



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

ESTUDIO Y DESARROLLO DE UN PROTOTIPO PARA GESTIÓN Y VENTA DE LOCALIDADES DE EVENTOS DEPORTIVOS

Alumno: Ramón Bailén Sánchez

Tutor: Prof. D. José Ramón Balsas Almagro
Dpto: Informática

Junio, 2015



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don José Ramón Balsas Almagro , tutor del Proyecto Fin de Carrera titulado: Aplicación web para gestionar la venta de localidades de eventos deportivos a través de Internet, que presenta Don Ramón Bailén Sánchez, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Junio de 2015

El alumno:

Los tutores:

D. Ramón Bailén Sánchez

D. José Ramón Balsas Almagro

Agradecimientos

En primer lugar, quisiera mostrar mi agradecimiento a José Ramón Balsas Almagro, por la oportunidad que me ha brindado de trabajar con él en la realización de este proyecto, además de la dedicación y actitud positiva mostrada durante estos meses.

A mis compañeros de clase, por haberme hecho dar lo mejor de mí, en afán de superarlos.

A mis compañeros del SeTIC, por su convivencia diaria y por su amistad.

A los profesores particulares con los que he pasado gran parte de mi tiempo, especialmente a Eva.

A mi tío José Carlos, por emplear tiempo y esfuerzo en explicarme ciertas asignaturas.

A mis compañeros y amigos Eduardo García Lendínez y Antonio Ruiz Amate, con los que he vivido y compartido grandes momentos dentro de la universidad y fuera.

A mi novia Gloria, por su apoyo, cariño y especial comprensión en el tiempo dedicado al desarrollo de este proyecto.

A mi hermana Laura, por hacerme ver que estar lejos, no significa no estar cerca.

A mis padres, por la confianza constante y por ser la base que ha hecho posible que llegara a mi meta.

A demás familiares y amigos que me han apoyado en esto.

Gracias a todos.

Índice de contenido

1. Introducción.....	14
1.1. Propósito.....	15
1.2. Objetivos	16
1.3. Metodología	17
1.4. Estructura del documento	18
2. Análisis.....	20
2.1. Análisis preliminar	21
2.1.1. Estado del arte sobre la venta de localidades en Internet	22
2.2. Pasarela integral de PayPal	27
2.2.1. Funcionamiento general.....	28
2.3. Entornos de desarrollo web.....	29
2.3.1. Spring MVC.....	30
2.3.2. JavaServer Faces (JSF).....	31
2.3.3. Bootstrap.....	32
2.3.4. Pure	33
2.4. Propuesta de solución.....	34
2.5. Requerimientos del sistema	35
2.5.1. Requerimientos funcionales	36
2.5.2. Requerimientos no funcionales	44
2.5.2.1. Requerimientos de la interfaz.....	44
2.6. Planificación de tareas	46
2.6.1. Estructura del plan de trabajo.....	46
2.6.2. Estimación de tiempos	47
2.6.3. Diagrama de Gantt.....	48
2.7. Estudio de viabilidad	50
2.8. Casos de uso	54
2.8.1. Especificación de casos de uso	61
2.9. Modelo de dominio	77
3. Diseño.....	79
3.1. Diseño de la representación de la información.....	80
3.1.1. Modelo Entidad-Relación	81
3.1.2. Tablas de la aplicación.....	83

3.2.	Diagrama de Clases.....	89
3.3.	Diagrama de Secuencia.....	99
3.4.	Diseño de la Interfaz	113
3.4.1.	Estilo	113
3.4.2.	Metáforas.....	115
3.4.3.	Prototipos de interfaz	116
3.4.4.	Notificaciones y mensajes de error.....	125
3.4.4.1.	Notificaciones.....	125
3.4.4.2.	Mensajes de error	127
4.	Implementación.....	130
4.1.	Arquitectura.....	131
4.2.	Detalles sobre implementación	133
4.2.1.	Entorno de desarrollo	133
4.2.2.	Lenguajes de programación	135
4.2.3.	Tecnologías utilizadas.....	137
4.2.4.	Interacción con PayPal.....	151
4.2.4.1.	Entorno de pruebas de pago	154
4.2.5.	Generación de código QR.....	156
4.3.	Pruebas.....	157
4.3.1.	Casos de prueba.....	158
4.3.2.	Resultados obtenidos.....	213
5.	Conclusiones.....	214
5.1.	Mejoras y trabajos futuros	216
6.	Bibliografía	218
	Bibliografía	219
	Apéndice I. Registro de cuenta en PayPal.....	222
	Apéndice II. Manual de Usuario.....	227
	Apéndice III. Descripción de contenidos suministrados	252

Tabla de Ilustraciones

Ilustración 1: Ticketmaster.....	23
Ilustración 2: Tengoentradas.com	24
Ilustración 3: Atrápalo.....	25
Ilustración 4: Viagogo.....	26
Ilustración 5: Pasarela integral	28
Ilustración 6: Estructura del plan de trabajo.....	46
Ilustración 7: Actividades.....	48
Ilustración 8: Diagrama de Gantt	49
Ilustración 9: Sistema	55
Ilustración 10: Diagrama Frontera	56
Ilustración 11: Identificación	57
Ilustración 12: Gestión de Usuarios	57
Ilustración 13: Gestión de deportes	58
Ilustración 14: Gestión de competiciones	58
Ilustración 15: Gestión de participantes.....	59
Ilustración 16: Gestión de eventos	59
Ilustración 17: Gestión de entradas	60
Ilustración 18: Gestión de compras	60
Ilustración 19: Gestión de datos personales.....	61
Ilustración 20: Modelo de Dominio	78
Ilustración 21: Modelo Entidad-Relación	82
Ilustración 22: Diagrama de paquetes MVC	90
Ilustración 23: Ejemplo MVC "Alta de usuario"	91
Ilustración 24: Controladores & Servicios	92
Ilustración 25: Controladores & Modelos	93
Ilustración 26: Servicios & Daos	94
Ilustración 27: Servicios & Modelos	95
Ilustración 28: Daos & Modelos	96
Ilustración 29: Daos & Excepciones	97
Ilustración 30: Editor.....	98
Ilustración 31: DS "Identificación"	100
Ilustración 32: DS "Registrarse"	101
Ilustración 33: DS "Listado de usuarios".....	102
Ilustración 34: DS "Editar deporte"	103
Ilustración 35: DS "Eliminar competición".....	104
Ilustración 36: DS "Alta de evento"	105
Ilustración 37: DS "Puesta en venta"	106
Ilustración 38: DS "Visualizar compra"	107
Ilustración 39: DS "Buscar eventos"	108
Ilustración 40: DS "Comprar entradas"	109
Ilustración 41: DS "Pagar entradas".....	110
Ilustración 42: DS "Ver compras realizadas"	111

Ilustración 43: DS "Imprimir entradas"	112
Ilustración 44: Layout	114
Ilustración 45: Prototipo 1 "Pantalla principal"	117
Ilustración 46: Prototipo 2 "Menú inicio administrador"	117
Ilustración 47: Prototipo 3 "Listado de usuarios"	118
Ilustración 48: Prototipo 4 "Alta de usuario"	118
Ilustración 49: Prototipo 5 "Editar usuario"	119
Ilustración 50: Prototipo 6 "Alta de competición"	119
Ilustración 51: Prototipo 7 "Editar participante"	120
Ilustración 52: Prototipo 8 "Alta de evento"	120
Ilustración 53: Prototipo 9 "Visualizar compra"	121
Ilustración 54: Prototipo 10 "Menú inicio cliente"	121
Ilustración 55: Prototipo 11 "Visualizar datos personales cliente"	122
Ilustración 56: Prototipo 12 "Ver deportes eventos"	122
Ilustración 57: Prototipo 13 "Categorías deporte"	123
Ilustración 58: Prototipo 14 "Eventos deporte"	123
Ilustración 59: Prototipo 15 "Comprar entradas"	124
Ilustración 60: Prototipo 16 "Confirmación de compra"	124
Ilustración 61: Notificación "Registro correcto"	125
Ilustración 62: Notificación "Eliminar"	125
Ilustración 63: Notificación "Salir"	126
Ilustración 64: Notificación "Cancelar compra"	126
Ilustración 65: Notificación "Alta de competición"	126
Ilustración 66: Notificación "Alta de participante"	127
Ilustración 67: Notificación "Alta de evento"	127
Ilustración 68: Notificación "Alta entrada"	127
Ilustración 69: Mensaje de error "Identificación"	127
Ilustración 70: Mensaje de error "Formulario 1"	128
Ilustración 71: Mensaje de error "Formulario 2"	128
Ilustración 72: Mensaje de error "Formulario 3"	128
Ilustración 73: Mensaje de error "Formulario 4"	128
Ilustración 74: Mensaje de error "Formulario 5"	129
Ilustración 75: Mensaje de error "Formulario 6"	129
Ilustración 76: Mensaje de error "Búsqueda de eventos"	129
Ilustración 77: Mensaje de error "Selección de entradas"	129
Ilustración 78: Arquitectura	131
Ilustración 79: JPA Anotaciones	136
Ilustración 80: JPA Integración	136
Ilustración 81: Creación de las tablas	137
Ilustración 82: Transacción	138
Ilustración 83: EHCache	139
Ilustración 84: EHCache Import	139
Ilustración 85: Cacheado	139
Ilustración 86: CacheEvict	139
Ilustración 87: Ejemplo controlador	140
Ilustración 88: Configuración anotaciones	141
Ilustración 89: Recursos	141
Ilustración 90: Recursos JSP	141

Ilustración 91: Estilo Doctype HTML5.....	142
Ilustración 92: Estilo Responsive	142
Ilustración 93: Grid de Bootstrap	142
Ilustración 94: Estilo iconos	143
Ilustración 95: Estilo tablas	144
Ilustración 96: Estilo error	144
Ilustración 97: CSS Ejemplo 1	145
Ilustración 98: CSS Ejemplo 2	145
Ilustración 99: HTML	146
Ilustración 100: Filtro Spring Security	147
Ilustración 101: Autenticación	147
Ilustración 102: Seguridad URLs	147
Ilustración 103: Import security.xml	147
Ilustración 104: Directiva JSTL	148
Ilustración 105: Ejemplo JSTL	148
Ilustración 106: Integración BB.DD.....	149
Ilustración 107: Bean Validation	149
Ilustración 108: Ejemplo mensaje	150
Ilustración 109: Bean Validation Configuración	150
Ilustración 110: Etiqueta error.....	150
Ilustración 111: Pago estándar	151
Ilustración 112: Botón de pagar.....	152
Ilustración 113: Código fuente PayPal.....	153
Ilustración 114: Entorno de desarrollo PayPal	155
Ilustración 115: Confirmación de pago	155
Ilustración 116: Implementación QR.....	156
Ilustración 117: Código QR	156
Ilustración 118: Formulario alta cuenta.....	223
Ilustración 119: Confirmación cuenta.....	224
Ilustración 120: Inicio sitio pruebas.....	224
Ilustración 121: Opción Accounts	225
Ilustración 122: Cuentas	225
Ilustración 123: Account details	226
Ilustración 124: Pantalla principal	228
Ilustración 125: Menú Administrador	229
Ilustración 126: Gestión de usuarios.....	229
Ilustración 127: Editar usuario	230
Ilustración 128: Alta de usuario	231
Ilustración 129: Listado de deportes	232
Ilustración 130: Editar deporte.....	232
Ilustración 131: Alta de deporte	233
Ilustración 132: Listado de competiciones	233
Ilustración 133: Editar competición	234
Ilustración 134: Alta de competición	234
Ilustración 135: Listado de participantes.....	235
Ilustración 136: Listado de eventos	235
Ilustración 137: Visualizar evento	236
Ilustración 138: Alta de evento	236

Ilustración 139: Editar evento	237
Ilustración 140: Listado de entradas	238
Ilustración 141: Alta de entrada	238
Ilustración 142: Editar entrada.....	239
Ilustración 143: Puesta en venta	240
Ilustración 144: Listado de compras	240
Ilustración 145: Visualizar compra.....	241
Ilustración 146: Menú cliente.....	241
Ilustración 147: Datos personales	242
Ilustración 148: Editar cliente.....	242
Ilustración 149: Ver eventos	243
Ilustración 150: Categorías deporte.....	243
Ilustración 151: Evento localizado	244
Ilustración 152: Entradas deportes	244
Ilustración 153: Confirmación de compra	245
Ilustración 154: Resumen PayPal.....	246
Ilustración 155: Pagar PayPal	247
Ilustración 156: Confirmación de pago	248
Ilustración 157: Histórico compras.....	249
Ilustración 158: Visualizar compra.....	249
Ilustración 159: Imprimir	250
Ilustración 160: Búsqueda	251

Relación de tablas

Tabla I: Gastos personal	50
Tabla II: Equipos informáticos	50
Tabla III: Periféricos	51
Tabla IV: Licencias software.....	52
Tabla V: Costes finales.....	53
Tabla VI: Caso de uso "Identificación"	62
Tabla VII: Caso de uso "Registrarse"	63
Tabla VIII: Caso de uso "Listado de usuarios"	64
Tabla IX: Caso de uso "Editar deporte"	65
Tabla X: Caso de uso "Eliminar competición".....	66
Tabla XI: Caso de uso "Alta de evento"	67
Tabla XII: Caso de uso "Puesta en venta"	68
Tabla XIII: Caso de uso "Visualizar Compra".....	69
Tabla XIV: Caso de uso "Buscar eventos"	70
Tabla XV: Caso de uso "Comprar entradas".....	71
Tabla XVI: Caso de uso "Pagar entradas"	73
Tabla XVII: Caso de uso "Ver compras realizadas"	74
Tabla XVIII: Caso de uso "Imprimir entradas"	76
Tabla XIX: Relaciones	83
Tabla XX: Tabla de usuarios	83
Tabla XXI: Tabla de deportes	84
Tabla XXII: Tabla de competiciones	84
Tabla XXIII: Tabla de participantes.....	85
Tabla XXIV: Tabla de países	85
Tabla XXV: Tabla de eventos	86
Tabla XXVI: Tabla "participa"	86
Tabla XXVII: Tabla de entradas.....	87
Tabla XXVIII: Tabla de compras.....	87
Tabla XXIX: Tabla "compras_entradas"	88
Tabla XXX: Metáforas	116
Tabla XXXI: Prueba 1.....	158
Tabla XXXII: Prueba 2.....	158
Tabla XXXIII: Prueba 3.....	159
Tabla XXXIV: Prueba 4	159
Tabla XXXV: Prueba 5	160
Tabla XXXVI: Prueba 6	160
Tabla XXXVII: Prueba 7	160
Tabla XXXVIII: Prueba 8	161
Tabla XXXIX: Prueba 9	161
Tabla XL: Prueba 10	161
Tabla XLI: Prueba 11	162
Tabla XLII: Prueba 12	162

