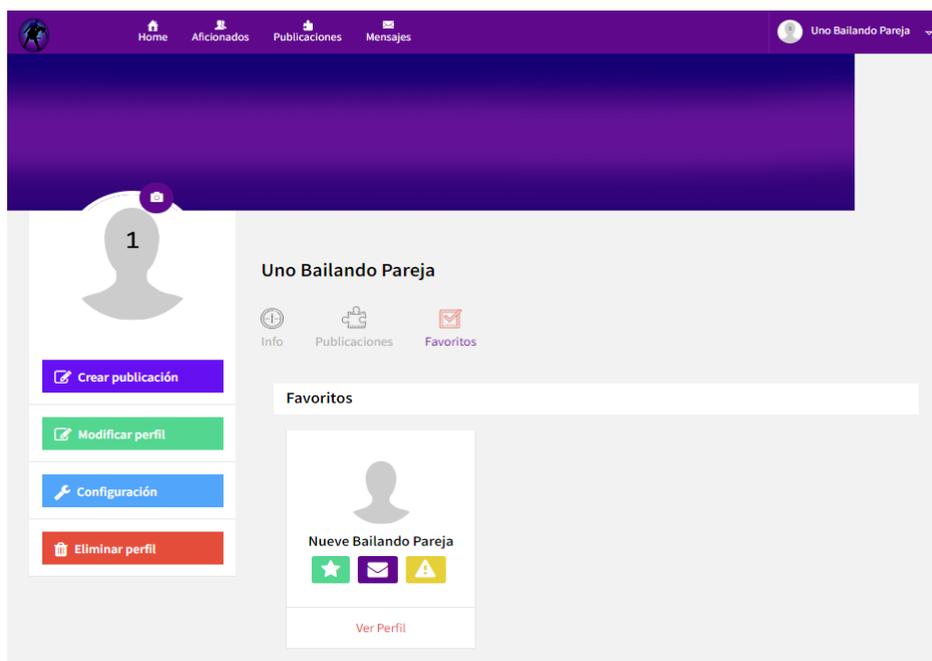


Figura 117. Pantalla mi perfil - favoritos.

Gracias a los perfiles de otros aficionados se ofrecen una serie de posibilidades:

- Estos perfiles pueden ser localizados inicialmente a través de:
 - El menú central de la cabecera tras la selección de 'Aficionados', puesto que aparecen aficionados activos en la aplicación. Asimismo, el menú de búsqueda con sus filtros puede ayudar a dar con usuarios más afines.
 - Haciendo uso del menú central de la cabecera con la selección de 'Publicaciones'. Es debido a que las publicaciones cuentan una imagen actual del perfil del autor, además del nombre y los apellidos.
 - Seleccionado 'Mensajes' del menú central de la cabecera, puesto que un aficionado puede escribir a otro, por lo que se recibe un mensaje, dando lugar a ser localizado por alguien. Localizándose así un aficionado que previamente ha comunicado algo.
- Se pueden ver otros perfiles en su totalidad:
 - Si el perfil aparece presentado como los presenta la opción 'Aficionados' (del menú central de la cabecera), se aprecia un recuadro con una foto del

perfil, más abajo tres botones la estrella de añadir a favoritos, el sobre de escribir, el símbolo de precaución para reportar. Y finalmente, más abajo la opción de 'Ver Perfil' dando acceso a la visualización completa del perfil. Recordando que email de registro y edad pueden encontrarse ocultos en función de la configuración.

- Si se está en la opción 'Publicaciones' (del menú central de la cabecera), se ven publicaciones por lo que aparecerán publicaciones de diferentes perfiles. Entonces, haciendo clic sobre la imagen o el nombre completo, se puede acceder al perfil y verle en su totalidad.
- Desde 'Favoritos' que se accede desde el desplegable donde aparece el propio perfil con el que se ha iniciado sesión, a través de la opción desplegable 'Mi perfil' situada a la derecha de la cabecera seleccionando sobre el recuadro de la imagen y el nombre completo. Aquí se ven los perfiles presentados como tras la selección de 'Aficionados', lo que sencillamente hay que clicar en 'Ver Perfil'.
- Otra manera posible forma de acceder al perfil en su totalidad puede ser, a través de la opción del menú central de la cabecera 'Mensajes', puesto que, seleccionado la conversación de un perfil, este queda arriba, se ve su conversación y se puede clicar sobre la foto del aficionado o el nombre completo del chat accediendo a ver su perfil completo.
- Estando ya en un perfil de otro aficionado, viéndole en su totalidad, uno encuentra:
 - La foto actual del perfil que está visualizando entre la cabecera y el menú de la izquierda.
 - Un menú a la izquierda (podría encontrarse arriba en pantallas verticales) con las siguientes opciones de arriba a abajo:
 - **Favorito**, una estrella en blanco acompañadas de la palabra 'Favorito' también en blanco.
 - **Enviar mensaje**, un sobre blanco seguido de las palabras blancas 'Enviar mensaje' sobre fondo morado, a través de una ventana modal.
 - **Reportar**, un triángulo blanco con exclamación junto a las palabras en blanco 'Reportar' sobre fondo amarillo, mediante una ventana modal que se abre.
 - Un menú superior en la zona de contenido con dos opciones de izquierda a derecha:
 - **'Info'**, para ver la información de ese usuario: email (puede estar oculto excepto para los administradores), ciudad, bailes, género, rol, edad (puede estar ocultarse) y la descripción de intereses.

- **‘Publicaciones’**, para ver las publicaciones activas de ese usuario. Es decir, las publicaciones no caducadas, y que además no hayan sido eliminadas o desactivadas por un administrador.
 - Tres puntitos, tienen las publicaciones dando la posibilidad de clicar sobre ellos para que aparezca la opción de reportar la propia publicación. Concretamente aparece un triángulo con exclamación acompañado de la palabra de ‘Reportar’ de la misma que cuando es un triángulo con exclamación blanco sobre un botón amarillo, pese a ser gris, abriendo un modal.
- Opciones de interacción con los perfiles de otros aficionados:
 - **‘Favorito’**: una estrella blanca sobre color verde. El clic provoca la añadidura del aficionado o el quitado del aficionado de la lista de favoritos (en función de si ya lo era o no), indicado mediante una alerta que hay que aceptar.
 - **‘Enviar mensaje’**: un sobre blanco sobre fondo morado. El clic abre una ventana modal que incluye un cuadro de introducción de texto para escribir un mensaje, además de los botones ‘Enviar’ para hacer efectivo el mensaje o de lo contrario ‘Cancelar’ para abandonar la idea.

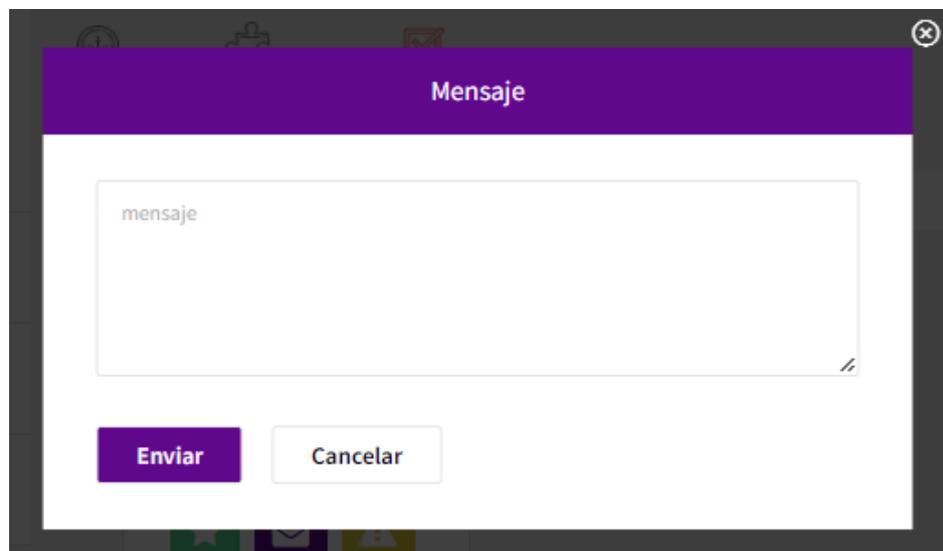


Figura 118. Modal enviar un mensaje.

- **‘Reportar’**: un triángulo blanco con exclamación sobre fondo amarillo. Al clicar sobre esta opción se abre una ventana modal en la que obligatoriamente se toma un motivo del desplegable, al cual se le puede añadir un comentario de texto a través del cuadro de texto. En caso de no querer seguir adelante, hay que hacer uso del botón ‘Cancelar’, pero en caso de seguir es tan sencillo como clicar sobre ‘Reportar’.

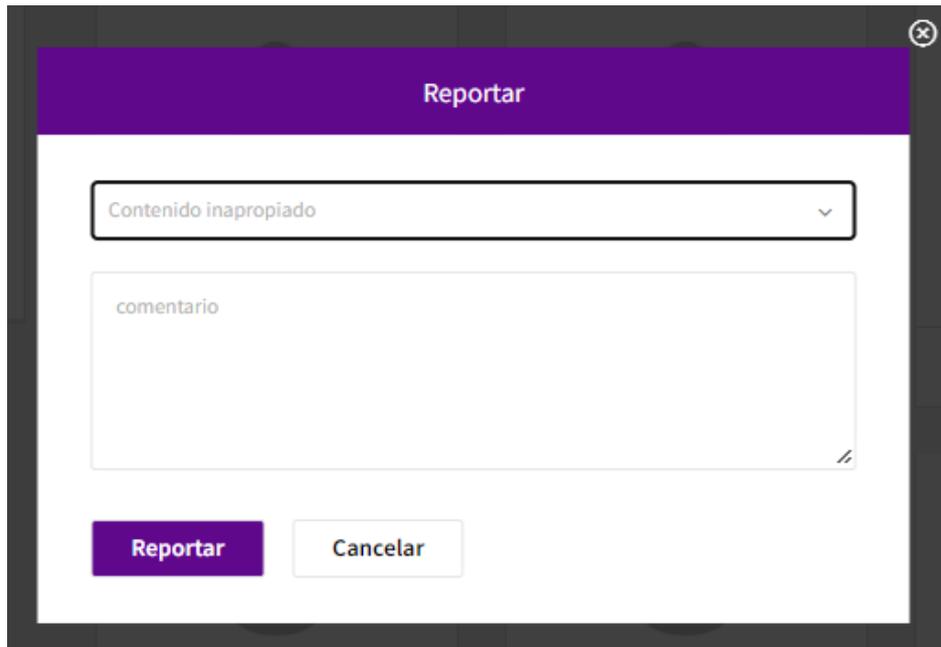


Figura 119. Modal reportar.

- Opciones de interacción en las publicaciones de otros aficionados:
 - **‘Enviar mensaje’**: un sobre blanco sobre fondo morado. Aparece en un pequeño cuadrado y permite enviar mensajes de la misma manera que lo permite un perfil del usuario, enviándose al autor de la publicación.
 - **‘Reportar’**: un triángulo blanco con exclamación sobre fondo amarillo. Es también un pequeño cuadrado, el cual permite realizar un reporte de esa misma publicación. Se abre el mismo modal y funciona de la misma manera que con un aficionado, solo que se asocia a la publicación.
 - **‘Likes’**: Las publicaciones pueden recibir me gusta por parte de todos los usuarios, incluso del propio creador de la publicación. Tras un clic sobre ‘likes’, el contador que se encuentra después se incrementa un uno, y si por lo que se un mismo usuario vuelve a hacer clic, el contador vuelve al estado original retirando el me gusta dado con el clic inicial sobre la publicación.

8.2.4 Manual de usuario administrador

Los usuarios administradores son usuarios con las mismas funcionalidades que los usuarios generales salvo por los filtros de búsqueda para perfiles y publicaciones. No obstante, hay funcionalidades extra, puesto que pueden visualizar los reportes, además de desactivar y activar tanto publicaciones como usuarios. Por lo que la búsqueda cuenta con filtros para localizar en función de los reportes y la activación de las publicaciones y los perfiles.

Los reportes pueden recibirlos tanto los usuarios generales como los usuarios administradores y también sus publicaciones. Los administradores en caso de querer reportar solo podrán reportar desde el propio perfil en detalle para reportar el perfil o sus publicaciones, dejando así alguna referencia, puesto que en vez salir la ventana modal con los reportes, sale el modal para reportar.

Solo los usuarios administradores tienen la capacidad de ver los reportes, no accediendo al perfil con detalles, solo según salen tras la búsqueda los perfiles y las publicaciones. Para ello, se hace clic sobre el triángulo blanco sobre fondo amarillo (sin estar dentro del perfil detallado), abriendo así una ventana modal, ya sea de un perfil o de una publicación, donde aparecen tres columnas 'Fecha', 'Motivo' y 'Comentario' mostrando en una línea los detalles de cada reporte, independientemente de si están dirigidos a un usuario o a una publicación.



The image shows a modal window with a purple header containing the text 'Reportes de Nueve Bailando Pareja'. Below the header is a table with three columns: 'Fecha', 'Motivo', and 'Comentario'. The table contains one row of data.

Fecha	Motivo	Comentario
2023-11-25	Perfil falso	Probando reporte de perfil para Nueve de Uno

Figura 120. Modal ver reportes.

Las inactivaciones y activaciones solo las pueden llevar a cabo los administradores pudiendo inactivar y activar tanto a usuarios generales, como a usuarios administradores, además de las publicaciones de cualquier usuario.

Cuando una publicación se desactiva por parte del administrador o se elimina por parte de creador, se pone el estado de la publicación a cero, dejando de ser visible para cualquier usuario general, a menos que un usuario administrador la active. Hay que

puntualizar, que los usuarios administradores si son capaces de ver las publicaciones caducadas.

Cuando un perfil se desactiva por parte del administrador o el propio usuario elimina su perfil, se pone el estado de usuario a cero. Entonces sus publicaciones dejarán accesibles desde la búsqueda, a menos que activen el perfil. Las conversaciones mantenidas en el chat dejaran de ser visible para todos los usuarios generales excepto los usuarios administradores, en caso de haber mantenido conversación.

A continuación, se detallan los filtros de los usuarios administradores por diferentes por su acceso según las opciones del menú central de la cabecera que son ‘Aficionados’ y ‘Publicaciones’:

- ‘**Aficionados**’, para el administrados aporta exclusivamente los siguientes filtros:
 - ‘**Buscar por:**’ bajo este título se encuentra el texto en grisáceo ‘email o nombre’ un cuadro de introducción de texto que no distingue entre mayúsculas y minúsculas, para la introducción del email o la una parte del nombre completo para filtrar en la búsqueda y obtenerlos aficionados coincidentes.
 - ‘**Inactivo**’, tiene un cuadradito que se puede clicar, apareciendo así una ‘v’ blanca sobre fondo morado en dicho cuadradito para activar el filtro y que salgan los usuarios inactivados.
 - ‘**Motivo del reporte**’ cuenta con un desplegable con diferentes opciones, para filtrar los aficionados en función de la motivación del reporte.

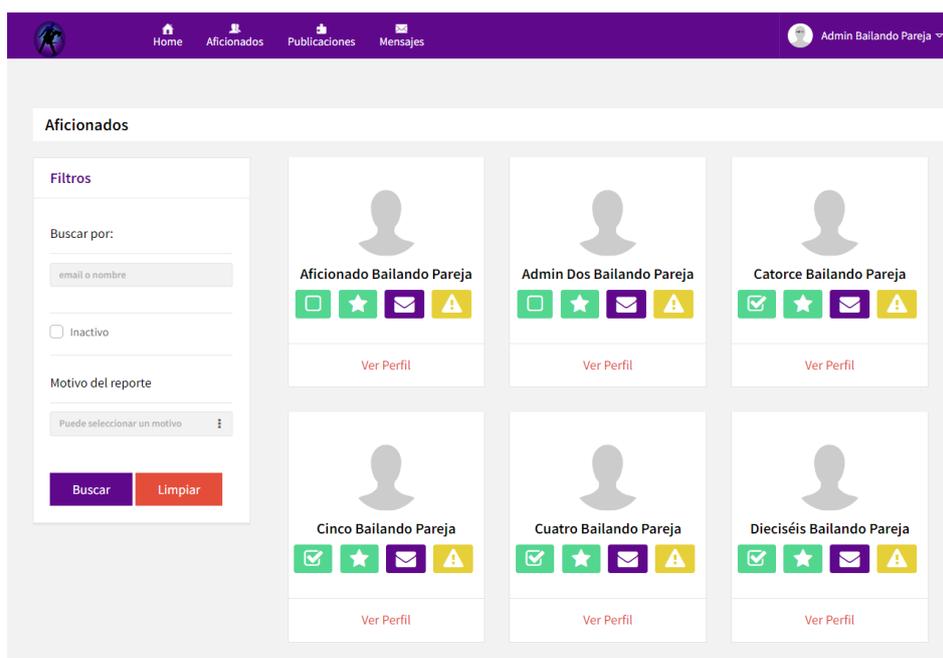


Figura 121. Pantalla buscar perfiles reportados.

- **‘Publicaciones’**, únicamente para el administrador cuenta con los siguientes filtros:
 - **‘Buscar por:’** cuenta debajo con un cuadro de entrada de texto que pese a tener el texto grisáceo ‘email, nomb. creador o descrip. evento’ permite la entrada de texto para filtrar sin distinguir entre mayúsculas y minúsculas, a través de la introducción del email o parte del nombre del creador o incluso parte o alguna palabrea contenida en la descripción.
 - **‘Inactivo’**, filtro con un pequeño cuadro para clicar, de manera que si está activo aparece una ‘v’ blanca sobre fondo morado, permitiendo así filtrar por publicaciones inactivas.
 - **‘Motivo del reporte’** desplegable con múltiples motivos por los que filtrar para encontrar las publicaciones que cuenten con el motivo seleccionado.

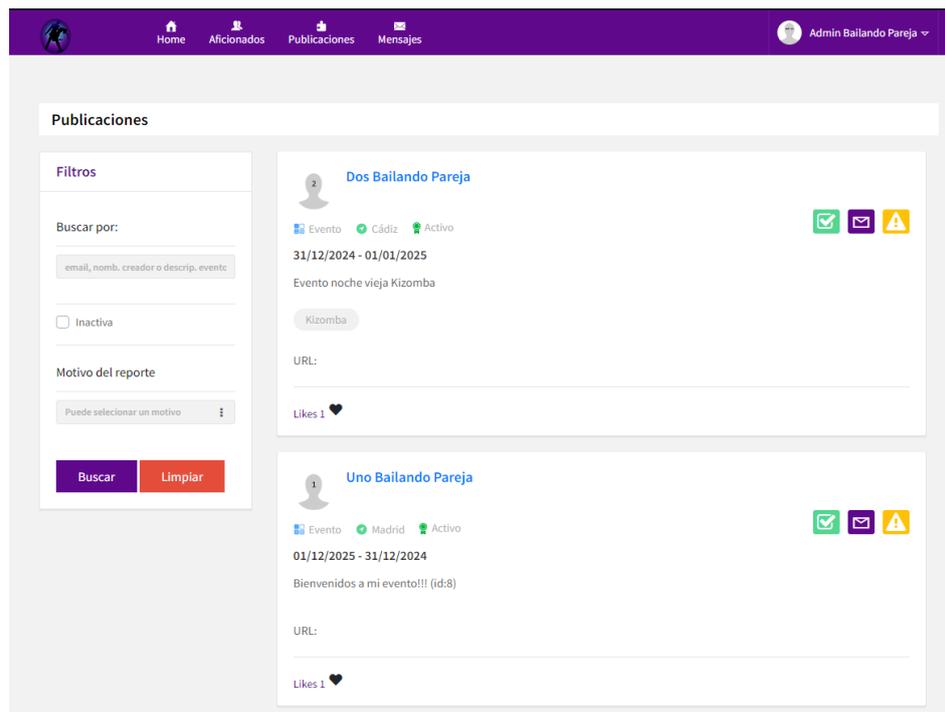


Figura 122. Pantalla buscar publicaciones reportadas.

Hay que tener bien presente que para que aficionado general cambie a ser administrador, es necesario que se le añada en la base de datos. La sentencia para que un usuario general pase a ser usuario administrador es: `INSERT INTO `bailandopareja`.`usuario_rol` (`usuario`, `rol_app`) VALUES ('emaildeejemplo@mail.xxx', '2');`

9. Conclusiones

Las conclusiones del desarrollo se van a exponer abordando primeramente las experiencias que ha supuesto el desarrollo de este trabajo de fin de grado, para posteriormente continuar con los conocimientos adquiridos y las líneas de trabajo detectadas.