Tabla XLIII: Prueba 13.....	162
Tabla XLIV: Prueba 14	163
Tabla XLV: Prueba 15	163
Tabla XLVI: Prueba 16	163
Tabla XLVII: Prueba 17	164
Tabla XLVIII: Prueba 18	164
Tabla XLIX: Prueba 19	164
Tabla L: Prueba 20.....	165
Tabla LI: Prueba 21.....	165
Tabla LII: Prueba 22.....	165
Tabla LIII: Prueba 23.....	166
Tabla LIV: Prueba 24	166
Tabla LV: Prueba 25	166
Tabla LVI: Prueba 26	167
Tabla LVII: Prueba 27	167
Tabla LVIII: Prueba 28.....	167
Tabla LIX: Prueba 29	168
Tabla LX: Prueba 30	168
Tabla LXI: Prueba 31	168
Tabla LXII: Prueba 32	169
Tabla LXIII: Prueba 33.....	169
Tabla LXIV: Prueba 34	169
Tabla LXV: Prueba 35	170
Tabla LXVI: Prueba 36	170
Tabla LXVII: Prueba 37	170
Tabla LXVIII: Prueba 38	171
Tabla LXIX: Prueba 39	171
Tabla LXX: Prueba 40	171
Tabla LXXI: Prueba 41	172
Tabla LXXII: Prueba 42	172
Tabla LXXIII: Prueba 43	173
Tabla LXXIV: Prueba 44.....	173
Tabla LXXV: Prueba 45.....	173
Tabla LXXVI: Prueba 46.....	174
Tabla LXXVII: Prueba 47.....	174
Tabla LXXVIII: Prueba 48.....	174
Tabla LXXIX: Prueba 49.....	175
Tabla LXXX: Prueba 50.....	175
Tabla LXXXI: Prueba 51.....	175
Tabla LXXXII: Prueba 52.....	176
Tabla LXXXIII: Prueba 53.....	176
Tabla LXXXIV: Prueba 54	176
Tabla LXXXV: Prueba 55	177
Tabla LXXXVI: Prueba 56	177
Tabla LXXXVII: Prueba 57	178
Tabla LXXXVIII: Prueba 58	178
Tabla LXXXIX: Prueba 59	179
Tabla XC: Prueba 60.....	179

Tabla XCI: Prueba 61	179
Tabla XCII: Prueba 62	180
Tabla XCIII: Prueba 63	180
Tabla XCIV: Prueba 64	180
Tabla XCV: Prueba 65	181
Tabla XCVI: Prueba 66	181
Tabla XCVII: Prueba 67	182
Tabla XCVIII: Prueba 68	182
Tabla XCIX: Prueba 69	182
Tabla C: Prueba 70	183
Tabla CI: Prueba 71	183
Tabla CII: Prueba 72	183
Tabla CIII: Prueba 73	184
Tabla CIV: Prueba 74	184
Tabla CV: Prueba 75	185
Tabla CVI: Prueba 76	185
Tabla CVII: Prueba 77	186
Tabla CVIII: Prueba 78	186
Tabla CIX: Prueba 79	186
Tabla CX: Prueba 80	187
Tabla CXI: Prueba 81	187
Tabla CXII: Prueba 82	187
Tabla CXIII: Prueba 83	188
Tabla CXIV: Prueba 84	188
Tabla CXV: Prueba 85	188
Tabla CXVI: Prueba 86	189
Tabla CXVII: Prueba 87	189
Tabla CXVIII: Prueba 88	189
Tabla CXIX: Prueba 89	190
Tabla CXX: Prueba 90	190
Tabla CXXI: Prueba 91	190
Tabla CXXII: Prueba 92	191
Tabla CXXIII: Prueba 93	191
Tabla CXXIV: Prueba 94	191
Tabla CXXV: Prueba 95	192
Tabla CXXVI: Prueba 96	192
Tabla CXXVII: Prueba 97	192
Tabla CXXVIII: Prueba 98	193
Tabla CXXIX: Prueba 99	193
Tabla CXXX: Prueba 100	193
Tabla CXXXI: Prueba 101	194
Tabla CXXXII: Prueba 102	194
Tabla CXXXIII: Prueba 103	195
Tabla CXXXIV: Prueba 104	195
Tabla CXXXV: Prueba 105	196
Tabla CXXXVI: Prueba 106	196
Tabla CXXXVII: Prueba 107	196
Tabla CXXXVIII: Prueba 108	197

Tabla CXXXIX: Prueba 109.....	197
Tabla CXL: Prueba 110.....	197
Tabla CXLI: Prueba 111.....	198
Tabla CXLII: Prueba 112.....	198
Tabla CXLIII: Prueba 113.....	198
Tabla CXLIV: Prueba 114.....	199
Tabla CXLV: Prueba 115.....	199
Tabla CXLVI: Prueba 116.....	199
Tabla CXLVII: Prueba 117.....	200
Tabla CXLVIII: Prueba 118.....	200
Tabla CXLIX: Prueba 119.....	200
Tabla CL: Prueba 120.....	201
Tabla CLI: Prueba 121.....	201
Tabla CLII: Prueba 122.....	201
Tabla CLIII: Prueba 123.....	202
Tabla CLIV: Prueba 124.....	202
Tabla CLV: Prueba 125.....	202
Tabla CLVI: Prueba 126.....	203
Tabla CLVII: Prueba 127.....	203
Tabla CLVIII: Prueba 128.....	203
Tabla CLIX: Prueba 129.....	204
Tabla CLX: Prueba 130.....	204
Tabla CLXI: Prueba 131.....	205
Tabla CLXII: Prueba 132.....	205
Tabla CLXIII: Prueba 133.....	206
Tabla CLXIV: Prueba 134.....	206
Tabla CLXV: Prueba 135.....	206
Tabla CLXVI: Prueba 136.....	207
Tabla CLXVII: Prueba 137.....	208
Tabla CLXVIII: Prueba 138.....	209
Tabla CLXIX: Prueba 139.....	210
Tabla CLXX: Prueba 140.....	211
Tabla CLXXI: Prueba 141.....	212
Tabla CLXXII: Prueba 142.....	213

1. Introducción

En este primer capítulo se determinan cuáles son los propósitos y objetivos del proyecto realizado. Se especifican las metodologías utilizadas y se explica cómo está estructurada la memoria.

1.1. Propósito

En este proyecto se plantea el estudio y desarrollo de una aplicación web para gestionar la venta de localidades de eventos deportivos a través de Internet. El cliente tendrá que estar registrado en el sistema para la adquisición de entradas. Se utilizará PayPal como forma de pago para comercio electrónico por medio de Internet [1].

Con carácter general, puede considerarse que la elección del proyecto viene determinada por algunos de los siguientes factores:

- Clara inclinación por el desarrollo de una aplicación web
- Implementación de un sistema útil hoy en día
- Gran interés en el ámbito deportivo
- Ampliación del conocimiento acerca del funcionamiento de Paypal

Los dos primeros factores son concluyentes a la hora de realizar este proyecto, debido al uso cotidiano de este tipo de aplicaciones y las ventajas que presentan, como por ejemplo el ahorro de tiempo o la posibilidad de compra por adelantado.

El tercer factor fue motivo de reflexión durante un período de tiempo. En un principio se planteó la posibilidad de desarrollar una aplicación web relacionada con un deporte en concreto, pero tras un exhausto estudio que posteriormente se describe se tomó la decisión de llevar a cabo un diseño en el que tienen cabida diversos deportes.

Por último, el uso masivo de internet supone una revolución tecnológica. Esto ha significado un cambio en los hábitos de compra de millones de personas. Las pasarelas de pago ofrecen una gran variedad de servicios que se adaptan a cada tipo de negocio, por lo que es la oportunidad adecuada para ahondar en este asunto e incluir en el sistema una forma de pago para comercio electrónico.

1.2. Objetivos

En este apartado vamos a exponer cuáles son los objetivos iniciales que se pretenden cubrir con este proyecto y que serán tratados a lo largo de la presente memoria:

- Estudio de necesidades para la venta de entradas a eventos deportivos a través de Internet. Se llevará a cabo un análisis acerca de la descripción del sistema o contexto del que se parte y para el que se realizará el desarrollo. La finalidad de este objetivo es examinar las soluciones existentes y las carencias detectadas.
- Estudio de uso de Paypal como pasarela de pago para comercio electrónico a través de Internet. Se detallará el funcionamiento general de la plataforma de pago y se elaborará un breve manual sobre cómo realizar la integración de Paypal en una aplicación web.
- Diseñar un sistema informático para gestionar la venta de localidades de eventos deportivos a través de Internet. Es el objetivo general de este proyecto. Esto significa que se requiere de la cumplimentación del resto de objetivos para alcanzar el objetivo final.
- Implementar un prototipo de aplicación web utilizando el framework Spring MVC. Se justificará el uso de dicho framework en la propuesta de solución y se determinará la arquitectura y las tecnologías empleadas en el proceso para posteriormente realizar un conjunto de pruebas que permitan verificar y validar el sistema desarrollado.

1.3. Metodología

Para el desarrollo del software, se ha seguido el ciclo de vida clásico [2], el cual sugiere un enfoque secuencial en el avance de las actividades, de manera que el inicio de cada una de ellas debe esperar a la finalización de la anterior. A pesar de la aparición de tecnologías ágiles como SCRUM [3], sigue siendo la solución predominante.

Aparte de ser fácil de implementar y entender, esta metodología presenta la ventaja de promover una disciplina de trabajo efectiva: definir antes que diseñar, diseñar antes que codificar. Sin embargo, un proyecto rara vez sigue una secuencia lineal en la vida real [2].

El ciclo de vida clásico se ha dividido entre las fases de análisis, la cual comprende desde la obtención de los requerimientos iniciales hasta la realización de casos de uso, la fase de diseño, centrada en la descripción del sistema sin entrar en características de implementación y las fases de implementación y pruebas, donde se construye el sistema y se comprueba el funcionamiento del mismo.

En función del modelo escogido, se propone el empleo de las siguientes metodologías durante las fases de desarrollo del software anteriores:

- Estudio de los requerimientos del sistema a desarrollar. Se llevará a cabo un análisis para conocer las necesidades de las aplicaciones de venta de entradas deportivas, así como del sistema de pago de PayPal.
- Recopilación bibliográfica sobre la temática y tecnologías relacionadas.
- Estimación temporal y de costes del desarrollo para conocer la viabilidad del proyecto, de manera que sea competitivo con otras soluciones del mercado.
- Diseño del sistema empleando una metodología de orientación a objetos utilizando el patrón de diseño Modelo-Vista-Controlador.
- Implementación del prototipo del sistema.

1.4. Estructura del documento

En este apartado se realiza una breve introducción acerca de los diferentes capítulos en los que se estructura la memoria del proyecto realizado. El presente documento se ha estructurado de la siguiente manera:

Como hemos podido ver, en este capítulo se ha elaborado una breve introducción del proyecto, indicando su propósito, los objetivos a alcanzar y las metodologías empleadas, así como una planificación temporal.

Los tres siguientes capítulos constituyen el núcleo fundamental de la memoria, ya que en ellos se describen las etapas de desarrollo de la ingeniería del software. En el capítulo segundo se llevará a cabo un estudio preliminar acerca del ámbito del proyecto para posteriormente formular una propuesta de solución. A continuación se definirán los requerimientos funcionales y no funcionales del sistema, además de incluirse el estudio de viabilidad del proyecto, el modelo de casos de uso, etc. El tercer capítulo está dedicado a la etapa del diseño del software. En él se adjuntan el diagrama de clases, el diagrama de entidad-relación y los principales diagramas de secuencia. Adicionalmente, se incluye el diseño de la interfaz implementada. En el cuarto y último capítulo perteneciente a la ingeniería del software se establece el tipo de arquitectura de la aplicación y se definen los detalles fundamentales sobre la implementación del sistema, como pueden ser los lenguajes o tecnologías de programación utilizados. Por otra parte en este mismo capítulo se establecen las pruebas que se llevarán a cabo para comprobar el correcto funcionamiento del software.

El quinto capítulo se dedicará a las conclusiones obtenidas durante el desarrollo del proyecto, así como se abordarán posibles mejoras y propuestas de trabajos para el futuro.

El capítulo sexto incluye las referencias bibliográficas utilizadas tanto para la implementación del proyecto como para el desarrollo de la memoria, de manera que se podrán consultar en caso de necesitar más información.

Por último, la sección final de la memoria contiene varios apéndices: el primero está dedicado a la adquisición de una cuenta en PayPal, el segundo incluye el

manual de usuario, donde se describe paso a paso cómo utilizar la aplicación y el tercero presenta una descripción de los contenidos suministrados en el CD del proyecto.

2. Análisis

En esta etapa se debe comprender de forma detallada cual es la problemática a resolver, verificando el entorno en el cual se encuentra dicho problema, de tal manera que se obtenga la información necesaria y suficiente para afrontar su respectiva solución [2].

2.1. Análisis preliminar

Internet es una herramienta que permite el acceso a grandes cantidades de información. Cada vez es mayor el número de empresas que ofertan sus productos a través de Internet. Este hecho implica un incremento de la competencia entre empresas que ofrezcan servicios del mismo tipo, por lo que es vital la búsqueda de productos diferenciadores, que atraigan a nuevos clientes y mantengan los actuales. Además, el uso de aplicaciones web ha adquirido bastante protagonismo en los últimos años. Se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador [4]. Presentan muchas ventajas, como pueden ser la ausencia de problemas de compatibilidad entre sistemas operativos y la no utilización de espacio en el disco duro, ya que todo el código se encuentra en el servidor. Todo esto nos proporciona las características necesarias para el desarrollo de este proyecto.

En el ámbito de la venta de localidades, muchas entidades deportivas proporcionan un sistema de adquisición de entradas a través de su página web. Dicho sistema puede ser útil si el usuario es aficionado del equipo en cuestión y tiene claro que desea asistir a un evento relacionado con alguna de las secciones del club.

Por otro lado existen numerosos portales que ofrecen la posibilidad de comprar entradas relacionadas con cualquier tipo de evento: deporte, cine, arte, teatro, festivales, música, etc. En este caso, toda esta información no siempre resulta de utilidad, ya que el usuario puede sentirse saturado por la multitud de temáticas ofrecidas. La limitación del tiempo del que el cliente dispone es un factor a tener en cuenta.