La idea de realizar la aplicación surgió de las necesidades experimentadas y escuchadas durante el desarrollo de la afición a los bailes de pareja, a lo que en su momento se sumó el deseo de experimentar cómo sería el desarrollo de una red social con funcionalidades básicas, haciendo uso de las tecnologías Thymeleaf y Spring Boot. Al final, la complejidad no solo ha estado en el uso de dichas tecnologías, algo ya esperado. Sino también en el diseño de las interfaces a partir de la plantilla siendo algo no tan esperado. Sobre todo, de cara a organizar la navegación por las interfaces gráficas dando acceso a las funcionalidades de la red social.

A pesar de la satisfacción de haber aprendido mucho, al diseñar y desarrollar una aplicación, además de por haber hecho uso de nuevas tecnologías, haciendo frente a numerosas y diferentes complicaciones. Hay que destacar, que ha supuesto un mayor esfuerzo por la compaginación con un máster, suponiendo un esfuerzo añadido, el cual ha llevado acumular desgaste y agotamiento. De manera que las partes tediosas sobre como puedan ser las de la documentación hayan resultado aún más tediosas de lo que deberían haber resultado. Lo que ha llevado a que este trabajo de fin de grado haya sido toda una prueba de resistencia y tenacidad, haciéndole a uno luchar mucho por no tirar la toalla y abandonar.

Una vez expresada la experiencia personal, se va a tratar seguidamente tanto lo aprendido como las posibles líneas de trabajo detectadas.

9.1. Conocimientos adquiridos

Los conocimientos adquiridos para el desarrollo de la aplicación *Bailando en pareja* no han sido pocos, además de haber supuesto un reto.

Antes de todo, ha sido necesaria una previa documentación sobre las tecnologías Thymeleaf más Spring Boot encontradas principalmente en los tutoriales online ofrecidos por Baeldung [16] y Spring by VMware Tanzu [17], así como los tutoriales de Javapoint [18] y de Arquitectura Java [19]. Para posteriormente, también tener que comenzar a recuperar conocimientos de bases de datos y poder empezar con las base de datos.

El proyecto servicios no ha sido sencillo, pese a la ayuda de la creación de las entidades gracias al uso de NetBeans, en varias ocasiones ha sido necesaria la modificación de las etiquetas como puedan ser las etiquetas correspondientes a las relaciones entre entidades. Además, también se ha requerido tener muy claro el uso de los diferentes tipos de peticiones y sus correspondientes anotaciones a nivel de parámetro,

para crear métodos diferentes a los genéricos. Sin olvidar que las JPA NamedQueries requieren de una organización distinta a las SQL Query.

Se ha ahondado en el conocimiento de la arquitectura con el uso de DTOs, es decir, objetos de transferencia de datos. A parte de evitar la generación y el parseo de las cadenas de JSON. Estos ayudan a separar las capas, permitiendo enviar y recibir con cada uno de ellos, sin contener ningún tipo de lógica. Los DTOs son objetos serializables que contienen todos los datos de una llamada, permitiendo agrupar varias llamadas remotas en una sola llamada.

Hay que destacar que la integración de Thymeleaf en la generación de un proyecto Spring Boot con Spring Initializr es muy sencilla debido a que se indica como una dependencia más. Integrándose Thymeleaf dentro del patrón MVC, permitiendo la redirección de las plantillas mediante las clases controladoras, el uso de Spring Expression Language, la integración de los formularios con: los beans, los result bindings y los manejadores de errores, e incluso la internacionalización mediante la interfaz MessageSource.

Las plantillas han sido modificadas para adaptarlas a las necesidades de la aplicación. Lo que ha implicado la adaptación de archivos HTML y CSS, e incluso la adaptación y creación de nuevos métodos de JavaScript.

En definitiva, el desarrollo de la aplicación ha supuesto no solo un reto, sino la adquisición y puesta en práctica de un montón de conocimientos.

9.2. Líneas de trabajo para el futuro

Las líneas de trabajo para el futuro podrían ser varias, así que más adelante se van a exponer algunas propuestas. Aunque, es relevante recordar, que la aplicación *Bailando en pareja* ha surgido como una idea para la realización de un trabajo de fin de grado. Por lo que se ha pretendido en todo momento cumplir con las expectativas de dicho cometido.

No obstante, como todo trabajo de fin de grado, debe ajustarse a unos tiempos de entrega, lo cual, ha limitado algunas características iniciales. Pese a ello, de cara al futuro, lo suyo sería reforzar las ventajas diferenciadoras del software respecto a la competencia, llevar a cabo ciertas mejoras que en breve se van a reflejar, así como buscar la manera de que la aplicación pudiera ser viable económicamente.

Es por ello, por lo que se van a proponer una serie de mejoras de cara al futuro para la aplicación *Bailando en pareja*:

- Un recaptcha sería ideal, sobre todo en el registro asegurando que la interacción con la aplicación web sea humana. Evitando así la interacción y generación de perfiles por softwares automatizados, a la vez que ofrecería una mayor seguridad ante posibles ataques y malos usos de dichos softwares.

- La posibilidad de añadir perfiles de usuarios como bloqueados, evitando así interacciones indeseadas. No siendo necesaria la inactivación de un perfil por parte de los usuarios administradores para evitar interacciones indeseadas por parte de ciertos usuarios molestos, además de comportamientos poco respetuosos. La base de datos está preparada, e incluso algunas clases, pero se ha descartado la funcionalidad finalmente por falta de tiempo.
- Implementación de un buzón de sugerencias para guiar el futuro desarrollo de la aplicación y así poder evolucionar junto con las necesidades de los usuarios aficionados al baile.
- Búsqueda de financiación para que el proyecto pueda ser de alguna manera rentable y por lo tanto viable económicamente. Una posible financiación mediante anuncios y/o recepción de ayudas para la promoción de la cultura y la salud podrían ser la clave.
- Poner la aplicación a disposición de los aficionados desplegándola para que puedan hacer uso de ella.

En definitiva, estas mejoras podrían ser las líneas de trabajo futuras para mejorar la aplicación, puesto que no han sido posibles por limitaciones temporales.

Referencias

- [1] **Un modelo de estimación de proyectos de software.** Adriana Gómez, María del C.López y Silvina Migani, Alejandra Otazú. Disponible en: <https://blogadmi1.wordpress.com/wp-content/uploads/2010/11/cocom011full.pdf> [Descargado el 28/03/2023].
- [2] **Function point languages table.** Quantitative Software Management. Disponible en: <https://www.qsm.com/resources/function-point-languages-table> [Última consulta el 02/04/2024].
- [3] **Workwise - The Freelancer And Social Networking HTML Template By Gambolthemes.** Plantilla HTML. Disponible en: https://themeforest.net/item/workwise-the-freelancer-and-social-networking-html-template/21647879?s_rank=17 [Descargado el 31/03/2023].
- [4] **Spring Initializr.** Disponible en: <https://start.spring.io> [Último acceso el 30/03/2023].
- [5] **Java JDK 17.** Open JDK. Disponible en: <https://openjdk.org/projects/jdk/17/> [Última consulta el 02/04/2024].
- [6] **Spring Boot.** Disponible en: <https://spring.io/projects/spring-boot> [Última consulta el 02/04/2024].
- [7] **Thymeleaf.** Disponible en: <http://www.thymeleaf.org/> [Última consulta el 02/04/2024].
- [8] **Bootstrap.** The world's most popular mobile-first and responsive front-end framework. Disponible en: <http://getbootstrap.com/> [Última consulta el 03/04/2024].
- [9] **JavaScript, HTML 5 y CSS 3.** W3Schools Online Web Tutorials. Disponible en: <https://www.w3schools.com/> [Última consulta el 03/04/2024].
- [10] **Maven.** Apache Maven Project. Disponible en: <https://maven.apache.org/> [Última consulta el 04/04/2024].
- [11] **REST.** Disponible en: <https://spring.io/guides/tutorials/rest> [Última consulta el 04/04/2024].
- [12] **JSON.** Introducción a JSON. Disponible en: <https://json.org/json-es.html> [Última consulta el 04/04/2024].
- [13] **XML.** XML Tutorial. Disponible en: <https://www.w3schools.com/xml/> [Última consulta el 05/04/2024].
- [14] **MySQL.** MySQL 8.0 Reference manual. Disponible en: <https://dev.mysql.com/doc/refman/8.0/en/innodb-storage-engine.html> [Última consulta el 05/04/2024].

- [15] **Tomcat.** Apache Tomcat. Disponible en: <https://tomcat.apache.org/> [Última consulta el 05/04/2024].
- [16] **Baeldung.** Sitio web que con tutoriales sobre Java y Spring Boot. Disponible en: <https://www.baeldung.com/> [Última consulta el 14/04/2024].
- [17] **Spring by VMware Tanzu.** VMware ofrece formación y certificación para impulsar su progreso. Disponible en: <https://spring.io/> [Última consulta el 02/04/2024].
- [18] **Javapoint.** Plataforma de aprendizaje gratuita para un futuro mejor. Disponible en: <https://www.javatpoint.com/> [Última consulta el 15/03/2024].
- [19] **Arquitectura Java.** Blog sobre Java EE. Disponible en: <https://www.arquitecturajava.com/> [Última consulta el 14/03/2023].
- [·] **Stack Overflow.** Where developers learn, share, & build careers. Disponible en: <https://stackoverflow.com> [Última consulta el 30/03/2024].
- [·] **GitHub.** Build software better, together. Disponible en: <https://github.com> [Última consulta el 16/03/2024].

Anexo I: Glosario

- **ApiREST:** se conoce también como Api RESTful, es una interfaz de programación para aplicaciones con una transferencia de estado representacional.
- **BBDD:** es la abreviatura de bases de datos. Las bases de datos es información organizada que permite su almacenamiento y obtención de datos de manera rápida y eficiente.
- **Framework:** es un esquema de trabajo para el desarrollo de software de manera más eficiente por parte de los programadores.
- **HTTP:** es un protocolo de transferencia de hiper textos usado en para la comunicación entre cliente y servidor.
- **JVM:** es la máquina virtual de Java, interpreta el código Java y del código máquina que los ordenadores interpretan, permitiendo la ejecución independientemente de la plataforma.
- **Login:** es la identificación del usuario con sus credenciales, para acceder al sistema iniciando sesión.
- **Logout:** es el cierre de sesión del usuario, para que otras personas no accedan al sistema.
- **MVC:** es un patrón de diseño software, en el que se separan las responsabilidades: la interfaces de usuario, la lógica de control y los datos.
- **POM:** es el archivo de modelo de objetos de proyecto, que contiene información esencial para el proyecto.
- **Spring Initializr:** herramienta para crear y configurar proyectos en Spring Boot de manera muy sencilla creando así la estructura básica.
- **URL:** es el localizador uniforme de recursos, una dirección para la identificación y localización de recursos en Internet.