Por todo ello es conveniente disponer de un sistema web donde el cliente pueda consultar de manera rápida y con detalle todos los eventos deportivos disponibles.

2.1.1. Estado del arte sobre la venta de localidades en Internet

Dentro del ámbito de este proyecto, se pueden encontrar diversas soluciones ya existentes. Es importante tener en cuenta que estas soluciones están enfocadas hacia dos vertientes: por un lado se encuentran los sistemas de venta de entradas pertenecientes a clubes deportivos y por otro lado plataformas que proporcionan un servicio de adquisición de localidades para diferentes actividades.

En primer lugar, los análisis realizados hasta el momento sobre sistemas de venta de localidades correspondientes a conjuntos deportivos determinan una propuesta de servicios claramente personalizada para las distintas entidades deportivas. Las prestaciones que ofrece el medio de venta de un equipo pueden no ser válidas para otro. Esta teoría se basa fundamentalmente en la magnitud de los diferentes clubes. Algunos de ellos disponen de varias secciones, por lo que habilitan un mayor número de instalaciones deportivas. Cada una ellas establece una diferente distribución de los asientos en el campo y un número de asientos predeterminado. Además, se ha podido apreciar la carencia de una herramienta de este tipo en un elevado número de conjuntos deportivos.

En relación a la segunda vertiente, se describen a continuación algunos sistemas con características similares al desarrollado en este proyecto y posteriormente, en la propuesta de solución ([capítulo 2.4](#)), se establecerá una leve comparación entre lo que ofrecen estos sistemas y lo que pretendemos lograr con este:

1) Ticketmaster



Ilustración 1: Ticketmaster

Como puede observarse, esta compañía proporciona un servicio de venta de entradas [5] para deportes y adicionalmente para otras temáticas, como pueden ser la música y el cine. Lógicamente, si un usuario desea realizar una búsqueda acerca de los eventos existentes en una ciudad determinada, no sólo le aparecerán los relacionados con el deporte sino que también los vinculados a las restantes temáticas.

2) Tengoentradas.com

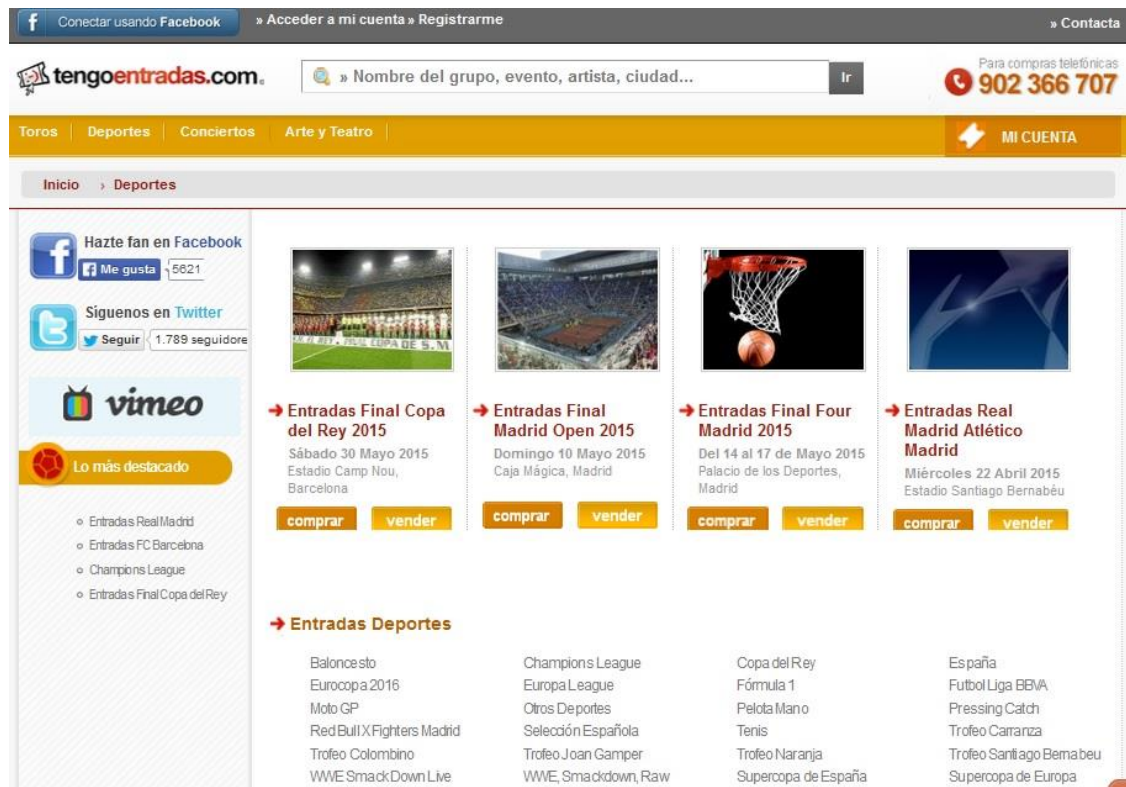


Ilustración 2: Tengoentradas.com

Es una plataforma donde se llevan a cabo operaciones de compra y venta de entradas [6]. Al igual que ticketmaster, permite adquirir entradas para eventos de diferentes materias. Si nos centramos concretamente en la deportiva y un usuario utiliza el buscador para filtrar una búsqueda de eventos dado un deporte o un deportista, no se obtiene ningún resultado, ya que los criterios de búsqueda utilizados son: nombre del grupo, evento, artista y ciudad. Sin embargo, en la parte inferior hay un menú que contiene los deportes y las competiciones más relevantes del momento. A través de este menú, el usuario sí puede acceder al deporte que desee, pero tiene que darse cuenta de la existencia del mismo.

3) Atrapalo.com

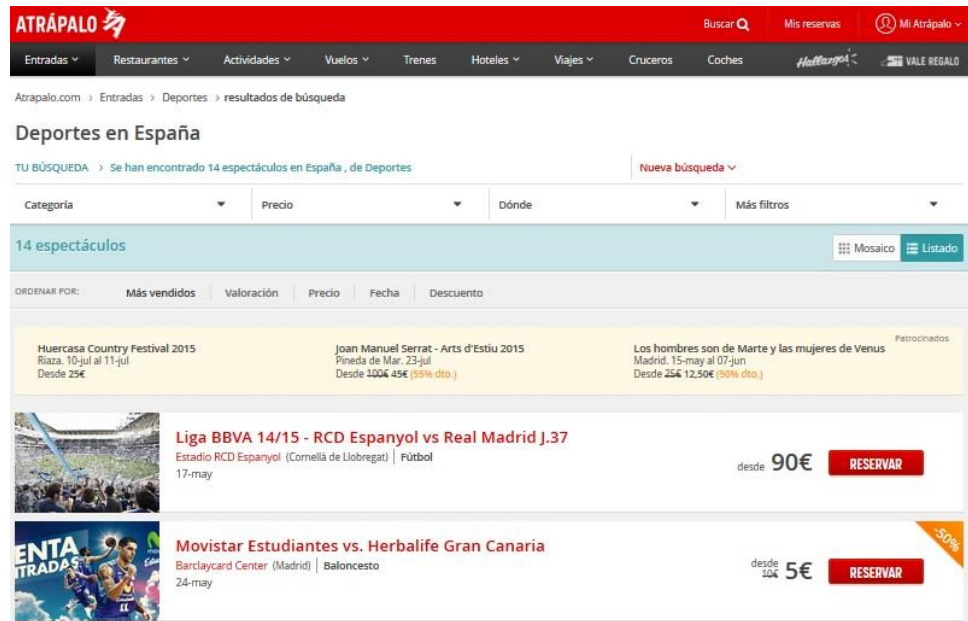


Ilustración 3: Atrapalo

Como puede apreciarse, es una plataforma de promoción de ofertas de ocio, restaurantes y servicios turísticos, tales como viajes, hoteles y alquiler de coches [7]. Presenta un rango amplio de productos e incluye la venta de entradas para deportes, musicales, monólogos y otras actividades. Para facilitar la búsqueda al usuario, dispone de un buscador que rastrea los eventos existentes dada una población espectáculo o artista, además de ofrecer la posibilidad de filtrar los eventos por categoría (deporte), precio y lugar (provincia).

4) Viagogo

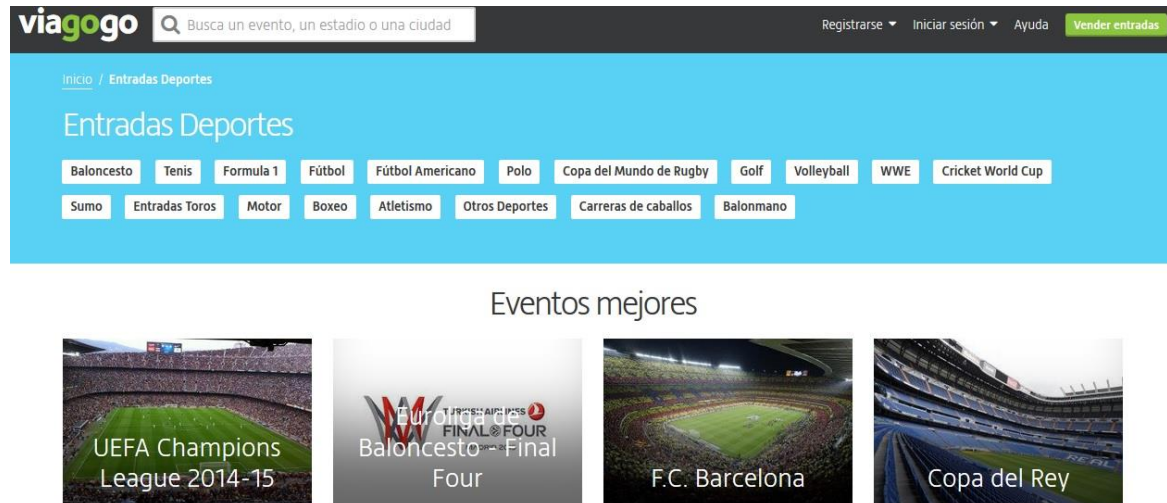


Ilustración 4: Viagogo

Proporciona un servicio de intercambio de entradas de conciertos, festivales, deportes y teatro [8]. El sitio facilita a los usuarios que desean comprar entradas la posibilidad de encontrar usuarios que desean venderlas. La sección de deportes dispone de un menú bastante completo, además de ofrecer al usuario una lista con los mejores eventos (adjuntados en la imagen), los más recomendados y los más cercanos dada una ubicación.

2.2. Pasarela integral de PayPal

Pasarela integral de PayPal es una plataforma de pago que permite a los vendedores recibir pagos con fondos ingresados con tarjeta de crédito o una cuenta PayPal. No tiene que capturar ni almacenar información de tarjetas de crédito en su sitio web. Es la opción que utilizan los vendedores que prefieren una solución en la que PayPal gestione todos sus datos financieros [9].

Algunas funciones y ventajas de Pasarela integral son la siguientes:

- **Cumplimiento de PCI DSS** (*Payment Card Industry Data Security Standard*, significa Estándar de Seguridad de Datos para la Industria de Tarjeta de Pago): estándar de seguridad global que se aplica a todas las empresas que recopilan, almacenan, procesan o transmiten información de titulares de tarjeta.
- **Protección del vendedor:** el programa le protege contra devoluciones de cargo de transacciones con fondos ingresados con tarjeta de crédito o una cuenta PayPal.
- **Pago estándar de PayPal:** es la forma más sencilla de añadir PayPal como opción de pago. El cliente realiza el pago y completa el pedido en PayPal.
- **Pago exprés de PayPal:** el cliente es dirigido a PayPal y autoriza su uso para el pago, pero al contrario que la opción anterior, realiza el pago y completa la compra en su sitio web.
- **Admite las principales tarjetas de crédito y débito**, como por ejemplo Visa o MasterCard
- **Compatibilidad con dispositivos móviles:** cuando un comprador ve la página de pago desde un navegador móvil compatible, PayPal muestra automáticamente un flujo de pago optimizado para móviles en lugar de la plantilla de diseño

El procedimiento a seguir para la creación de una cuenta en PayPal y su puesta en marcha se detalla en el [apéndice I](#) de la memoria.

2.2.1. Funcionamiento general

La figura siguiente muestra el flujo llevado a cabo para pagar con la cuenta PayPal (parte superior) y para pagar con tarjeta (parte inferior):

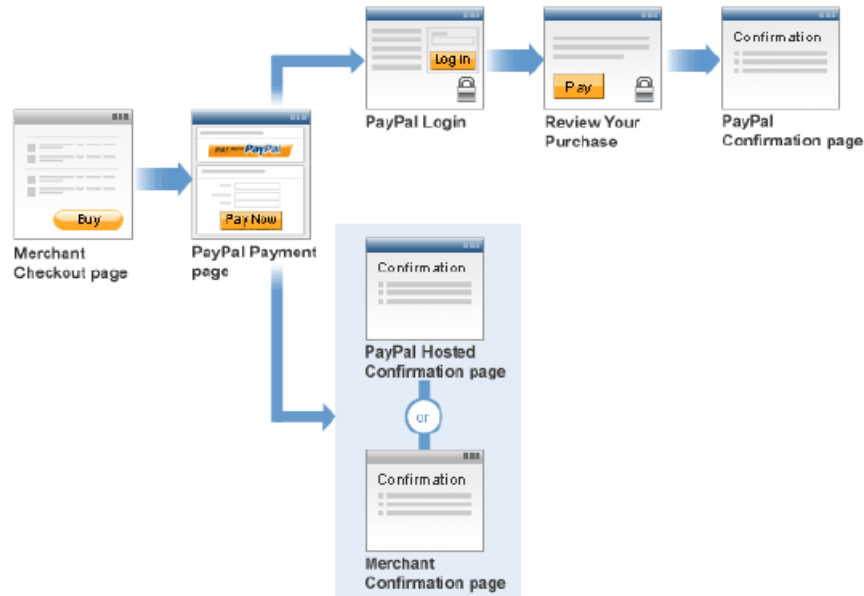


Ilustración 5: Pasarela integral

Una vez que el cliente haya seleccionado los artículos deseados a través de la aplicación web, accionará el botón de **comprar** o **pagar**, según la etiqueta utilizada para el botón. Automáticamente, es redirigido a la página de pago alojada por PayPal. En ella, el comprador introduce la información de la cuenta de PayPal o bien la de la tarjeta de crédito o débito y hace click en el botón **Pagar ahora**. Si la transacción es correcta, el comprador visualiza la página de confirmación de PayPal o es redirigido a una URL especificada. Si por el contrario no es correcta, se muestra un mensaje de error para que el comprador reintente la transacción.

2.3. Entornos de desarrollo web

En este punto estudiaremos el panorama de marcos de desarrollo web en el lado del servidor para la plataforma Java y en el lado del cliente, con el objetivo de seleccionar la mejor opción para implementar el prototipo del sistema.

Un framework es una colección de software con un comportamiento genérico que puede ser personalizado para realizar un comportamiento específico. El principal objetivo que persigue un framework es la definición de un esquema en el que se congreguen principios de diseño, patrones y buenas prácticas que permitan la implementación de aplicaciones con características o necesidades similares [10]. Entre otras herramientas puede incorporar bibliotecas de código que sirven de ayuda para unir los distintos elementos de un proyecto.

La importancia de utilizar frameworks radica en que ofrecen a los programadores facilidades para el diseño, desarrollo y mantenimiento del sistema.

Los frameworks de desarrollo JVM (Java Virtual Machine) pueden seguir dos tipos de arquitectura [10]:

- Push-Based (o action based)

Se realiza un enrutamiento (URLs) a acciones (controlador) que envían los datos a las vistas. Algunos ejemplos de esta arquitectura son Spring MVC o Django.

- Pull-Based (o component based)

Es una alternativa a la arquitectura anterior. Se realiza un enrutamiento (URLs) a vistas que extraen los datos de controladores. Cabe destacar como ejemplo de esta arquitectura el framework de desarrollo web JavaServer Faces (JSF).

El modelo-vista-controlador es un aspecto importante de cualquier aplicación web. Este modelo arquitectónico define la separación del código en tres elementos diferentes: modelos, vistas y controladores. Existen numerosos frameworks que implementan este patrón. La mayoría de ellos son OpenSource (software cuyo código fuente está disponible para el uso y/o modificación por parte de cualquier persona).

Por tanto, para poner en marcha el proyecto se ha realizado un análisis que permite determinar las propuestas tecnológicas más adecuadas para la evolución del sistema. En este proyecto se va a llevar a cabo el estudio y desarrollo de una aplicación web para gestionar la venta de localidades de eventos deportivos a través de Internet, tal y como se indica en el propósito de la memoria ([capítulo 1.1](#)), con tecnología Java.

El ritmo al que aparecen nuevas librerías y frameworks para Java hace de esta tecnología una de las más interesantes. Es complicado elegir las herramientas que mejor se adapten a nuestro sistema, además de combinarlas eficientemente para aprovechar toda la flexibilidad y potencia de éstas.

Dentro de Java y en el ámbito específico de aplicaciones web, los frameworks Spring MVC y JSF, mencionados anteriormente, son dos opciones interesantes. A continuación se describen las principales características de cada uno de ellos:

2.3.1. Spring MVC

Spring es un framework para el desarrollo de aplicaciones para la plataforma Java. Se basa en la utilización de objetos de negocio denominados beans, que se definen como pojos (clases de objetos sencillas sin implementación de interfaces) que residen en un contenedor IoC (inversion of control) [11].

El contexto de la aplicación se define en un archivo con extensión XML. Spring permite que la configuración se encuentre en este fichero.

Spring maneja toda la infraestructura, de manera que el usuario puede centrarse en la aplicación, ocupándose solamente de la lógica de la misma y de la configuración del framework. Esto quiere decir que esta tecnología fusiona todos los componentes de la aplicación, gobierna su ciclo de vida y la interacción entre ellos.

En primer lugar, aparece en 2001 Java Enterprise Edition (J2EE 1.3), que se define como una plataforma de programación para el desarrollo de aplicaciones empresariales distribuidas. Era muy complejo y tedioso, con implementaciones lentas y un estilo de programación lleno de tareas repetitivas.

Los objetos de negocio en J2EE se implementan mediante Enterprise JavaBeans. (EJB). Los EJBs requieren de un contenedor EJB ejecutándose en un servidor de aplicaciones.

Para hacer un simple “Hola Mundo” en EJB 1.3 se necesita una cantidad de clases desproporcionada, además de la configuración del fichero XML. Todo esto ocasionó numerosas críticas.

Los orígenes de Spring se remontan al libro escrito por Rod Johnson titulado “Expert One on One J2EE Design and Development”. En él, el autor propone un framework ligero en oposición a J2EE, el cuál derivaría en Spring Framework 1.0 en Marzo de 2004. Dicho framework hacía lo mismo que los EJBs. Sin embargo, los EJBs fueron sustituidos por los beans de Spring, explicados anteriormente [12].

2.3.2. JavaServer Faces (JSF)

Al igual que Spring, JSF es un framework MVC para aplicaciones web en Java [13]. La característica principal de esta tecnología es el uso de un modelo de componentes UI que constituyen la interfaz de de usuario. Un componente puede ser simple, como un campo de texto o un botón, o estar formado por varios componentes, como una tabla.

Para la creación de las vistas emplea un conjunto de páginas con extensión xhtml, denominadas facelets. Estos facelets aportan características de plantillas por medio de etiquetas. La conexión entre estas páginas y las clases Java se realiza mediante Expression Language (EL), un lenguaje de programación destinado fundamentalmente en aplicaciones web en Java para introducir expresiones en páginas web [14].

Para su ejecución, requiere un contenedor de servlets, como Apache Tomcat, además de utilizar beans para representar el estado (modelo) y la lógica de las vistas (controlador). En el modelo los beans se asocian a componentes de la vista, mientras que en el controlador proporcionan métodos propios de éste a ciertos componentes o vistas: conversión, validación, ejecución de acciones, control de flujo, etc [10].

Adicionalmente a los frameworks anteriores, existen otros, denominados front-end, que nos ofrecen las características adecuadas para conseguir interfaces atractivas, "responsive" y que garantizan la compatibilidad entre navegadores. No sólo pueden utilizarse para el producto final, si no que también nos permiten generar prototipos para mostrárselos al cliente al comienzo del proyecto.

A continuación se describen las principales características de algunos de ellos:

2.3.3. Bootstrap

Contiene un conjunto de plantillas CSS y ficheros JavaScript que permiten crear interfaces web [15].

Este marco de trabajo proporciona la creación de diseños bastante intuitivos, mediante la utilización de botones, menús, formularios, etc., además de incluir la integración de jQuery (biblioteca JavaScript) que extiende funcionalidades a la interfaz de usuario, como cuadros de diálogo o alertas.

Básicamente, lo que aporta Bootstrap es la configuración de una serie de clases que permiten reducir de manera considerable el código escrito en una etapa anterior, el cual originaba el mismo resultado. Por ejemplo, para añadir en nuestro desarrollo web una barra de navegación con un menú utilizando CSS, tendríamos que utilizar bastantes atributos hasta alcanzar el diseño deseado. Sin embargo, utilizando Bootstrap se reduce el uso de dichos atributos mediante el empleo de una clase (clase nav, en este caso). De esta manera, se alcanza el mismo objetivo en un período de tiempo inferior.

Bootstrap nació en 2011 gracias a Mark Otto y Jacob Thornton, desarrolladores en Twitter, con el fin de facilitar el trabajo para el resto del equipo, ya que inicialmente existían aspectos frágiles en el desarrollo web, debido al uso de distintas librerías para potenciar las interfaces de usuario.

A pesar de que su aparición se justifica como solución intrínseca de Twitter, los autores se dieron cuenta del enorme potencial de la herramienta. Por ello, el framework fue liberado como código abierto. A raíz de esto, miles de desarrolladores colaboraron en el proyecto de tal manera que Bootstrap adquirió

mayor notoriedad y se convirtió en el framework de diseño habitual para el desarrollo responsive.

Esta idea de configurar formatos y estilos en una clase muestra un gran abanico de posibilidades en el diseño web. Por ello, se ha consolidado como una herramienta necesaria para los desarrolladores.

2.3.4. Pure

Es un framework creado por Yahoo [16]. El código empleado por esta herramienta es ligero y modular. Se basa en la utilización de un normalizador de código CSS denominado Normalize.CSS (pequeño archivo CSS), cuyo principal objetivo es mejorar la consistencia de los navegadores, corrigiendo los errores más comunes entre estos y afinando la usabilidad.

Presenta algunas características similares a Bootstrap, como pueden ser el diseño responsive y la utilización de elementos básicos que componen la interfaz de usuario, tales como botones o menús. La diferencia entre ambos radica en la propuesta por parte de Bootstrap de componentes JavaScript, muy útiles hoy en día.

2.4. Propuesta de solución

A raíz del breve análisis inicial realizado en el capítulo anterior, se produce la primera toma de contacto con el sistema.

El fin de este proyecto es el desarrollo de un sistema informático que permita gestionar la venta de localidades para eventos deportivos a través de Internet. Este sistema viene a cubrir los huecos encontrados en otras aplicaciones del mismo índole, descritas en el [capítulo 2.1.1](#).

Como se indica en dicho capítulo, tras el análisis desarrollado se determina la existencia de dos enfoques: el primero hace referencia a la existencia de sistemas de venta personalizados para clubes deportivos, es decir, cualquier sociedad de personas dedicada a la práctica de un deporte; el segundo está orientado hacia el uso de plataformas que proporcionan servicios que permiten la adquisición de entradas para diferentes actividades. En relación a este último enfoque, cabe destacar también la existencia de aspectos positivos en las aplicaciones web analizadas. Algunas de ellas proporcionan características interesantes, como pueden ser la existencia de menús atractivos o la posibilidad de filtrar las búsquedas por varios criterios. Por el contrario, la mayoría de estas aplicaciones web son demasiado genéricas, además de proporcionar métodos de búsqueda que no siempre obtienen los resultados deseados por el cliente.

Como respuesta a la problemática real observada en la actualidad, se llevará a cabo el desarrollo de una aplicación web accesible para todo el mundo, que utilice medios de pago extendidos y que pueda ser personalizada según las necesidades, teniendo el mismo esquema inicial, todo ello con el objetivo de lograr una mayor satisfacción al cliente.

Se detallarán todas las características de la aplicación, su estructura y su funcionamiento. Para ello, se van a utilizar tecnologías que han sido trabajadas durante los años de formación académica en la Universidad de Jaén.

2.5. Requerimientos del sistema

El procedimiento seguido para el desarrollo de la aplicación web ha sido el ciclo de vida clásico o modelo en cascada, que alberga las siguientes actividades [2]:

- Especificación de requerimientos: consiste en identificar las necesidades de los usuarios finales del software, obteniéndose los objetivos del sistema.
- Análisis del sistema: se definen las tareas específicas que el software ha de poder realizar.
- Diseño del sistema: una vez que se tiene la suficiente información del problema, se determina la estrategia a seguir para resolverlo.
- Implementación: a partir del análisis y diseño de la solución, se desarrolla el código fuente que solucione el problema.
- Pruebas: antes de ser entregado al usuario final, se comprueba que el software desarrollado funciona correctamente y que cumple con los requisitos.

Debido a la importancia de cada una de las etapas anteriores, se han dividido en capítulos. Así, la etapa de especificación de requerimientos y análisis del sistema se detallan en este capítulo ([capítulo 2](#)), el diseño del sistema se expone en el [capítulo 3](#) y la fase de implementación y pruebas es analizada en el [capítulo 4](#).

La primera etapa en el ciclo consiste en determinar los objetivos del proyecto, las propiedades que debe satisfacer y las limitaciones existentes. Esta etapa es considerada de vital importancia para que la realización de un proyecto software sea capaz de cumplir el propósito.

Una vez definidos el propósito y los objetivos del proyecto en el [capítulo 1](#), el siguiente paso consiste en especificar los requerimientos del mismo. Estos son un conjunto de condiciones o capacidades exigidas por el usuario para solucionar un problema o alcanzar un objetivo [17].

2.5.1. Requerimientos funcionales

Los requerimientos funcionales describen lo que hace un sistema o lo que se espera que haga; es decir, su funcionalidad. Son declaraciones sobre servicios que el sistema debe ofrecer, cómo el sistema debe responder a determinadas entradas y cómo el sistema debería comportarse en situaciones particulares. El problema se describe desde el punto de vista del usuario y el requisito se plantea desde el punto de vista del sistema [17].

Las funcionalidades esperadas de la aplicación pueden dividirse en tres: las que se corresponden con el administrador del sistema (ROLE_ADMIN), las relacionadas con los clientes potenciales (ROLE_USER) y las comunes a ambos. Para distinguir cada uno de los requisitos funcionales se ha utilizado la notación RF-Número_del_requisito: Nombre_del_requisito.

En primer lugar, se declaran los requerimientos funcionales comunes a ambos tipos de usuario:

1) RF-01: Identificación de un usuario

El usuario podrá identificarse, introduciendo un nombre de usuario y una contraseña. De esta forma, podrá acceder a las distintas funcionalidades que ofrece la aplicación, según el rol del usuario autenticado.

2) RF-02: Salir de la aplicación

El usuario podrá abandonar la sesión iniciada en cualquier momento, regresando a la página principal de la aplicación.

3) RF-03: Registro de usuario

Permite el registro de un nuevo usuario en el sistema, introduciendo los siguientes campos: nombre, correo, ciudad, nombre de usuario, clave, y edad. Por defecto, el usuario registrado tendrá ROL_USER.

Una vez finalizados los requerimientos comunes, se definen los que son competencia del administrador:

4) Gestión de usuarios

Los requerimientos funcionales relacionados con la gestión de usuarios son:

- **RF-04: Alta de usuario**

El sistema ofrece al administrador la posibilidad de dar de alta un nuevo usuario en el sistema, introduciendo los siguientes campos: nombre, correo, ciudad, nombre de usuario, clave, y rol.

- **RF-05: Editar usuario**

El sistema ofrece al administrador la posibilidad de editar: nombre, correo, ciudad, nombre de usuario, clave y rol de un usuario.

- **RF-06: Eliminar usuario**

Consiste en eliminar toda la información referente a un usuario.

- **RF-07: Listado de usuarios**

Muestra un listado en el que se visualizan los usuarios registrados en el sistema. Para cada uno de ellos se mostrará: identificador de usuario, nombre, correo, ciudad, nombre de usuario, clave, edad y rol.

5) Gestión de deportes

Los requerimientos funcionales relacionados con la gestión de deportes son:

- **RF-08: Alta de deporte**

Permite el registro de un nuevo deporte en el sistema, introduciendo los siguientes campos: nombre, denominación del participante y denominación del evento.

- **RF-09: Editar deporte**

El sistema ofrece al administrador la posibilidad de editar todos los campos asociados a un deporte, salvo el identificador del mismo, generado de manera automática.

- **RF-10: Eliminar deporte**

Consiste en eliminar toda la información referente a un deporte.