Anexo II: Casos de uso

En el presente anexo se adjuntan el resto de los casos de uso que no han sido incluidos en el apartado “4.2.2 Especificaciones de los casos de uso”. Los casos de uso se presentan organizados por actores y siguen el orden de la tabla (ver tabla 14) incluida en el apartado “4.2.1 Listado de casos de uso”.

Usuario general

Id y caso de uso	UG1 - Ver perfil propio
Descripción	Un usuario accede a ver su propio perfil.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación.
Secuencia	1. El usuario solicita ver su propio perfil. 2. La aplicación consulta los datos de la BBDD sobre el propio usuario.
Postcondición	Pos-1: La aplicación muestra el perfil del usuario que ha iniciado sesión.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EDU.01, RI.EDU.02, RI.EDU.03, RI.EDU.04, RI.EDU.05, RI.EDU.06, RI.EDU.07, RI.RDUB.01, RI.RDUB.02, RI.EB.01, RI.EDU.02, RI.ECI.01, RI.ECI.02 Dep. - Reglas de negocio: RN.01, RN.05
Observaciones	El email y la edad es siempre visible en el perfil propio, independientemente de la configuración.

Tabla 107. Caso de uso UG1.

Id y caso de uso	UG2 - Consultar configuración propia
Descripción	Un usuario accede a ver su propia configuración.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación.
Secuencia	1. El usuario solicita ver su propia configuración. 2. La aplicación consulta los datos de la BBDD sobre su propia configuración.
Postcondición	Pos-1: La aplicación muestra la configuración del usuario que ha iniciado sesión.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EC.01, RI.EC.02, RI.EC.03, RI.RT3.01, RI.RT3.02 Dep. - Reglas de negocio: RN.01, RN.08
Observaciones	

Tabla 108. Caso de uso UG2.

Id y caso de uso	UG3 - Gestionar las notificaciones por email de los mensajes recibidos
Descripción	Un usuario gestiona si recibe o no un email por cada mensaje recibido por el chat.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que encontrarse consultando su propia configuración.
Secuencia	1. El usuario solicita activar o desactivar el notificar los mensajes nuevos por email. 2. La aplicación actualiza la BBDD cambiando su propia configuración.
Postcondición	Pos-1: La aplicación muestra la nueva configuración sobre notificaciones de mensajes.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EC.01, RI.EC.02, RI.EC.03, RI.RT3.01, RI.RT3.02 Dep. - Reglas de negocio: RN.01, RN.08
Observaciones	La activación y desactivación se lleva a cabo y se representa mediante un switch.

Tabla 109. Caso de uso UG3.

Id y caso de uso	UG4 - Gestionar las notificaciones por email de las publicaciones de los favoritos
Descripción	Un usuario gestiona si recibe o no un email por cada publicación de un perfil añadido a favorito.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que encontrarse consultando su propia configuración.
Secuencia	1. El usuario activa o desactiva el notificar las publicaciones de favoritos. 2. La aplicación actualiza la BBDD cambiando su propia configuración.
Postcondición	Pos-1: La aplicación muestra la nueva configuración sobre notificaciones de publicaciones.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EC.01, RI.EC.02, RI.EC.03, RI.RT3.01, RI.RT3.02 Dep. - Reglas de negocio: RN.01, RN.08
Observaciones	Esta notificación solo funciona con usuarios añadidos a favoritos, en caso de estar activada. La activación y desactivación se lleva a cabo y se representa mediante un switch.

Tabla 110. Caso de uso UG4.

Id y caso de uso	UG5 - Gestionar la visibilidad de la edad
Descripción	Un usuario gestiona si esconde o no la edad del perfil para el resto de los usuarios.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que encontrarse consultando su propia configuración.
Secuencia	1. El usuario activa o desactiva el esconder la edad al resto de usuarios. 2. La aplicación actualiza la BBDD cambiando su propia configuración.
Postcondición	Pos-1: La aplicación muestra la nueva configuración sobre esconder la edad.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EC.01, RI.EC.02, RI.EC.03, RI.RT3.01, RI.RT3.02 Dep. - Reglas de negocio: RN.01, RN.08
Observaciones	La activación y desactivación se lleva a cabo y se representa mediante un switch.

Tabla 111. Caso de uso UG5.

Id y caso de uso	UG6 - Gestionar la visibilidad del email
Descripción	Un usuario gestiona si esconde o no el email del perfil para el resto de los usuarios.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que encontrarse consultando su propia configuración.
Secuencia	1. El usuario activa o desactiva el esconder el email al resto de usuarios. 2. La aplicación actualiza la BBDD cambiando su propia configuración.
Postcondición	Pos-1: La aplicación muestra la nueva configuración sobre esconder el email.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EC.01, RI.EC.02, RI.EC.03, RI.RT3.01, RI.RT3.02 Dep. - Reglas de negocio: RN.01, RN.08
Observaciones	La activación y desactivación se lleva a cabo y se representa mediante un switch. Para los usuarios administradores el email siempre es visible en el perfil, independientemente de la configuración.

Tabla 112. Caso de uso UG6.

Id y caso de uso	UG7 - Editar perfil
Descripción	Un usuario accede a modificar su propio perfil.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que encontrarse consultando su propio perfil.
Secuencia	1. El usuario solicita modificar su perfil habiendo generado una consulta. 2. El usuario puede modificar los datos que sí lo permiten. 2. La aplicación actualiza los datos del perfil en la BBDD.
Postcondición	Pos-1: La aplicación muestra el perfil con los datos actualizados.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EDU.01, RI.EDU.02, RI.EDU.03, RI.EDU.04, RI.EDU.05, RI.EDU.06, RI.EDU.07, RI.RDUB.01, RI.RDUB.02, RI.EB.01, RI.EDU.02, RI.ECI.01, RIECI.02 Dep. - Reglas de negocio: RN.01, RN.05, RN.10
Observaciones	El email y la edad de registro no son modificables.

Tabla 113. Caso de uso UG7.

Id y caso de uso	UG8 - Eliminar perfil
Descripción	Un usuario solicita eliminar su propio perfil.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que encontrarse consultando su propio perfil o consultando su configuración.
Secuencia	1. El usuario solicita eliminar su perfil. 2. El usuario introduce su contraseña actual, enfrentándose a una comprobación (Exc-1). 3. La aplicación actualiza el estado del usuario en de la BBDD.
Postcondición	Pos-1: La aplicación muestra la página de inicio de sesión y registro no permitiendo el inicio de sesión y el registro con el email del usuario eliminado.
Excepciones	Exc-1: Si la contraseña introducida y cifrada, no es coincidente con la almacenada cifrada, la aplicación muestra error y se vuelve al paso 2.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EU.05, RI.RT1.01, RI.RT1.02, RI.ERA.01, RI.ERA.02 Dep. - Reglas de negocio: RN.01, RN.15, RN.17
Observaciones	Posteriormente la activación se puede solicitar a un usuario administrador a través de email encontrado en el pie de página del registro o inicio de sesión.

Tabla 114. Caso de uso UG8.

Id y caso de uso	UG9 - Cambiar contraseña
Descripción	Un usuario accede al cambio de su contraseña.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que encontrarse consultando su propia configuración.
Secuencia	1. El usuario solicita cambiar su contraseña. 2. La aplicación presenta la página insertar la contraseña actual y dos veces la nueva. 3. La aplicación verifica y valida las contraseñas (Exc-1, Exc-2 y Exc-3). 4. La aplicación actualiza la contraseña del usuario en de la BBDD.
Postcondición	Pos-1: La aplicación indica que se ha cambiado la contraseña exitosamente.
Excepciones	Exc-1: Si la contraseña antigua no coincide, la aplicación muestra error y se comienza en el punto 2. Exc-2: Si las nuevas contraseña no son coincidentes, la aplicación muestra error y se vuelve al punto 2. Exc-3: Si las nuevas contraseña propuestas no cumple los requisitos de seguridad.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EU.02 Dep. - Reglas de negocio: RN.01, RN.02, RN.03
Observaciones	Requisitos de seguridad para las contraseñas: 8 a 41 caracteres, 1 letra mayúscula, 1 letra minúscula, 1 número y 1 carácter especial.

Tabla 115. Caso de uso UG9.

Id y caso de uso	UG10 - Ver publicaciones propias
Descripción	Un usuario accede a ver sus propias publicaciones.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que encontrarse consultando su perfil.
Secuencia	1. El usuario solicita ver sus publicaciones. 2. La aplicación consulta los datos de la BBDD sobre sus publicaciones.
Postcondición	Pos-1: La aplicación muestra las publicaciones del usuario.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EP.01, RI.EP.02, RI.EP.03, RI.EP.04, RI.EP.05, RI.EP.06, RI.EP.07, RI.EB.01, RI.EDU.02, RI.RA.01, RI.RA.02, RI.EB.01, RI.EDU.02 Dep. - Reglas de negocio: RN.01, RN.05, RN.16.
Observaciones	Desde el apartado de publicaciones propias, se pueden ver las caducadas y editarlas sin exceder de tres publicaciones activas.

Tabla 116. Caso de uso UG10.

Id y caso de uso	UG11 - Crear publicación
Descripción	Un usuario accede a crear su propia publicación.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que encontrarse consultando su perfil.
Secuencia	1. La aplicación ha consultado que haya menos de tres publicaciones activas al dirigirse a la consulta del propio perfil, permitiendo solicitar la creación de publicaciones en el menú (Exc-1). 2. El usuario añade los datos de la publicación, donde las fechas se enfrentan a una validación (Exc-2 y Exc-3). 3. La aplicación crea la publicación en la BBDD, y consulta si ya hay tres publicaciones activas para mostrar uno la opción de crear publicación (Exc-1).
Postcondición	Pos-1: El perfil cuenta con una nueva publicación. Pos-2: Si han añadido al autor de la publicación a favoritos y estos están activos y tiene configuradas las notificaciones de publicaciones, reciben un email.
Excepciones	Exc-1: Si hay ya hay tres publicaciones activas, la aplicación no muestra la opción de crear. Exc-2: Si las fechas son vencidas, la aplicación muestra error, volviendo al punto 2. Exc-3: Si la fecha fin no es mayor o igual a la fecha de inicio, muestra error, volviendo al punto 2.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EP.01, RI.EP.02, RI.EP.03, RI.EP.04, RI.EP.05, RI.EP.06, RI.EP.07, RI.EB.01, RI.EDU.02, RI.RA.01, RI.RA.02, RI.EB.01, RI.EDU.02 Dep. - Reglas de negocio: RN.01, RN.05, RN.08, RN.09
Observaciones	Solo se pueden crear un máximo de 3 publicaciones activas.