- **RF-11: Listado de deportes**

Muestra un listado en el que se visualizan los deportes registrados en el sistema. Para cada uno de ellos se mostrará: identificador de deporte, nombre, denominación del participante y denominación del evento.

6) Gestión de competiciones

Los requerimientos funcionales relacionados con la gestión de competiciones son:

- **RF-12: Alta de competición**

Permite el registro de una nueva competición en el sistema, introduciendo el nombre y el deporte al que pertenece.

- **RF-13: Editar competición**

El sistema ofrece al administrador la posibilidad de editar el nombre y el deporte de la competición.

- **RF-14: Eliminar competición**

Consiste en eliminar toda la información referente a una competición.

- **RF-15: Listado de competiciones**

Muestra un listado en el que se visualizan las competiciones registradas en el sistema. Para cada una de ellas se mostrará: identificador de competición, nombre y deporte.

7) Gestión de participantes

Los requerimientos funcionales relacionados con la gestión de participantes son:

- **RF-16: Alta de participante**

Permite el registro de un nuevo participante en el sistema, introduciendo el nombre y deporte al que pertenece.

- **RF-17: Editar participante**

El sistema ofrece al administrador la posibilidad de editar el nombre y el deporte en el que interviene el participante.

- **RF-18: Eliminar participante**

Consiste en eliminar toda la información referente a un participante.

- **RF-19: Listado de participantes**

Muestra un listado en el que se visualizan los participantes registrados en el sistema. Para cada uno de ellos se mostrará: identificador de participante, nombre y deporte.

8) Gestión de eventos

Los requerimientos funcionales relacionados con la gestión de eventos son:

- **RF-20: Alta de evento**

Permite el registro de un nuevo evento en el sistema, introduciendo los siguientes campos: nombre, fecha, hora, lugar, ciudad, país, número de entradas, deporte, competición y participantes.

- **RF-21: Editar evento**

El sistema ofrece al administrador la posibilidad de editar: nombre, fecha, hora, lugar, ciudad, país, deporte, competición y participantes de un evento.

- **RF-22: Eliminar evento**

Consiste en eliminar toda la información referente a un evento.

- **RF-23: Listado de eventos**

Muestra un listado en el que se visualizan los eventos registrados en el sistema. Para cada uno de ellos se mostrará: identificador del evento, nombre, fecha, hora, ciudad y país.

- **RF-24: Visualizar evento**

Permite la visualización de la información relacionada con un evento, tal como: identificador del evento, nombre, fecha, hora, lugar, ciudad, país, número de entradas, vendida (indica si las entradas del evento están puestas a la venta), deporte, competición y participantes.

9) Gestión de entradas

Previamente señalamos que el campo denominado 'comprada' indica si una entrada está comprada. De este modo, se procede a detallar los requerimientos funcionales relacionados con la gestión de entradas:

- **RF-25: Alta de entrada**

Permite el registro de una nueva entrada en el sistema, introduciendo los siguientes campos: descripción, precio y evento al que pertenece.

- **RF-26: Editar entrada**

El sistema ofrece al administrador la posibilidad de editar el precio, la descripción y el evento de una entrada.

- **RF-27: Eliminar entrada**

Consiste en eliminar toda la información referente a una entrada.

- **RF-28: Listado de entradas**

Muestra un listado en el que se visualizan las entradas registradas en el sistema. Para cada uno de ellas se mostrará: identificador de la entrada, descripción, precio, compra y evento.

- **RF-29: Puesta en venta**

El sistema permite al administrador la opción de habilitar al cliente la visualización de las entradas disponibles para cada evento.

10) Gestión de compras

Los requerimientos funcionales relacionados con la gestión de compras son:

- **RF-30: Visualizar compra**

Permite la visualización de la información relacionada con una compra, tal como: identificador de compra, precio total, usuario que realizó la operación, entradas asociadas, fecha de compra y fecha de pago.

- **RF-31: Listado de compras**

Muestra un listado en el que se visualizan las compras registradas en el sistema. Para cada uno de ellas se mostrará: identificador de la compra, usuario que realizó la operación, fecha de compra y fecha de pago.

A continuación, se detallan los requerimientos funcionales atribuidos al cliente:

11)Gestión de datos personales

Los requerimientos funcionales relacionados con la gestión de datos personales son:

- **RF-32: Editar cliente**

El sistema habilita al usuario la opción de editar el nombre, correo, ciudad y clave.

- **RF-33: Eliminar cliente**

El sistema ofrece al cliente la posibilidad de eliminar sus datos personales.

- **RF-34: Visualizar datos cliente**

Permite la visualización de los datos personales de un usuario, tales como: identificador de usuario, nombre, correo, ciudad, nombre de usuario y clave.

12)RF-35: Buscar eventos

Dado un criterio de búsqueda introducido o seleccionado por el cliente, el sistema mostrará un listado en el que se pueden visualizar todos los eventos existentes para dicho criterio.

13)RF-36: Ver compras realizadas

El cliente tiene la posibilidad de visualizar un listado en el que se contemplan las compras realizadas por éste.

14)RF-37: Comprar entradas

El sistema proporciona al cliente la posibilidad de adquirir entradas para un determinado evento.

15)RF-38: Pagar entradas

El sistema permite al cliente el abono del precio de las entradas seleccionadas para un evento concreto a través de PayPal.

16)RF-39: Imprimir entradas

El sistema ofrece al cliente la posibilidad de imprimir las entradas adquiridas pertenecientes a una compra determinada.

2.5.2. Requerimientos no funcionales

Los requerimientos no funcionales describen aspectos del sistema que están relacionados con el grado de cumplimiento de los requisitos funcionales.

No describen funcionalidad si no la calidad que se espera de los servicios que va a ofrecer el sistema. De ahí, que se denominen también requerimientos de calidad.

Habitualmente están relacionados con características que puedan limitar el sistema, como puede ser la robustez, rendimiento, usabilidad, plataforma sobre la que se va a ejecutar o el equipo informático a utilizar. Al igual que los requisitos funcionales, el grado de importancia de los requerimientos no funcionales es elevado, ya que pueden llegar a ser cruciales para el sistema.

Inicialmente podemos deducir que no existen restricciones concernientes a la organización. Con respecto al equipo, la ejecución del sistema se llevará a cabo en cualquier sistema operativo que disponga de máquina virtual Java, mientras que se hará uso de una base de datos relacional, sobre la cual se podrán realizar futuras consultas. Además, deberá poderse acceder a la aplicación empleando cualquier navegador web.

Por lo tanto, se obtendrán y analizarán los requerimientos no funcionales referentes a la interfaz.

2.5.2.1. Requerimientos de la interfaz

Los requerimientos de la interfaz gráfica están fuertemente vinculados a la usabilidad y sus principios. Para que un sistema interactivo cumpla sus objetivos tiene que ser usable y accesible a la mayor parte de la población humana [18].

La usabilidad es la medida en la que un producto se puede usar por determinados usuarios para conseguir unos objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso dado [18]. La efectividad es la capacidad que permite a los usuarios alcanzar los objetivos especificados. La eficiencia hace referencia a los recursos empleados para lograr esos objetivos, como

por ejemplo el tiempo. La satisfacción se define como una actitud positiva del usuario hacia el producto.

Los principios básicos de la usabilidad definen los requerimientos no funcionales que deberá satisfacer la interfaz, y son los siguientes [18]:

- Facilidad de aprendizaje: el tiempo requerido desde el no conocimiento de una aplicación hasta su uso productivo debe ser mínimo. Para que un sistema sea fácil de aprender debe ser:
 - Sintetizable: el usuario debe poder percibir el efecto que acciones anteriores han ocasionado sobre el sistema.
 - Familiar: los elementos de la interfaz deben resultar conocidos y familiares para el usuario.
- Flexibilidad: hace referencia a la diversidad de maneras en que el usuario y el sistema pueden intercambiar información. El usuario es quien decide cuando iniciar y finalizar una interacción. El sistema localiza los errores y se los muestra al usuario.
- Consistencia: un sistema es consistente si todos los mecanismos que se utilizan son siempre usados de la misma manera. Para realizar una tarea determinada siempre se seguirán los mismos pasos.
- Robustez: es el grado en que el sistema es capaz de tolerar fallos. Debe permitir al usuario conseguir sus objetivos sin problemas. Para que el sistema sea robusto, es fundamental controlar la introducción de información incorrecta por parte del usuario, informando a éste acerca del error y cómo podría solucionarlo.
- Recuperabilidad: el sistema debe permitir al usuario corregir una acción una vez que ésta ha sido reconocida como errónea.
- Tiempo de respuesta: es el tiempo que necesita el sistema para reflejar los cambios realizados por el usuario. Tiene que ser soportable para éste.

2.6. Planificación de tareas

En este apartado se llevará a cabo la planificación temporal de la secuencia de ejecución de las tareas del proyecto.

2.6.1. Estructura del plan de trabajo

A continuación se presenta el esquema del plan de trabajo que será necesario llevar a cabo para alcanzar los [objetivos del proyecto](#):

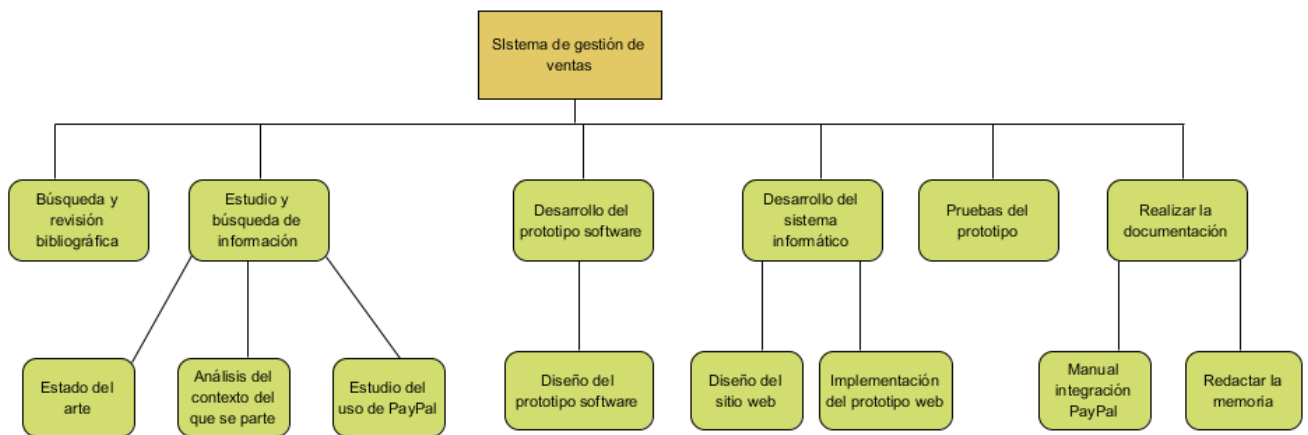


Ilustración 6: Estructura del plan de trabajo

Las tareas a las que se concede mayor relevancia son el estudio del uso de PayPal y el diseño del sitio web. Una parte importante de la aplicación es la implantación de un sistema de pago, por lo tanto se realizará un análisis sobre el uso de Paypal en el comercio electrónico a través de Internet. Por otro lado, el diseño del sitio web sigue las directrices del patrón de diseño Modelo-Vista-Controlador, proporcionándonos una arquitectura de separación en tres capas que facilita el mantenimiento y la escalabilidad de nuestra aplicación.

Se prevé que la tarea relacionada con la implementación del prototipo web sea la de mayor duración. Aunque se conocen las tecnologías que se utilizarán, no puede decirse que la curva de aprendizaje sea suave, debido a que algunos conceptos son novedosos.

Por otro lado, se definirán algunos hitos. Los hitos son actividades de duración nula que indican fechas o puntos dentro de un proyecto que se han de tener en especial consideración [19].

2.6.2. Estimación de tiempos

En este subapartado se estimará el tiempo necesario para llevar a cabo cada una de las tareas del punto anterior, así como la persona encargada de la misma:

- Búsqueda y revisión bibliográfica ----- 10 días (Analista/Diseñador)
- Estudio y búsqueda de información
 - Estado del arte ----- 5 días (Analista/Diseñador)
 - Análisis del contexto del que se parte ----- 5 días (Analista/Diseñador)
 - Estudio del uso de Paypal ----- 10 días (Analista/Diseñador)

HITO 1: Estudio y búsqueda de información

- Desarrollo del prototipo software
 - Diseño del prototipo software ----- 5 días (Analista/Diseñador)
- Desarrollo del sistema informático
 - Diseño del sitio web ----- 10 días (Analista/Diseñador)
 - Implementación del prototipo web ----- 50 días (Programador)
- Pruebas del prototipo ----- 5 días (Programador)

HITO 2: Prototipo

- Realizar la documentación
 - Manual de integración de PayPal ----- 5 días (Analista/Diseñador)
 - Completar la memoria ----- 40 días (Analista/Diseñador)

HITO 3: Documentación

Tiempo total ----- 120 días

Se estima que el tiempo de realización del proyecto es de 120 días (6 meses), excediéndose así los cuatro meses del cuatrimestre. La suma de la duración de

cada una de las tareas específicas no obtiene como resultado dicho número, debido a que algunas tareas han sido ejecutadas en paralelo con otras.

2.6.3. Diagrama de Gantt

Finalmente, se muestra la tabla de actividades y diagrama de Gantt correspondientes a la planificación del proyecto que se va a llevar a cabo:

- **Tabla de actividades**

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Búsqueda y revisión bibliográfica	10 días	lun 05/01/15	vie 16/01/15	
2	Estado del arte	5 días	lun 19/01/15	vie 23/01/15	1
3	Análisis del contexto del que se parte	5 días	lun 19/01/15	vie 23/01/15	1
4	Estudio del uso de PayPal	10 días	lun 19/01/15	vie 30/01/15	1
5	Hito 1	0 días	vie 30/01/15	vie 30/01/15	4
6	Diseño del prototipo software	5 días	lun 02/02/15	vie 06/02/15	5
7	Diseño del sitio web	10 días	lun 09/02/15	vie 20/02/15	6
8	Implementación del prototipo web	50 días	lun 09/02/15	vie 17/04/15	6
9	Pruebas del prototipo	5 días	lun 20/04/15	vie 24/04/15	7;8
10	Hito 2	0 días	vie 24/04/15	vie 24/04/15	9
11	Manual de integración de PayPal	5 días	lun 27/04/15	vie 01/05/15	10
12	Completar la memoria	40 días	lun 27/04/15	vie 19/06/15	10
13	Hito 3	0 días	vie 19/06/15	vie 19/06/15	12

Ilustración 7: Actividades

Cabe destacar que no se han establecido días de descanso especiales. El período laboral se desarrolla de lunes a viernes (5 días), considerándose jornadas de trabajo de 8 horas, mientras que el no laborable abarca los fines de semana (2 días).

- **Diagrama de Gantt**

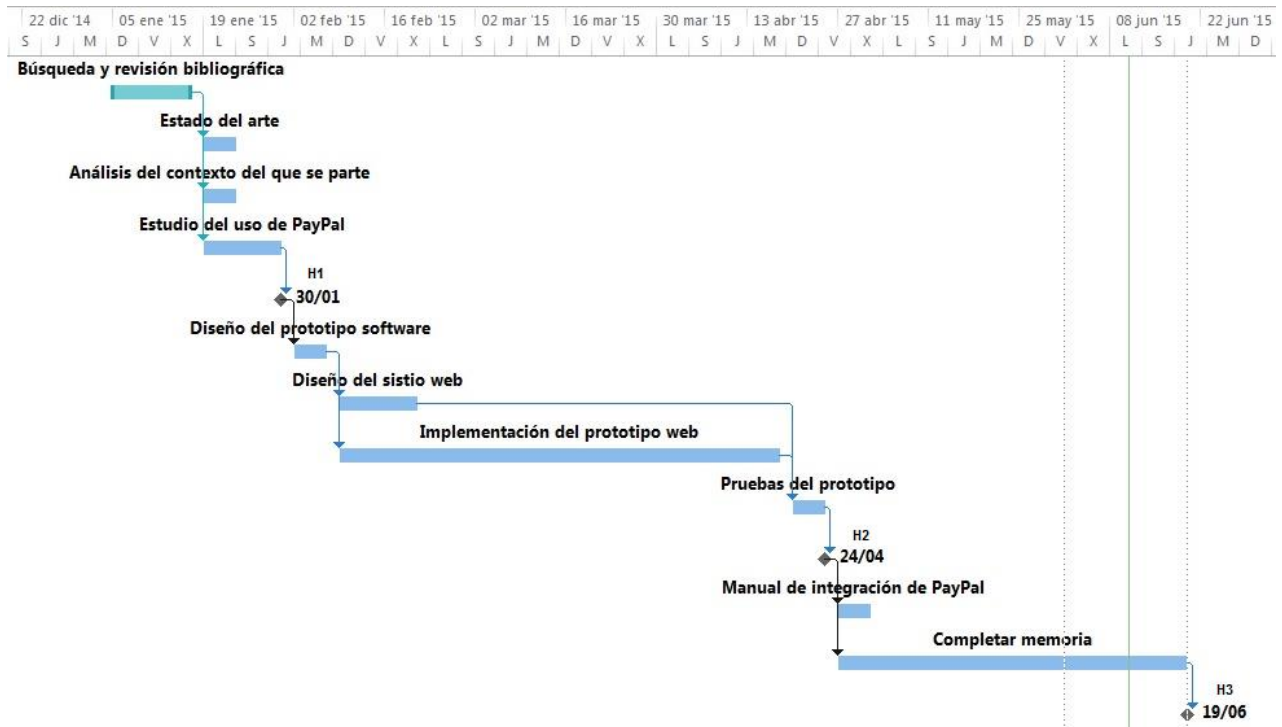


Ilustración 8: Diagrama de Gantt

Como puede observarse, se han determinado 3 hitos dentro del proyecto, como señal de que hemos llegado a un punto importante del mismo:

- El primero, cuya fecha es el 30 de Enero de 2015, es la meta inicial. Hace referencia a la finalización del estudio y búsqueda de información. Una vez realizadas las cuatro primeras tareas, se dispone de un amplio catálogo de información y por lo tanto se puede iniciar el desarrollo de la solución.
- El segundo, establecido el día 24 de Abril de 2015 determina la aceptación final del sistema software. En él se incluyen las tareas relacionadas con el diseño del prototipo y del sitio web, así como la implementación y las pruebas de dicho prototipo. Este hito marca el comienzo del período de documentación de la aplicación desarrollada.
- El último hito, fijado el 19 de Junio de 2015, representa el hito final del proyecto y abarca las tareas relacionadas con la documentación, finalizando de esta manera el desarrollo del sistema informático.

2.7. Estudio de viabilidad

En este apartado se abordará un análisis económico, revisando los costes de desarrollo necesarios para llevar a cabo este proyecto.

Al ser un proyecto de carácter informático, el requerimiento de producción principal para el producto final de la aplicación es la disposición de un programador y un analista/diseñador a jornada completa durante 6 meses, de modo que la colaboración y el entendimiento entre ambos será determinante para conseguir la realización de un software que atienda a los objetivos propuestos.

A continuación se muestra una tabla en la que se recogen los gastos del personal citado en relación al número de horas de trabajo productivas:

Concepto	Horas/día	Días de trabajo	Nómina mensual	Coste/hora	Coste total
Analista/diseñador	8	90	1497,48 €	9,35925 €	6738,66 €
Programador	8	55	1328,78 €	8,304875 €	3654,15 €
Total costes				17,664125 €	10392,81 €

Tabla I: Gastos personal

Para establecer los sueldos del personal citado, se ha usado el Convenio colectivo estatal de las empresas de seguridad, publicado en el BOE [20].

En el proceso de fabricación del producto software es esencial el empleo del material adecuado. Por un lado, se encuentran los equipos informáticos utilizados por el programador y el analista. El tiempo de vida estimado de cada uno de estos equipos es de 5 años. Teniendo en cuenta que el precio de los equipos es de 600 €/unidad y que se manejaron durante 6 meses:

Concepto	Cantidad	Coste unidad	Tiempo de vida estimado	Tiempo utilizado	Coste/año	Coste tiempo utilizado
Equipo informático	2	600 €	5 años	6 meses	240 €	120 €
Total costes					240 €	120 €

Tabla II: Equipos informáticos

Además, para el funcionamiento de los equipos informáticos se incorporan gastos de periféricos:

Concepto	Cantidad	Coste unidad	Tiempo de vida estimado	Tiempo utilizado	Coste/año	Coste tiempo utilizado
Ratón inalámbrico	2	15 €	5 años	6 meses	6 €	1 €
Impresora multifunción	1	139 €	5 años	6 meses	27,8 €	4,63 €
Total costes					33,8 €	5,63 €

Tabla III: Periféricos

Siguiendo en esta línea, el importe del ADSL (Movistar Nueva Fibra Óptica 300 Mb) asciende a 55,4 €/mes, el cual incluye de manera gratuita llamadas a fijos nacionales, 500 min/mes de fijo a móviles de cualquier operador los fines de semana, 50 min/mes de fijo a móviles de cualquier operador de lunes a viernes y el equipamiento de un router wifi 802.11n.

Por otro lado, hay que considerar el coste de las licencias de software utilizadas. A pesar de haberse utilizado software de apoyo libre y gratuito, también se hallan los costes relacionados con algunas licencias comerciales.

Cuando se adquiere una licencia de software dada por el fabricante del mismo, puede utilizarse ese software solamente bajo los términos especificados por esa licencia. Dichos términos indican el número de usuarios y el número de ordenadores que pueden ejecutar el software, en ocasiones uno en ambos casos. Así, existen usuarios que no han leído detenidamente la licencia y lo utilizan en varios ordenadores, arriesgándose a ser denunciados.

Cabe destacar que en la obtención de las licencias no se han tenido en cuenta las condiciones anteriores, por lo que el importe abonado en la compra de estas ha sido el siguiente:

Concepto	Licencia	Versión	Coste/año	Coste tiempo utilizado
Ofimática	Microsoft Office	Professional Plus 2010	209 €	104,5 €
Ofimática	Microsoft Project	Professional 2013	549,6 €	274,8 €
Sistema operativo	Windows	8.1	119 €	119 €
Diseño	Visual Paradigm	11.2	271 €	271 €
IDE	NetBeans	8.0.2	Free	Free
Framework y bibliotecas externas	Spring	4.0.1	Free	Free
Apache	Tomcat	8.0.9	Free	Free
Oracle	Java SE Development Kit	8	Free	Free
Diseño	Bootstrap	3.3.1	Free	Free
Navegador	Google Chrome	43.0.2357.81 m	Free	Free
Total costes			1148,6 €	769,3€

Tabla IV: Licencias software

Los campos marcados con *free* hacen referencia a la utilización de software libre, lo que permite una reducción de los costes.

A todos los gastos anteriores se les incluye un margen de maniobra del 5%, con el fin de equilibrar la economía en caso de existir un exceso en los gastos o algún desembolso extraordinario.

Si fusionamos todos los gastos anteriores, se obtiene la tabla final de costes estimados en el desarrollo del proyecto:

Concepto	Total
Personal	10392,81 €
Equipos informáticos	120 €
Periféricos	5,63 €
ADSL	332,4 €
Licencias	769,3€
Margen de maniobra	581,007 €
Total	12201,147€

Tabla V: Costes finales

En resumen, se estima un coste total de desarrollo del proyecto durante los 6 meses de duración de 12201,147 €.

2.8. Casos de uso

Los casos de uso especifican el comportamiento de un sistema software. Definen lo que hace el sistema desde el punto de vista del usuario (actor), es decir, tanto lo que hace el actor como lo que hace el sistema cuando interactúa con él por el uso de una determinada funcionalidad. Es esencial que dichos casos de uso comprendan los requerimientos funcionales anteriormente especificados [2].

Son documentados con lenguaje natural. Gráficamente, se representan mediante un óvalo y un texto significativo en su interior.

Los casos de uso constan de los siguientes elementos:

- Nombre del caso de uso: tiene que ser único e unívoco.
- Actor primario: es el que inicia o requiere del caso de uso.
- Sistema al que pertenece el caso de uso.
- Participantes: hace referencia al conjunto de actores que participan en un caso de uso, tanto primarios (definidos anteriormente) como secundarios (aquellos que simplemente intervienen en el caso de uso).
- Condición previa: requisito indispensable para la ejecución del caso de uso.
- Operaciones básicas del escenario principal: secuencia numerada de los pasos que sigue el actor para interactuar con el sistema.
- Alternativas: lista de errores o excepciones que tienen lugar en el curso normal del caso de uso. No tienen sentido fuera del caso de uso en el que ocurren.

Se pueden emplear varios tipos de relaciones a la hora de especificar un caso de uso:

- Comunicación: es el tipo de relación más básico. Vincula a un actor con un caso de uso. Esta relación se representa mediante una flecha continua
- Inclusión: permite el enriquecimiento de un caso de uso a raíz de otro. Básicamente, cuando se relacionan dos casos de uso usando este tipo

de relación, se está diciendo que el primero incluye al segundo. Por lo tanto, el segundo es parte fundamental del primero, ya que sin éste, el primero no podría cumplir su objetivo. Esta relación se representa mediante una flecha discontinua acompañada del estereotipo `<<include>>`.

- Extensión: al contrario que la relación anterior, se trata de una funcionalidad opcional. Nuevamente se establece una relación entre dos casos de uso pero en esta ocasión, no es imprescindible que ocurra el segundo caso de uso para que el primero cumpla sus objetivos. Sin embargo, cuando lo hace presenta un valor adicional al objetivo inicial del primer caso de uso. Esta relación se representa mediante una flecha discontinua acompañada del estereotipo `<<extend>>`.
- Herencia: facilita la especialización de un caso de uso en otro u otros, obteniendo uno o varios subcasos de uso. El subcaso hereda el comportamiento y las relaciones del supercaso de uso, además de poder redefinir el comportamiento de éste. Este tipo de relación se representa mediante una línea finalizada en un triángulo orientado desde el caso de uso especializado hacia el caso de uso general.

Una vez definidos los elementos y las relaciones fundamentales de un caso de uso, es necesario determinar el diagrama de casos de uso del sistema, que incluye los actores que participan en cada uno de los casos de uso:

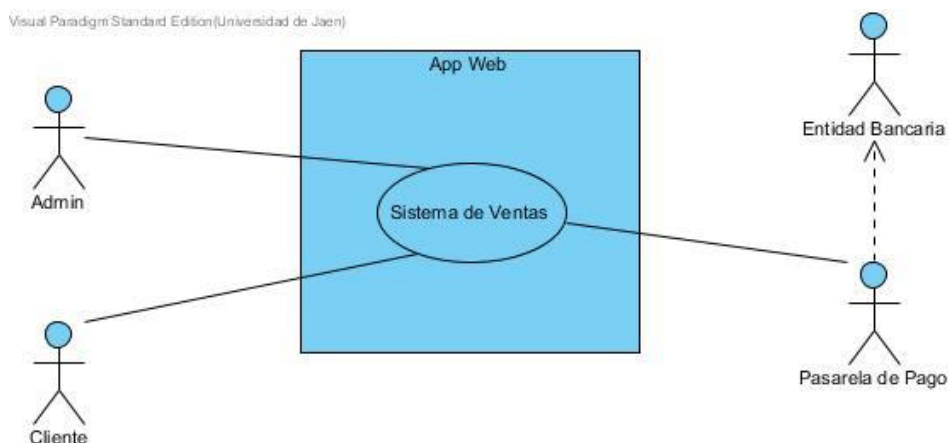


Ilustración 9: Sistema

Como puede observarse, los actores pertenecientes al sistema son:

- Administrador: se define como el usuario responsable de la aplicación, encargado de controlar y optimizar los servicios de la misma.
- Cliente: hace referencia al usuario que accede a los servicios brindados por la aplicación para la adquisición de entradas.
- Pasarela de pago: se corresponde con la entidad que permite enviar y recibir pagos a través de Internet.
- Entidad bancaria: se encarga de la prestación de servicios que permiten financiar las operaciones de pago realizadas.

El primer paso para realizar el modelo de casos de uso es la obtención de los diferentes diagramas de casos de uso que reflejan el funcionamiento del sistema, al que denominaremos “Sistema de Ventas”. El primero de ellos, denominado diagrama frontera, describe completamente la funcionalidad de sistema. Incluye todos los casos de uso genéricos del sistema, que podrán ser detallados en mayor profundidad en nuevos diagramas de casos de uso si no fueran lo suficientemente exactos.