Tabla 117. Caso de uso UG11.

Id y caso de uso	UG12 - Editar publicación
Descripción	Un usuario accede a editar una publicación propia.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que encontrarse consultando su perfil y sus publicaciones.
Secuencia	1. El usuario puede solicitar la edición de su publicación genera una consulta para mostrar los datos modificables. 2. Los datos modificables se pueden editar y las nuevas fechas se enfrentan a la validación, además de validarse que no haya más de tres publicaciones activas y que las fechas sean correctas (Exc-1, Exc-2 y Exc-3). 3. La aplicación actualiza la publicación en la BBDD.
Postcondición	Pos-1: El perfil modifica una publicación existente. Pos-2: Si los que han añadido al autor de la publicación a favoritos están activos y tiene configuradas las notificaciones de publicaciones, reciben un email.
Excepciones	Exc-1: Si ya hay tres publicaciones activas, la aplicación no deja modificar dando lugar a cuatro publicaciones activas, mostrando un error. Exc-2: Si las fechas son vencidas, la aplicación muestra error, volviendo al punto 2.

	Exc-3: Si la fecha fin no es mayor o igual a la fecha de inicio, muestra error, volviendo al punto 2.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EP.01, RI.EP.02, RI.EP.03, RI.EP.04, RI.EP.05, RI.EP.06, RI.EP.07, RI.EB.01, RI.EDU.02, RI.RA.01, RI.RA.02, RI.EB.01, RI.EDU.02 Dep. - Reglas de negocio: RN.01, RN.05, RN.08, RN.09, RN.10
Observaciones	Se pueden editar publicaciones vencidas, sin exceder de 3 publicaciones activas.

Tabla 118. Caso de uso UG12.

Id y caso de uso	UG13 – Eliminar publicación
Descripción	Un usuario accede a borrar su propia publicación.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que encontrarse consultando su perfil y sus publicaciones.
Secuencia	1. La aplicación solicita la eliminación de su publicación. 2. La aplicación actualiza el estado de la publicación en la BBDD.
Postcondición	Pos-1: La publicación deja de ser visible para los usuarios excepto para los administradores.
Excepciones	No
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EP.01, RI.EP.07 Dep. - Reglas de negocio: RN.01, RN.16
Observaciones	Solo se actualiza el estado de la publicación a cero, como tal no se elimina de la BBDD. Las publicaciones siguen siendo visibles como inactivas para los administradores, a no ser que el perfil asociado a esa publicación se inactive.

Tabla 119. Caso de uso UG13.

Id y caso de uso	UG14 - Dar me gusta
Descripción	Un usuario accede dar me gusta a una publicación cualquiera incrementando el contador y en caso de volver a dar se retira el me gusta.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que encontrarse visualizando una publicación.
Secuencia	1. El usuario da al like de la publicación. 2. La publicación incrementa en uno el contador del like, actualizándose la BBDD. 3. Si el usuario vuelve a dar a like, el like dado se quita. 4. Si se vuelve a dar al like, el contador vuelve a aumentar, actualizándose la BBDD.
Postcondición	Pos-1: Si se da al like un número impar se incrementa el contador, a diferencia de si se da un número par de veces.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EP.01, RI.EL.01, RI.EL.02 Dep. - Reglas de negocio: RN.01, RN.09
Observaciones	

Tabla 120. Caso de uso UG14.

Id y caso de uso	UG15 - Consultar la lista propia de favoritos
Descripción	Un usuario accede a visualizar los aficionados que ha seleccionado como favoritos.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que encontrarse consultando su perfil.
Secuencia	1. El usuario solicita ver los usuarios añadidos a favoritos. 2. La aplicación consulta la BBDD para mostrar los usuarios añadidos a favoritos.
Postcondición	Pos-1: La aplicación muestra los usuarios añadidos a favoritos.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.RR.01, RI.RR.02, RI.RR.03 Dep. - Reglas de negocio: RN.01, RN.07
Observaciones	

Tabla 121. Caso de uso UG15.

Id y caso de uso	UG16 - Buscar perfiles
Descripción	Un usuario accede a explorar otros perfiles de otros aficionados.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que haber seleccionado aficionados en el menú de la cabecera.
Secuencia	1. El usuario puede hacer uso de los filtros para la búsqueda. 2. La aplicación consulta los datos de la BBDD para mostrar los perfiles.
Postcondición	Pos-1: La aplicación muestra los perfiles activos en la aplicación, según los criterios de búsqueda.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.ERA.01, RI.ERA.02, RI.RT1.01, RI.RT1.02, RI.RT1.02, RI.EDU.03, RI.ECI.01, RI.ECI.02, RI.EB.01, RI.EDU.02, RI.ECI.01, RI.ECI.02, RI.RDUB.01, RI.RDUB.02 Dep. - Reglas de negocio: RN.01, RN.05, RN.06, RN.17
Observaciones	Las perfiles desactivados no son visibles para los usuarios generales, pero sí para los administradores.

Tabla 122. Caso de uso UG16.

Id y caso de uso	UG17 - Ver perfiles
Descripción	Un usuario acceder a ver con más detalle un perfil.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación.
Secuencia	1. El usuario solicita al pulsar ver perfil cuando se ve el perfil poco detallado, ya sea tras seleccionar aficionados en el menú de la cabecera o en la lista de favoritos, o también al pulsar la foto o el nombre, ya sea desde de una publicación o del chat. 2. La aplicación consulta los datos de la BBDD de ese perfil.
Secuencia alternativa	No.
Postcondición	Pos-1: La aplicación muestra el perfil detallado según la configuración de privacidad.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EDU.01, RI.EDU.02, RI.EDU.03, RI.EDU.04, RI.EDU.05, RI.EDU.06, RI.EDU.07, RI.RDUB.01, RI.RDUB.02, RI.EB.01, RI.EDU.02, RI.ECI.01, RI.ECI.02 Dep. - Reglas de negocio: RN.01, RN.05, RN.06, RN.17
Observaciones	La configuración permite ocultar la edad y también el email, aunque el email siempre es visible en el perfil para los administradores.

Tabla 123. Caso de uso UG17.

Id y caso de uso	UG18 - Gestionar la inclusión como favorito
Descripción	Un usuario añade y quita de ser favorito un perfil de usuario.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación.
Secuencia	1. El usuario solicita al pulsar la estrella sobre fondo verde ya sea desde el perfil poco detallado o desde el perfil detallado. 2. La aplicación añade ese perfil como favorito de ese perfil en la BBDD. 3. Si el usuario vuelve a dar a la estrella sobre fonde verde. 4. Se quita de ser favorito en la BBDD.
Postcondición	Pos-1: Si se da un número de veces un impar se añade a favoritos, en caso de dar un número impar de veces se a diferencia de si se da un número par de veces. Pos-2: Tanto si se añade como si se quita de ser favorito es notificado por la pantalla.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.RR.01, RI.RR.02, RI.RR.03 Dep. - Reglas de negocio: RN.01, RN.07
Observaciones	Tanto los perfiles de usuarios generales como administradores se pueden añadir como ser añadidos como favoritos, así como dejar de estar añadidos.

Tabla 124. Caso de uso UG18.

Id y caso de uso	UG19 - Reportar perfil
Descripción	Un usuario deja constancia sobre un perfil inapropiado.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación.
Secuencia normal	1. El usuario solicita reportar al pulsar el triángulo con exclamación sobre fondo amarillo ya sea desde el perfil poco detallado o desde el perfil detallado. 2. La aplicación abre una ventana modal para reportar. 3. Se crea un nuevo reporte sobre el perfil en la BBDD.
Secuencia alternativa	No.
Postcondición	Pos-1: La aplicación muestra una notificación indicando que el reporte se ha realizado.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.ER.01, RI.ER.02, RI.ER.03, RI.ER.04, RI.ER.05, RI.EMR.01, RI.EMR.02 Dep. - Reglas de negocio: RN.01, RN.05, RN.11
Observaciones	Solo los usuarios administradores pueden ver los reportes.

Tabla 125. Caso de uso UG19.

Id y caso de uso	UG20 - Escribir mensaje privado al perfil
Descripción	Un usuario escribe al perfil de otro usuario por el chat privado.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario puede escribir a otro usuario si clica el sobre con fondo morado, ya sea desde el perfil poco detallado, más detallado o incluso de una publicación del usuario, siendo también posible desde el chat privado en caso de haber recibido previamente.
Secuencia	1. El usuario escribe en el recuadro para el envío de texto que se valida que haya texto antes de enviar (Exc-1). 2. La aplicación crea un mensaje en la BBDD.
Postcondición	Pos-1: La aplicación muestra una notificación de que el mensaje se ha enviado correctamente. Pos-2: Si el receptor del mensaje así lo ha configurado, recibe un email.
Excepciones	Exc-1: Si el recuadro de entrada de texto no tiene texto se da a enviar, la aplicación muestra un error volviendo al punto 2.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EM.01, RI.EM.02, RI.EM.03, RI.EM.04, RI.EP.05 Dep. - Reglas de negocio: RN.01, RN.05, RN.17
Observaciones	Desde la opción mensajes del menú de la cabecera solo es posible escribir a perfiles de usuarios si ellos han escrito previamente. Si un perfil se desactiva, el perfil deja de ser visible y su chat también, excepto para los usuarios administradores.

Tabla 126. Caso de uso UG20.