A continuación se puede observar el diagrama frontera del sistema:

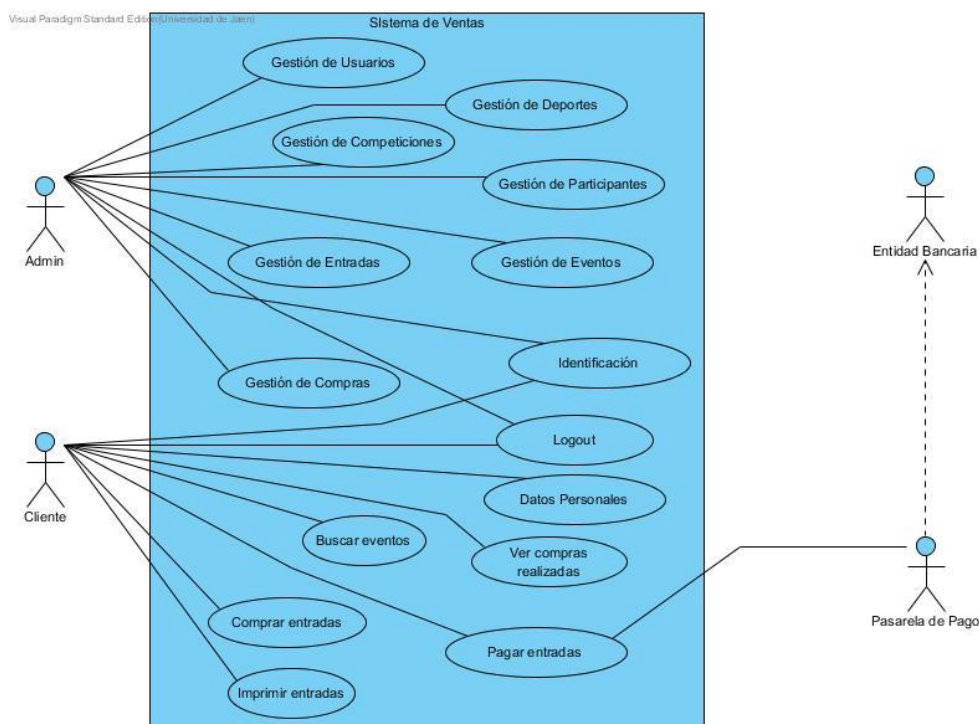


Ilustración 10: Diagrama Frontera

En este caso, existen varios casos de uso que deben ser analizados con mayor precisión. Por lo tanto, se procede a ver con mayor detalle dichos casos de uso:

- **Diagrama de casos de uso: Identificación**

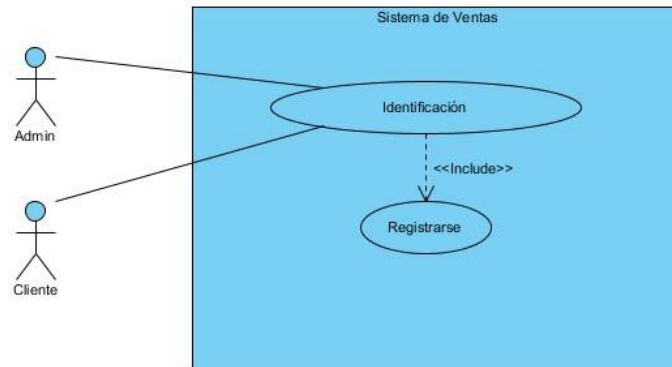


Ilustración 11: Identificación

Los actores que intervienen en el caso de uso son el administrador y el cliente. Un requisito necesario para acceder a la plataforma es estar registrado en la misma.

- **Diagrama de casos de uso: Gestión de usuarios**

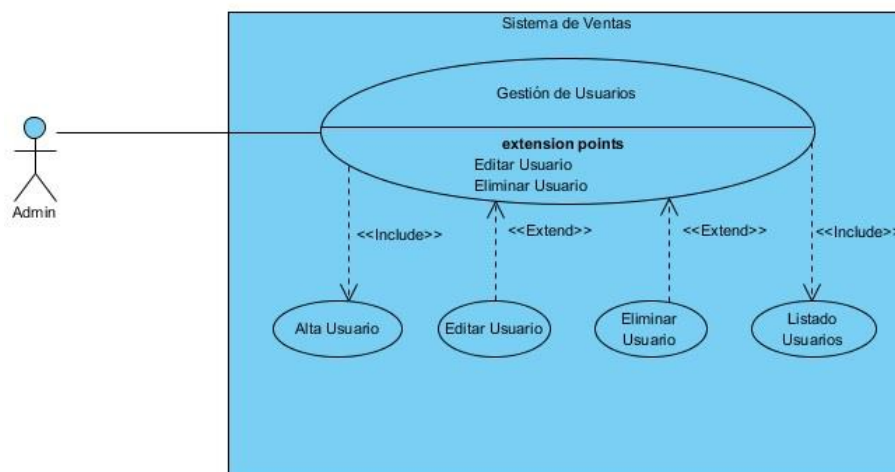


Ilustración 12: Gestión de Usuarios

El administrador es el encargado de realizar la gestión de los usuarios del sistema. Para ello, es imprescindible que puedan darse de alta nuevos usuarios y disponer de un listado que permita la administración de éstos. Si lo desea, también tiene la posibilidad de editar o eliminar un usuario.

- **Diagrama de casos de uso: Gestión de deportes**

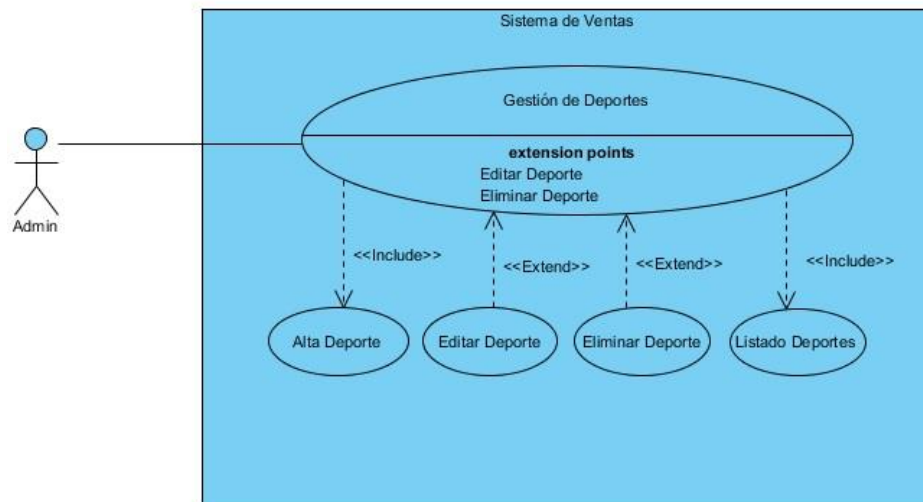


Ilustración 13: Gestión de deportes

El administrador es el encargado de realizar la gestión de los deportes del sistema. Para ello, es imprescindible que puedan darse de alta nuevos deportes y disponer de un listado que permita la administración de éstos. Si lo desea, también tiene la posibilidad de editar o eliminar un deporte.

- **Diagrama de casos de uso: Gestión de competiciones**

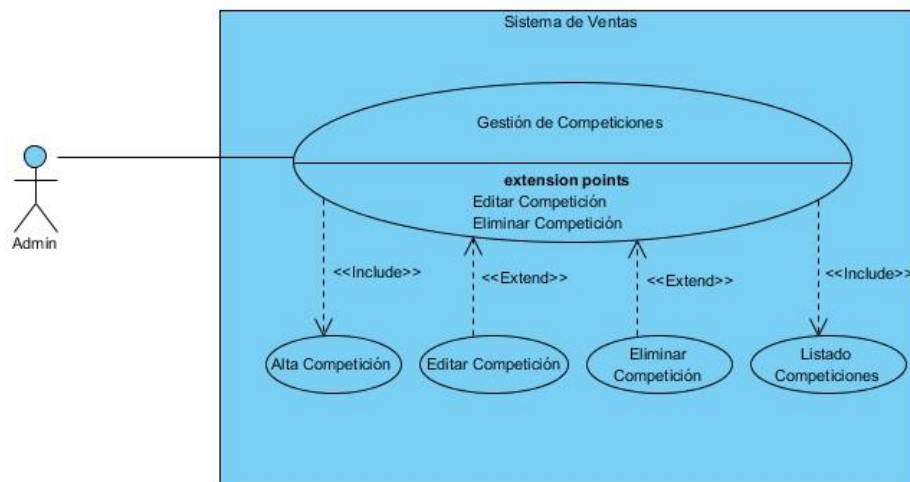


Ilustración 14: Gestión de competiciones

El administrador es el encargado de realizar la gestión de las competiciones del sistema. Para ello, es imprescindible que puedan darse de alta nuevas competiciones y disponer de un listado que permita la administración de éstas. Si lo desea, también tiene la posibilidad de editar o eliminar una competición.

- **Diagrama de casos de uso: Gestión de participantes**

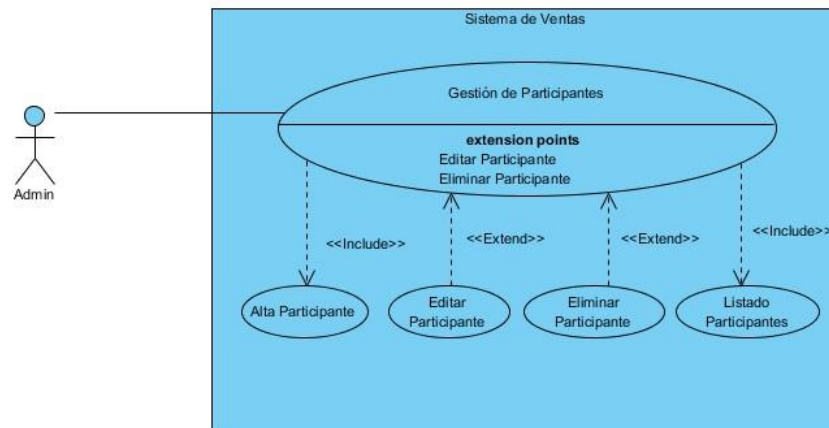


Ilustración 15: Gestión de participantes

El administrador es el encargado de realizar la gestión de los participantes del sistema. Para ello, es imprescindible que puedan darse de alta nuevos participantes y disponer de un listado que permita la administración de éstos. Si lo desea, también tiene la posibilidad de editar o eliminar un participante.

- **Diagrama de casos de uso: Gestión de eventos**

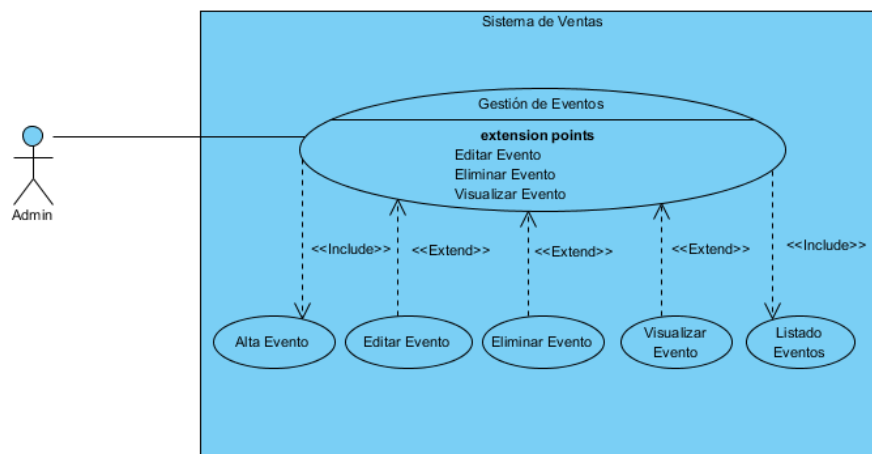


Ilustración 16: Gestión de eventos

El administrador es el encargado de realizar la gestión de los eventos del sistema. Para ello, es imprescindible que puedan darse de alta nuevos eventos y disponer de un listado que permita la administración de éstos. Si lo desea, también tiene la posibilidad de editar, eliminar o visualizar evento.

- **Diagrama de casos de uso: Gestión de entradas**

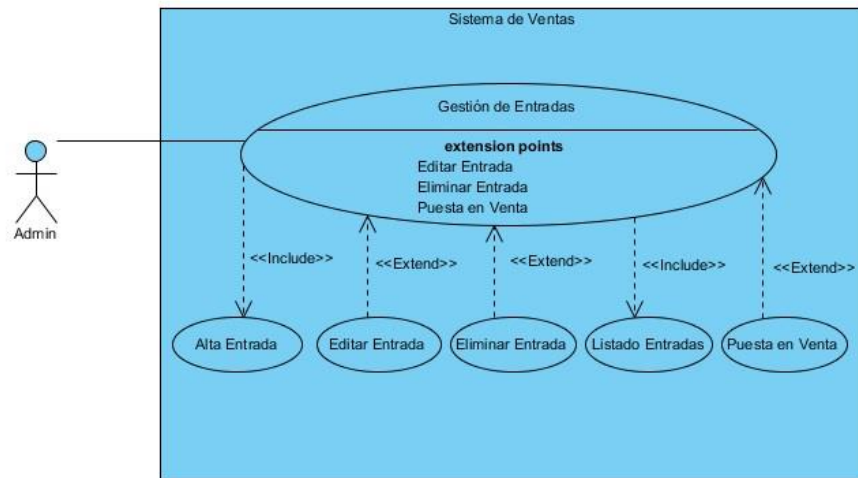


Ilustración 17: Gestión de entradas

El administrador es el encargado de realizar la gestión de las entradas del sistema. Para ello, es imprescindible que puedan darse de alta nuevas entradas y disponer de un listado que permita la administración de éstas. Si lo desea, también tiene la posibilidad de editar o eliminar una entrada, además de ponerlas a la venta.

- **Diagrama de casos de uso: Gestión de compras**

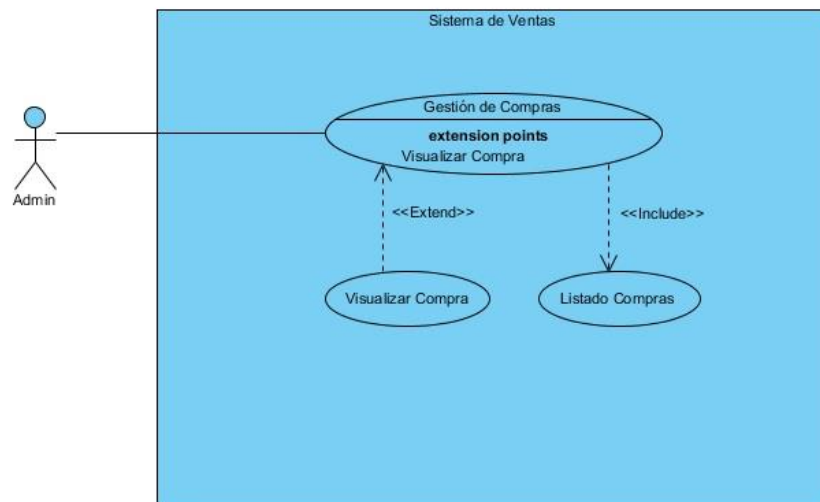


Ilustración 18: Gestión de compras

El administrador es el encargado de realizar la gestión de las compras del sistema. Para ello, es imprescindible disponer de un listado que permita la administración de éstas. Si lo desea, también tiene la posibilidad de visualizar una compra concreta.