Id y caso de uso	UG21 - Buscar publicaciones
Descripción	Un usuario accede a explorar otras publicaciones de otros perfiles.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que haber seleccionado publicaciones en el menú de la cabecera.
Secuencia normal	1. El usuario puede hacer uso de los filtros para la búsqueda, en el que las fechas se enfrentan a validación Exc-1 y Exc-2). 2. La aplicación consulta los datos de la BBDD para mostrar las publicaciones.
Postcondición	Pos-1: La aplicación muestra las publicaciones activas y vigentes en la aplicación.
Excepciones	Exc-1: Si las fechas son vencidas, la aplicación muestra error, volviendo al punto 1. Exc-2: Si la fecha fin no es mayor o igual a la fecha de inicio, muestra error, volviendo al punto 1.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.ERA.01, RI.ERA.02, RI.RT1.01, RI.RT1.02, RI.EP.02, RI.EP.03, RI.EP.04, RI.EP.07, RI.RA.01, RI.RA.02, RI.EB.01, RI.EDU.02, RI.ECI.01, RI.ECI.02. Dep. - Reglas de negocio: RN.01, RN.05, RN.06, RN.16, RN.17
Observaciones	Las publicaciones desactivadas no son visibles para los usuarios generales, pero si para los administradores siempre y cuando no se haya desactivado el perfil, independiente si siguen vigentes, es decir, si han caducado o no.

Tabla 127. Caso de uso UG21.

Id y caso de uso	UG22 - Ver publicaciones
Descripción	Un usuario accede a explorar publicaciones de otros perfiles.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación.
Secuencia	1. Las publicaciones pueden ser vistas con el mismo nivel de detalle desde la opción aficionados del menú de la cabecera o desde el perfil, propio o ajeno., suponiendo una consulta para la BBDD.
Postcondición	Pos-1: La aplicación muestra diferentes publicaciones en función del acceso.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EP.01, RI.EP.02, RI.EP.03, RI.EP.04, RI.EP.05, RI.EP.06, RI.EP.07, RI.RA.01, RI.RA.02, RI.EB.01, RI.EDU.02, RI.ECI.01, RI.ECI.02 Dep. - Reglas de negocio: RN.01, RN.05, RN.06, RN.16, RN.17
Observaciones	Las publicaciones vencidas, es decir, caducadas, solo se pueden ver desde el perfil autor si inicia sesión el autor o desde la opción publicaciones del menú de cabecera por un usuario administrador.

Tabla 128. Caso de uso UG22.

Id y caso de uso	UG23 - Reportar publicación
Descripción	Un usuario deja constancia sobre una publicación inapropiada.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación.
Secuencia	1. El usuario solicita reportar generando una consulta con motivos para seleccionar uno, pudiendo añadir un comentario al reporte. 2. Se crea un nuevo reporte sobre la publicación en la BBDD.
Postcondición	Pos-1: La aplicación muestra una notificación indicando que el reporte se ha realizado.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EP.01, RI.ER.01, RI.ER.02, RI.ER.03, RI.ER.04, RI.ER.05, RI.EMR.01, RI.EMR.02, RI.EMR.01, RI.EMR.02 Dep. - Reglas de negocio: RN.01, RN.05, RN.11
Observaciones	

Tabla 129. Caso de uso UG23.

Id y caso de uso	UG24 - Acceder al chat
Descripción	Un usuario accede al chat
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que haber seleccionado mensajes en el menú de la cabecera.
Secuencia	1. El usuario ha generado una consulta cargando los usuarios y una conversación.
Postcondición	Pos-1: La aplicación muestra los usuarios quedando más arriba aquellos que han enviado un mensaje más reciente y mostrando la conversación del que ha quedado arriba.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.ERA.01, RI.ERA.02, RI.RT1.01, RI.RT1.02, RI.EM.01, RI.EM.02, RI.EM.03, RI.EM.04, RI.EP.05 Dep. - Reglas de negocio: RN.01, RN.05, RN.17
Observaciones	En el chat solo aparecen los usuarios con los que ha habido algún mensaje con el usuario logeado. Si un usuario es inactivado, desaparece del chat, excepto para los usuarios administradores.

Tabla 130. Caso de uso UG24.

Id y caso de uso	UG25 - Visualizar mensajes
Descripción	Un usuario accede a ver los mensajes mantenidos con un usuario.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que haber seleccionado mensajes en el menú de la cabecera.
Secuencia	1. El usuario solicita ver la conversación mantenida con otro usuario. 2. La aplicación genera una consulta en la BBDD para mostrar la conversación.
Postcondición	Pos-1: La aplicación muestra la conversación de un usuario.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.ERA.01, RI.ERA.02, RI.RT1.01, RI.RT1.02, RI.EM.01, RI.EM.02, RI.EM.03, RI.EM.04, RI.EP.05 Dep. - Reglas de negocio: RN.01, RN.05, RN.17
Observaciones	

Tabla 131. Caso de uso UG25.

Id y caso de uso	UG26 - Marcar como leído
Descripción	Un usuario accede a ver una conversación no vista, desapareciendo el número de mensajes no leídos.
Actor	Usuario general.
Precondición	Pre-1: El usuario tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario tiene que haber seleccionado mensajes en el menú de la cabecera.
Secuencia normal	1. El usuario al ver la conversación mantenida con otro usuario, la cual no había visto antes. 2. La aplicación actualiza el atributo leído de la BBDD.
Postcondición	Pos-1: La aplicación deja de mostrar el número de mensajes sin leer al lado del usuario.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.EM.01, RI.EP.05
Observaciones	

Tabla 132. Caso de uso UG26.

Usuario administrador

Id y caso de uso	UA1 - Buscar perfil reportado
Descripción	Un usuario administrador accede a ver perfiles con reportes.
Actor	Usuario administrador. (Exclusivamente).
Precondición	Pre-1: El usuario administrador tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario administrador tiene que haber seleccionado aficionados en el menú de la cabecera.
Secuencia	1. El usuario administrador puede hacer uso de los filtros para la búsqueda. 2. La aplicación consulta los datos de la BBDD para mostrar los perfiles.
Postcondición	Pos-1: La aplicación muestra los perfiles en la aplicación, según los criterios de búsqueda.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.ERA.01, RI.ERA.02, RI.RT1.01, RI.RT1.02, RI.RT1.02, RI.EDU.03, RI.ECI.01, RI.ECI.02, RI.EB.01, RI.EDU.02, RI.ECI.01, RI.ECI.02, RI.RDUB.01, RI.RDUB.02, RI.EU.05, RI.EDU.04, RI.EDU.07, RI.ER.01, RI.ER.02, RI.ER.03, RI.ER.04, RI.ER.05, RI.EMR.01, RI.EMR.02 Dep. - Reglas de negocio: RN.01, RN.05, RN.12, RN.13, RN.16, RN.17
Observaciones	Los usuarios administradores también ven los perfiles inactivos.

Tabla 133. Caso de uso UA1.

Id y caso de uso	UA2 - Buscar publicación reportada
Descripción	Un usuario administrador accede a ver publicaciones con reportes de otros perfiles.
Actor	Usuario administrador. (Exclusivamente).
Precondición	Pre-1: El usuario administrador tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario administrador tiene que haber seleccionado publicaciones en el menú de la cabecera.
Secuencia	1. El usuario administrador puede hacer uso de los filtros para la búsqueda. 2. La aplicación consulta los datos de la BBDD para mostrar las publicaciones.
Postcondición	Pos-1: La aplicación muestra las publicaciones según los filtros, incluyendo las no vigentes, es decir, cuya fecha de finalización ha pasado.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.ERA.01, RI.ERA.02, RI.RT1.01, RI.RT1.02, RI.EP.02, RI.EP.03, RI.EP.04, RI.EP.07, RI.RA.01, RI.RA.02, RI.EB.01, RI.EDU.02, RI.ECI.01, RI.ECI.02, RI.EP.05, RI.EP.07, RI.ER.01, RI.ER.02, RI.ER.03, RI.ER.04, RI.ER.05, RI.EMR.01, RI.EMR.02 Dep. - Reglas de negocio: RN.01, RN.05, RN.12, RN.13, RN.16, RN.17
Observaciones	Los usuarios administradores también ven las publicaciones inactivas. Pero sin un perfil está inactivado, sus publicaciones dejan de ser visibles para los usuarios administradores.

Tabla 134. Caso de uso UA2.

Id y caso de uso	UA3 - Visualizar reportes del perfil
Descripción	Un usuario administrador accede a visualizar los reportes de un perfil.
Actor	Usuario administrador. (Exclusivamente).
Precondición	Pre-1: El usuario administrador tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario administrador tiene que estar viendo el perfil desde la opción aficionados del menú de la cabecera donde los perfiles se muestran con poco detalle y dar al triángulo sobre fondo amarillo cuando el perfil muestra pocos detalle.
Secuencia	1. El usuario administrador solicita acceder a los reportes del perfil. 2. La aplicación realiza una consulta de los reportes asociados a ese perfil en la BBDD.
Postcondición	Pos-1: La aplicación muestra los reportes asociados a ese perfil.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.ERA.01, RI.ERA.02, RI.RT1.01, RI.RT1.02, RI.ER.01, RI.ER.02, RI.ER.03, RI.ER.04, RI.ER.05, RI.EMR.01, RI.EMR.02 Dep. - Reglas de negocio: RN.01, RN.05, RN.12, RN.13, RN.16, RN.17
Observaciones	El usuario administrador si diera al triángulo sobre fondo amarillo cuando el perfil muestra más detalle podría acceder a reportar, por si fuera necesario como anotación visible para otro administrador.

Tabla 135. Caso de uso UA3.

Id y caso de uso	UA4 – Visualizar reportes de la publicación
Descripción	Un usuario administrador accede a visualizar los reportes de una publicación.
Actor	Usuario administrador. (Exclusivamente).
Precondición	Pre-1: El usuario administrador tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario administrador tiene que estar viendo la publicación desde la opción publicaciones del menú de la cabecera.
Secuencia normal	1. El usuario administrador solicita acceder a los reportes de la publicación. 2. La aplicación realiza una consulta de los reportes asociados a esa publicación en la BBDD.
Postcondición	Pos-1: La aplicación muestra los reportes asociados a esa publicación.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.ERA.01, RI.ERA.02, RI.RT1.01, RI.RT1.02, RI.EP.01, RI.ER.01, RI.ER.02, RI.ER.03, RI.ER.04, RI.ER.05, RI.EMR.01, RI.EMR.02 Dep. - Reglas de negocio: RN.01, RN.05, RN.12, RN.13, RN.16, RN.17
Observaciones	El usuario administrador si diera a los tres puntos que despliegan un triángulo gris cuando las publicaciones se ven desde el perfil mostrando más detalles, podría acceder a reportar la publicación, por si fuera necesario como anotación visible para otro administrador.

Tabla 136. Caso de uso UA4.

Id y caso de uso	UA5 - Gestionar la activación del perfil
Descripción	Un usuario administrador accede a desactivar y activar un perfil.
Actor	Usuario administrador. (Exclusivamente).
Precondición	Pre-1: El usuario administrador tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario administrador tiene que estar viendo el perfil desde la opción aficionados del menú de la cabecera donde los perfiles se muestran con poco detalle y con un cuadrado sobre fondo verde que puede tener o no un check en función de si está activado o no.
Secuencia normal	1. El usuario administrador solicita modificar la activación del perfil. 2. La aplicación actualiza el estado de ese perfil en la BBDD.
Postcondición	Pos-1: La aplicación muestra un aviso de que se ha desactivado o activado el perfil.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.ERA.01, RI.ERA.02, RI.RT1.01, RI.RT1.02, RI.EU.05 Dep. - Reglas de negocio: RN.01, RN.05, RN.13, RN.16, RN.17
Observaciones	Con check el perfil está activo y sin check desactivado. Inactivar un perfil, implica solo los usuarios administradores pueden ver su perfil y su chat, en cambio, sus publicaciones dejan de verse incluso para los usuarios administradores, a no ser que el usuario se vuelva a activar.

Tabla 137. Caso de uso UA5.

Id y caso de uso	UA6 - Gestionar la activación de la publicación
Descripción	Un usuario administrador accede a desactivar y activar una publicación.
Actor	Usuario administrador. (Exclusivamente).
Precondición	Pre-1: El usuario administrador tiene que haber iniciado sesión en la aplicación. Pre-2: El usuario administrador tiene que estar viendo la publicación desde la opción publicaciones del menú de la cabecera donde aparecen las publicaciones con un cuadrado sobre fondo verde que puede tener o no un check en función de si está activada o no.
Secuencia normal	1. El usuario administrador solicita modificar la activación de la publicación. 2. La aplicación actualiza el estado de esa publicación en la BBDD.
Postcondición	Pos-1: La aplicación muestra un aviso de que se ha desactivado o activado la publicación.
Excepciones	No.
Dependencias	Dep. - Requisitos de información: RI.EU.01, RI.ERA.01, RI.ERA.02, RI.RT1.01, RI.RT1.02, RI.EP.07 Dep. - Reglas de negocio: RN.01, RN.05, RN.13, RN.16, RN.17
Observaciones	Con check la publicación está activa y sin check desactivada. Inactivar un perfil, implica solo los usuarios administradores pueden ver su perfil y su chat, en cambio, sus publicaciones dejan de verse incluso para los usuarios administradores, a no ser que el usuario se vuelva a activar.

Tabla 138. Caso de uso UA6.

Anexo III: Requisitos funcionales

En el presente anexo se adjuntan el resto de los requisitos de usuario que no han sido incluidos en el apartado “4.4 Requisitos funcionales”. Los requisitos funcionales están organizados por actores siguiendo el orden de la tabla (ver tabla 14) incluida en el apartado “4.2.1 Listado de casos de uso”.

Usuario general

ID	Caso de uso
UG1	Ver perfil propio
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.023	El sistema consultará en la base de datos los datos del usuario, sus publicaciones y sus favoritos.
RF.024	El sistema mostrará por pantalla los datos del usuario.

Tabla 139. Requisitos funcionales en el caso de uso UG1.

ID	Caso de uso
UG2	Consultar configuración propia
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.025	El sistema consultará en la base de datos la configuración.
RF.026	El sistema mostrará por pantalla la configuración.

Tabla 140. Requisitos funcionales en el caso de uso UG2.

ID	Caso de uso
UG3	Gestionar las notificaciones por email de los mensajes recibidos
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.027	El sistema actualizará la nueva configuración del perfil sobre las notificaciones por email de los mensajes recibidos.
RF.028	El sistema redirigirá a la página de configuración de usuario.
RF.025	El sistema consultará en la base de datos la configuración.
RF.026	El sistema mostrará por pantalla la configuración.

Tabla 141. Requisitos funcionales en el caso de uso UG3.

ID	Caso de uso
UG4	Gestionar las notificaciones por email de las publicaciones de los favoritos
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.029	El sistema actualizará la nueva configuración del perfil sobre la notificación por email de las publicaciones de los favoritos.
RF.028	El sistema redirigirá a la página de configuración de usuario.
RF.025	El sistema consultará en la base de datos la configuración.
RF.026	El sistema mostrará por pantalla la configuración.

Tabla 142. Requisitos funcionales en el caso de uso UG4.

ID	Caso de uso
UG5	Gestionar la visibilidad de la edad
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.030	El sistema actualizará la nueva configuración del perfil sobre la visibilidad de la edad.
RF.028	El sistema redirigirá a la página de configuración de usuario.
RF.025	El sistema consultará en la base de datos la configuración.
RF.026	El sistema mostrará por pantalla la configuración.

Tabla 143. Requisitos funcionales en el caso de uso UG5.

ID	Caso de uso
UG6	Gestionar la visibilidad del email
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.031	El sistema actualizará la nueva configuración del perfil sobre la visibilidad del email.
RF.028	El sistema redirigirá a la página de configuración de usuario.
RF.025	El sistema consultará en la base de datos la configuración.
RF.026	El sistema mostrará por pantalla la configuración.

Tabla 144. Requisitos funcionales en el caso de uso UG6.

ID	Caso de uso
UG7	Editar perfil
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.032	El sistema mostrará un error si el formato de la foto no es el correcto y/o tiene un tamaño excesivo.
RF.033	El sistema mostrará una ventana modal para el cambio de los datos tras seleccionar modificar perfil.
RF.034	El sistema actualizará los nuevos datos del perfil en la base de datos.
RF.035	El sistema redirigirá a la página de su propio perfil.
RF.023	El sistema consultará en la base de datos los datos del usuario, sus publicaciones y sus favoritos.
RF.024	El sistema mostrará por pantalla los datos del usuario.

Tabla 145. Requisitos funcionales en el caso de uso UG7.

ID	Caso de uso
UG8	Eliminar perfil
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.036	El sistema verificará que no se trata de un administrador para permitir la opción.
RF.037	El sistema mostrará un formulario tras seleccionar eliminar perfil para introducir la contraseña antigua.
RF.038	El sistema mostrará una advertencia de la implicación de la acción, en caso de cancelación no seguirá, pero si se acepta seguirá.
RF.013	El sistema comprobará si coinciden las contraseñas cifradas.
RF.039	El sistema mostrará un error en caso de que la contraseña no sea correcta.
RF.040	El sistema actualizará el estado del usuario en caso de ser la contraseña correcta.
RF.020	El sistema invalidará la sesión de usuario abierta.
RF.021	El sistema redirigirá al usuario a la página de login.

Tabla 146. Requisitos funcionales en el caso de uso UG8.

ID	Caso de uso
UG9	Cambiar contraseña
ID	Requisitos funcionales
RF.041	El sistema mostrará un formulario de cambio de contraseña tras seleccionar cambiar contraseña.
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.013	El sistema comprobará si coinciden las contraseñas cifradas.
RF.003	El sistema validará que la nueva contraseña sea segura.
RF.004	El sistema validará que ambas contraseñas introducidas sean iguales.
RF.042	El sistema mostrará un error en caso de que la contraseña antigua no coincida con la almacenada, la nueva no sea segura o coincida con la repetición.
RF.043	El sistema actualizará la nueva contraseña de manera cifrada en caso de superar las comprobaciones.
RF.044	El sistema mostrará que la contraseña se ha cambiado de forma exitosa.

Tabla 147. Requisitos funcionales en el caso de uso UG9.

ID	Caso de uso
UG10	Ver publicaciones propias
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.023	El sistema consultará en la base de datos los datos del usuario, sus publicaciones y sus favoritos.
RF.045	El sistema mostrará por pantalla sus publicaciones.
RF.046	El sistema indicará que no hay publicaciones en caso de que no lo haya.

Tabla 148. Requisitos funcionales en el caso de uso UG10.

ID	Caso de uso
UG11	Crear publicación
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.047	El sistema solo presentará la opción de crear publicación si no hay tres publicaciones activas por parte del perfil.
RF.048	El sistema abre una ventana modal para la creación de una publicación.
RF.049	El sistema mostrará un error en caso de las fechas no sean vigentes o la fecha fin sea mayor o igual que la de inicio.
RF.050	El sistema creará una nueva publicación en la base de datos.
RF.051	El sistema creará un email para enviar a los perfiles activos que tengan configuradas las notificaciones de publicaciones por email y tengan al perfil del autor añadido como favorito.
RF.035	El sistema redirigirá a la página de su propio perfil.
RF.023	El sistema consultará en la base de datos los datos del usuario, sus publicaciones y sus favoritos.
RF.024	El sistema mostrará por pantalla los datos del usuario.

Tabla 149. Requisitos funcionales en el caso de uso UG11.

ID	Caso de uso
UG12	Editar publicación
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.052	El sistema solo presentará la opción de editar cuando el usuario visualice sus propias publicaciones.
RF.053	El sistema tomará el identificador de la publicación.
RF.054	El sistema abrirá una ventana modal para la edición de la publicación.
RF.049	El sistema mostrará un error en caso de las fechas no sean vigentes o la fecha fin sea mayor o igual que la de inicio.
RF.054	El sistema mostrará un error indicando que no es posible superar el límite de publicaciones vigentes si ya hay 3 publicaciones activas.
RF.055	El sistema actualizará la publicación en la base de datos.
RF.051	El sistema creará un email para enviar a los perfiles activos que tengan configuradas las notificaciones de publicaciones por email y tengan al perfil del autor añadido como favorito.
RF.035	El sistema redirigirá a la página de su propio perfil.
RF.023	El sistema consultará en la base de datos los datos del usuario, sus publicaciones y sus favoritos.
RF.024	El sistema mostrará por pantalla los datos del usuario.

Tabla 150. Requisitos funcionales en el caso de uso UG12.

ID	Caso de uso
UG13	Eliminar publicación
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión iniciada.
RF.056	El sistema solo presentará la opción de eliminar cuando el usuario visualice sus propias publicaciones.
RF.053	El sistema tomará el identificador de la publicación.
RF.057	El sistema actualizará el estado de la publicación en la base de datos.
RF.058	El sistema alertará de que la publicación será borrada.
RF.059	El sistema dejará de mostrar por pantalla las publicación eliminada.
RF.046	El sistema indicará que no hay publicaciones en caso de que no lo haya.

Tabla 151. Requisitos funcionales en el caso de uso UG13.

ID	Caso de uso
UG14	Dar me gusta
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión iniciada.
RF.060	El sistema creará un registro en la base de datos con la identificación del usuario y de la publicación en caso de existir, de lo contrario lo eliminará.
RF.061	El sistema mostrará en la pantalla el contador de likes de la publicación actualizada.

Tabla 152. Requisitos funcionales en el caso de uso UG14.

ID	Caso de uso
UG15	Consultar la lista propia de favoritos
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.023	El sistema consultará en la base de datos los datos del usuario, sus publicaciones y sus favoritos.
RF.062	El sistema mostrará por pantalla sus favoritos.
RF.063	El sistema indica que no hay favoritos en caso de que no los haya.

Tabla 153. Requisitos funcionales en el caso de uso UG15.

ID	Caso de uso
UG16	Buscar perfiles
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.036	El sistema verificará que no se trata de un administrador para permitir la opción.
RF.064	El sistema mostrará los filtros de búsqueda de perfiles para aficionados.
RF.065	El sistema mostrará en función de la selección de filtros de búsqueda de perfiles para aficionados los perfiles.
RF.066	El sistema mostrará solo los perfiles activos con la posibilidad de ser reportados.

Tabla 154. Requisitos funcionales en el caso de uso UG16.

ID	Caso de uso
UG17	Ver perfiles
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.067	El sistema solo presentará perfiles desactivados a administradores.
RF.023	El sistema consultará en la base de datos los datos del usuario, sus publicaciones y sus favoritos.
RF.068	El sistema permite ver los detalles del perfil y sus publicaciones vigentes y activas desde este, pero no los perfiles añadidos a favoritos.
RF.069	El sistema mostrará los detalles del perfil según configuración, con la excepción del email que siempre será visible para los administradores.
RF.070	Los perfiles solo darán acceso a la visualización de reportes y a la gestión de activación a los administradores según salen como resultado de búsqueda.
RF.071	Los perfiles siempre darán acceso a ser añadidos y quitados como favoritos, la escritura de mensajes privados, y ser reportados.

Tabla 155. Requisitos funcionales en el caso de uso UG17.

ID	Caso de uso
UG18	Gestionar la inclusión como favorito
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión iniciada.
RF.072	El sistema creará un registro en la base de datos con la identificación del usuario y del usuario destino añadido como favorito en caso de existir, de lo contrario lo eliminará.
RF.073	El sistema mostrará una alerta indicando si el usuario se ha hecho favorito o lo ha dejado de ser.

Tabla 156. Requisitos funcionales en el caso de uso UG18.

ID	Caso de uso
UG19	Reportar perfil
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.074	El sistema abrirá una ventana modal para creación de un reporte.
RF.075	El sistema creará un reporte en la base de datos recogiendo la identificación de lo reportado y el reportador, además de un motivo y un posible comentario.

Tabla 157. Requisitos funcionales en el caso de uso UG19.

ID	Caso de uso
UG20	Escribir mensaje privado al perfil
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.079	El sistema abrirá una ventana modal con un recuadro para escribir un mensaje en caso de que no le haya para el envío del mensaje.
RF.076	El sistema no permitirá el envío sin texto con una alerta.
RF.077	El sistema creará un mensaje con el identificador del emisor y receptor, además del contenido del mensaje, con la fecha y como no leído.
RF.078	El sistema enviará un email en caso del que el receptor tengan activadas las notificaciones por email de los mensajes recibidos.
RF.079	El sistema informará con una alerta cuando se haya enviado el mensaje correctamente.

Tabla 158. Requisitos funcionales en el caso de uso UG20.

ID	Caso de uso
UG21	Buscar publicaciones
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.036	El sistema verificará que no se trata de un administrador para permitir la opción.
RF.080	El sistema mostrará filtros de búsqueda de publicaciones para aficionado.
RF.081	El sistema mostrará en función de la selección de filtros de búsqueda de publicaciones para aficionados las publicaciones.
RF.082	El sistema mostrará solo las publicaciones activas con la posibilidad de ser reportadas.

Tabla 159. Requisitos funcionales en el caso de uso UG21.

ID	Caso de uso
UG22	Ver publicaciones
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.083	El sistema solo presentará publicaciones desactivadas a administradores desde el buscador siempre y cuando no esté el perfil del autor desactivado.
RF.023	El sistema consultará en la base de datos los datos del usuario, sus publicaciones y sus favoritos.
RF.084	El sistema permite ver las publicaciones vigentes y activas desde el perfil.
RF.085	Los publicaciones solo darán acceso a la visualización de reportes y a la gestión de activación a los administradores según salen como resultado de búsqueda.
RF.086	Las poblaciones ajenas siempre darán acceso al autor, a la escritura de mensajes privados al autor, y ser reportadas.

Tabla 160. Requisitos funcionales en el caso de uso UG22.

ID	Caso de uso
UG23	Reportar publicación
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.074	El sistema abrirá una ventana modal para creación de un reporte.
RF.078	El sistema creará un reporte en la base de datos recogiendo la identificación de lo reportado y el reportador, además de un motivo y un posible comentario.

Tabla 161. Requisitos funcionales en el caso de uso UG23.

ID	Caso de uso
UG24	Acceder al chat
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.087	El sistema verificará no se trate de un administrador para ocultar los perfiles desactivados.
RF.088	El sistema consultará todos los perfiles de usuario con los que se ha mantenido conversación.
RF.089	El sistema mostrará una lista de perfiles con los que ha mantenido conversación mostrando más arriba los que han mandado más recientemente un mensaje.
RF.090	El sistema indicará que no hay mensajes en caso de que no se haya mantenido conversación con ningún perfil.

Tabla 162. Requisitos funcionales en el caso de uso UG24.

ID	Caso de uso
UG25	Visualizar mensajes
ID	Requisitos funcionales
RF.091	El sistema mostrará por defecto la conversación del perfil situado más arriba.
RF.092	El sistema mostrará la conversación de cualquier otro perfil tras su selección.
RF.093	El sistema consultará los mensajes mantenidos con los perfiles.
RF.094	El sistema agrupará los mensajes por perfiles quedando en la parte inferior los más recientes.

Tabla 163. Requisitos funcionales en el caso de uso UG25.

ID	Caso de uso
UG26	Marcar como leído
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.095	El sistema actualizará los mensajes en la base de datos poniéndoles como leídos.
RF.096	El sistema dejará de mostrar el contador de los mensajes de no leídos de ese usuario.

Tabla 164. Requisitos funcionales en el caso de uso UG26.

Usuario administrador

ID	Caso de uso
UA1	Buscar perfil reportado
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.100	El sistema verificará que se trata de un administrador para permitir la opción.
RF.97	El sistema mostrará los filtros de búsqueda para administradores de perfiles inactivos y reportados.
RF.98	El sistema mostrará en función de la selección de filtros de búsqueda para administradores de perfiles los perfiles.
RF.99	El sistema mostrará los perfiles con la posibilidad de ver sus reportes y de inactivarse.

Tabla 165. Requisitos funcionales en el caso de uso UA1.

ID	Caso de uso
UA2	Buscar publicación reportada
ID	Requisitos funcionales
RF.22	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.100	El sistema verificará que se trata de un administrador para permitir la opción.
RF.101	El sistema mostrará los filtros de búsqueda para administradores de publicaciones inactivas y reportadas.
RF.102	El sistema mostrará en función de la selección de filtros de búsqueda para administradores de publicaciones las publicaciones.
RF.0103	El sistema mostrará publicaciones vencidas e inactivas siempre y cuando no se haya inactivado su perfil.
RF.104	El sistema mostrará las publicaciones con la posibilidad de ver sus reportes y de inactivarse.

Tabla 166. Requisitos funcionales en el caso de uso UA2.

ID	Caso de uso
UA3	Visualizar reportes del perfil
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.100	El sistema verificará que se trata de un administrador para permitir la opción.
RF.105	El sistema consultará en la base de datos los reportes asociados al identificador de los reportados.
RF.106	El sistema abrirá una ventana modal sin identificador de lo reportado para visualización de los reportes.
RF.107	El sistema mostrará por pantalla los reportes.

Tabla 167. Requisitos funcionales en el caso de uso URN1.

ID	Caso de uso
UA4	Visualizar reportes de la publicación
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.100	El sistema verificará que se trata de un administrador para permitir la opción.
RF.105	El sistema consultará en la base de datos los reportes asociados al identificador de los reportado.
RF.108	El sistema abrirá una ventana modal con identificador de lo reportado para visualización de los reportes.
RF.107	El sistema mostrará por pantalla los reportes.

Tabla 168. Requisitos funcionales en el caso de uso UA4.

ID	Caso de uso
UA5	Gestionar la activación del perfil
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.100	El sistema verificará que se trata de un administrador para permitir la opción.
RF.109	El sistema tomará el identificador del perfil a actualizar el estado.
RF.110	El sistema mostrará el perfil desactivado.

Tabla 169. Requisitos funcionales en el caso de uso UA5.

ID	Caso de uso
UA6	Gestionar la activación de la publicación
ID	Requisitos funcionales
RF.022	El sistema recuperará el identificador del usuario que tiene la sesión abierta.
RF.100	El sistema verificará que se trata de un administrador para permitir la opción.
RF.111	El sistema tomará el identificador de la publicación a actualizar el estado.
RF.112	El sistema mostrará la publicación desactivada.

Tabla 170. Requisitos funcionales en el caso de uso UA6.

Anexo IV: Contenido del repositorio

El contenido del repositorio está compuesto por los siguientes directorios organizando así el contenido tal y como refleja la siguiente tabla (ver tabla 171).

Directorio	Contenido	Ficheros
Memoria	Memoria del TFG <i>Bailando en pareja</i> .	○ RubenBurgosPerela_BailandoEnPareja_TFG.pdf
Diagramas	Diagramas que por tamaño podrían no ser visualizados adecuadamente en la memoria.	○ Arquitectura lógica ○ Entidad-relación ○ Gantt ○ Diagramas de clases
Archivos del proyecto	Archivos pertenecientes a la aplicación desarrollada.	○ BBDD ○ WebMVC ○ ServiciosREST

Tabla 171. Contenido del repositorio.