- **Diagrama de caso de uso: Datos personales**

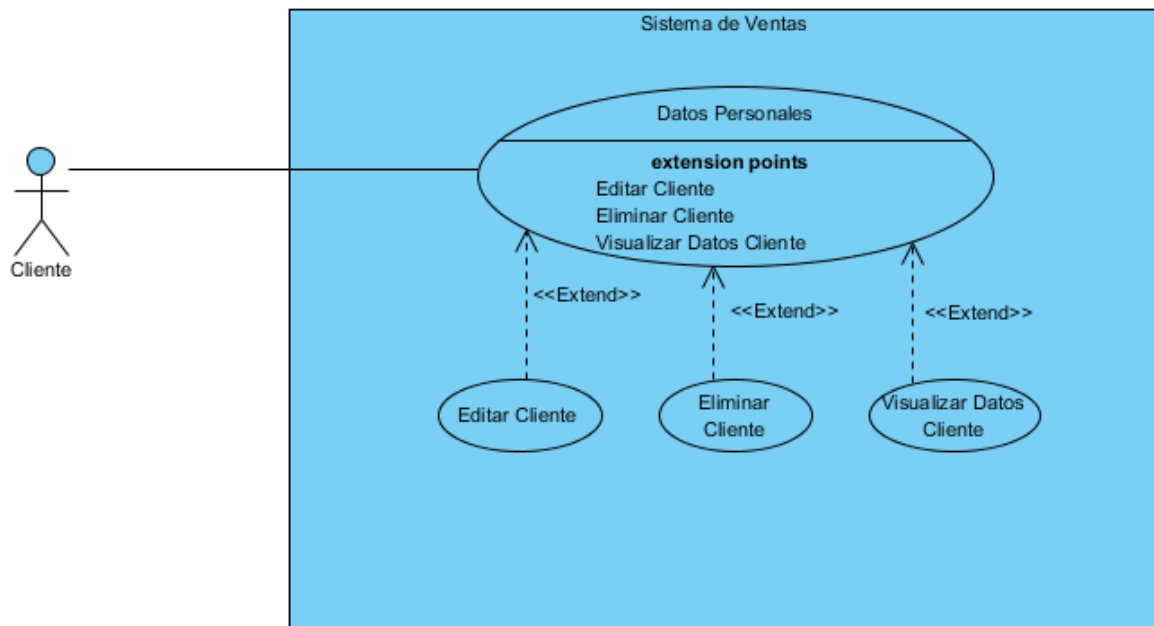


Ilustración 19: Gestión de datos personales

Una vez identificado, el cliente tiene la posibilidad de editar, eliminar o visualizar sus datos.

2.8.1. Especificación de casos de uso

Como puede observarse, tanto en el diagrama frontera como en los diagramas de casos de uso previos existen multitud de casos de uso diferentes. Para especificarlos no es suficiente con los diagramas precedentes, por lo que seguidamente se describen de forma detallada los casos de uso anteriores que aportan una mayor relevancia sobre el sistema, debido al gran número de estos y a la funcionalidad similar proporcionada entre algunos de ellos:

- **Caso de uso 1: Identificación**

Caso de uso	Identificación
Actor primario	Admin, Cliente
Sistema	Sistema de Ventas
Participantes	Admin, Cliente, Sistema
Condición previa	El usuario tiene que estar registrado en el sistema
Operaciones básicas	
1	Acceso a la aplicación por parte del usuario
2	Introducción de credenciales
3	El sistema recibe los datos del usuario
4	El usuario accede al contenido de la aplicación
Alternativas	
3.A	¿Son correctos los datos?
3.A.1	Si sí, continuar
3.A.2	Si no, el sistema mostrará un mensaje de error y volver a 2

Tabla VI: Caso de uso "Identificación"

- **Caso de uso 2: Registrarse**

Caso de uso	Registrarse
Actor primario	Cliente
Sistema	Sistema de Ventas
Participantes	Cliente, Sistema
Condición previa	Para acceder a la aplicación se necesita disponer de conexión a Internet
Operaciones básicas	
1	Acceso a la aplicación por parte del cliente
2	El usuario accede al formulario de registro
3	El usuario rellena los campos habilitados para darse de alta
4	El sistema recibe los datos del usuario
Alternativas	
4.A	¿Son correctos los datos?
4.A.1	Si sí, el sistema mostrará un mensaje de confirmación y finalizar
4.A.2	Si no, el sistema mostrará un mensaje de error y volver a 3

Tabla VII: Caso de uso "Registrarse"

- **Caso de uso 3: Listado de usuarios**

Caso de uso	Listado Usuarios
Actor primario	Admin
Sistema	Sistema de Ventas
Participantes	Admin, Sistema
Condición previa	El administrador tiene que estar registrado en el sistema
Operaciones básicas	
1	Acceso a la aplicación por parte del administrador
2	Introducción de credenciales
3	El sistema recibe los datos del administrador
4	El administrador accede al contenido de la aplicación
5	El administrador accede a Gestión de Usuarios
6	El sistema muestra el listado de usuarios
Alternativas	
3.A	¿Son correctos los datos?
3.A.1	Si sí, continuar
3.A.2	Si no, el sistema mostrará un mensaje de error y volver a 2

Tabla VIII: Caso de uso "Listado de usuarios"

Análogamente, los casos de usos del sistema relacionados con los listados pertenecientes a las gestiones de las distintas entidades se ejecutarán de manera semejante.

- **Caso de uso 4: Editar deporte**

Caso de uso	Editar Deporte
Actor primario	Admin
Sistema	Sistema de Ventas
Participantes	Admin, Sistema
Condición previa	El administrador tiene que estar registrado en el sistema
Operaciones básicas	
1	Acceso a la aplicación por parte del administrador
2	Introducción de credenciales
3	El sistema recibe los datos del administrador
4	El administrador accede al contenido de la aplicación
5	El administrador accede a Gestión de Deportes y selecciona "Editar deporte" en el deporte requerido
6	El administrador tiene la posibilidad de editar todos los campos. Tras finalizar, continuar
7	El sistema recibe los datos del administrador
Alternativas	
3.A	¿Son correctos los datos?
3.A.1	Si sí, continuar
3.A.2	Si no, el sistema mostrará un mensaje de error y volver a 2
7.A	¿Son correctos los datos?
7.A.1	Si sí, finalizar
7.A.2	Si no, el sistema mostrará un mensaje de error y volver a 6

Tabla IX: Caso de uso "Editar deporte"

Cabe destacar que la edición de otras entidades se realiza de manera equivalente al anterior caso de uso.

- **Caso de uso 5: Eliminar competición**

Caso de uso	Eliminar Competición
Actor primario	Admin
Sistema	Sistema de Ventas
Participantes	Admin, Sistema
Condición previa	El administrador tiene que estar registrado en el sistema
Operaciones básicas	
1	Acceso a la aplicación por parte del administrador
2	Introducción de credenciales
3	El sistema recibe los datos del administrador
4	El administrador accede al contenido de la aplicación
5	El administrador accede a Gestión de Competiciones y selecciona "Eliminar" la competición requerida
6	El sistema muestra una notificación indicando la confirmación de la operación
7	El administrador pulsa sobre el botón de eliminar
8	El sistema se actualiza
Alternativas	
3.A	¿Son correctos los datos?
3.A.1	Si sí, continuar
3.A.2	Si no, el sistema mostrará un mensaje de error y volver a 2
6.A	¿Son correctos los datos?
6.A.1	Si sí, continuar
6.A.2	Si no, el administrador cancelará la operación y volver a 5

Tabla X: Caso de uso "Eliminar competición"

El procedimiento de eliminación anteriormente descrito puede extrapolarse a las demás entidades del sistema.

- **Caso de uso 6: Alta de evento**

Caso de uso	Alta Evento
Actor primario	Admin
Sistema	Sistema de Ventas
Participantes	Admin, Sistema
Condición previa	El administrador tiene que estar registrado en el sistema
Operaciones básicas	
1	Acceso a la aplicación por parte del administrador
2	Introducción de credenciales
3	El sistema recibe los datos del administrador
4	El administrador accede al contenido de la aplicación
5	El administrador accede a Gestión de Eventos
6	El administrador accede a "Nuevo Evento"
7	El administrador rellena los datos del nuevo evento
8	El sistema recibe los datos del administrador
Alternativas	
3.A	¿Son correctos los datos?
3.A.1	Si sí, continuar
3.A.2	Si no, el sistema mostrará un mensaje de error y volver a 2
8.A	¿Son correctos los datos?
8.A.1	Si sí, finalizar
8.A.2	Si no, el sistema mostrará un mensaje de error y volver a 7

Tabla XI: Caso de uso "Alta de evento"

Igualmente, los casos de uso relacionados con el alta de una entidad se desarrollarán de la misma manera.

- **Caso de uso 7: Puesta en venta**

Caso de uso	Puesta en Venta
Actor primario	Admin
Sistema	Sistema de Ventas
Participantes	Administrador, Sistema
Condición previa	El administrador tiene que estar registrado en el sistema, además de existir eventos cuyas entradas no estén puestas a la venta
Operaciones básicas	
1	Acceso a la aplicación por parte del administrador
2	Introducción de credenciales
3	El sistema recibe los datos del administrador
4	El administrador accede al contenido de la aplicación
5	El administrador accede a la Gestión de Entradas
6	El administrador accede a Puesta en Venta
7	El administrador selecciona el evento cuyas entradas quiere poner en venta
8	El sistema recibe los datos del administrador y se actualiza
Alternativas	
3.A	¿Son correctos los datos?
3.A.1	Si sí, continuar
3.A.2	Si no, el sistema mostrará un mensaje de error y volver a 2

Tabla XII: Caso de uso "Puesta en venta"

- **Caso de uso 8: Visualizar compra**

Caso de uso	Visualizar Compra
Actor primario	Admin
Sistema	Sistema de Ventas
Participantes	Admin, Sistema
Condición previa	El administrador tiene que estar registrado en el sistema
Operaciones básicas	
1	Acceso a la aplicación por parte del administrador
2	Introducción de credenciales
3	El sistema recibe los datos del administrador
4	El administrador accede al contenido de la aplicación
5	El administrador accede a Gestión de Compras
6	El administrador accede a "Visualizar" la entrada requerida
7	El sistema muestra los datos sobre la compra seleccionada
Alternativas	
3.A	¿Son correctos los datos?
3.A.1	Si sí, continuar
3.A.2	Si no, el sistema mostrará un mensaje de error y volver a 2

Tabla XIII: Caso de uso "Visualizar Compra"

El método de visualización de una compra es homólogo al resto de operaciones de este tipo.

- **Caso de uso 9: Buscar eventos**

Caso de uso	Comprar entradas
Actor primario	Cliente
Sistema	Sistema de Ventas
Participantes	Cliente, Sistema
Condición previa	El cliente tiene que estar registrado en el sistema
Operaciones básicas	
1	Acceso a la aplicación por parte del cliente
2	Introducción de credenciales
3	El sistema recibe los datos del cliente
4	El cliente accede al contenido de la aplicación
5	El cliente utiliza el buscador para rastrear el evento deseado o bien pulsa en la opción "Ver Eventos" del menú, en cuyo caso tendrá que seleccionar varias categorías de búsqueda
6	El sistema recibe los datos del cliente
7	El sistema muestra una lista con los eventos que coinciden con el/los criterio/s de búsqueda, según el método de búsqueda utilizado
Alternativas	
3.A	¿Son correctos los datos?
3.A.1	Si sí, continuar
3.A.2	Si no, el sistema mostrará un mensaje de error y volver a 2
6.A	¿Son correctos los datos?
6.A.1	Si sí, continuar
6.A.2	Si no, el sistema mostrará un mensaje de error y volver a 5

Tabla XIV: Caso de uso "Buscar eventos"

- **Caso de uso 10: Comprar entradas**

Caso de uso	Comprar entradas
Actor primario	Cliente
Sistema	Sistema de Ventas
Participantes	Cliente, Sistema
Condición previa	El cliente tiene que estar registrado en el sistema
Operaciones básicas	
1	Acceso a la aplicación por parte del cliente
2	Introducción de credenciales
3	El sistema recibe los datos del cliente
4	El cliente accede al contenido de la aplicación
5	El cliente utiliza el buscador para rastrear el evento deseado o bien pulsa en la opción "Ver Eventos" del menú, en cuyo caso tendrá que seleccionar varias categorías de búsqueda
6	El sistema recibe los datos del cliente
7	El sistema muestra una lista con los eventos que coinciden con el/los criterio/s de búsqueda, según el método de búsqueda utilizado
8	El cliente pulsa el botón donde se habilita la posibilidad de ver las entradas del evento en cuestión
9	El sistema muestra una lista con las entradas disponibles para ese evento
10	El cliente selecciona las entradas requeridas y acciona el botón de compra
11	El sistema recibe los datos del cliente
12	El sistema muestra la información acerca de la compra realizada
Alternativas	
3.A	¿Son correctos los datos?
3.A.1	Si sí, continuar
3.A.2	Si no, el sistema mostrará un mensaje de error y volver a 2
6.A	¿Son correctos los datos?
6.A.1	Si sí, continuar
6.A.2	Si no, el sistema mostrará un mensaje de error y volver a 5
11.A	¿Son correctos los datos?
11.A.1	Si sí, continuar
11.A.2	Si no, el sistema mostrará un mensaje de error y volver a 10

Tabla XV: Caso de uso "Comprar entradas"

- **Caso de uso 11: Pagar entradas**

Caso de uso	Pagar entradas
Actor primario	Cliente
Sistema	Sistema de Ventas
Participantes	Cliente, Sistema
Condición previa	El cliente tiene que estar registrado en el sistema
Operaciones básicas	
1	Acceso a la aplicación por parte del cliente
2	Introducción de credenciales
3	El sistema recibe los datos del cliente
4	El cliente accede al contenido de la aplicación
5	El cliente utiliza el buscador para rastrear el evento deseado o bien pulsa en la opción "Ver Eventos" del menú, en cuyo caso tendrá que seleccionar varias categorías de búsqueda
6	El sistema recibe los datos del cliente
7	El sistema muestra una lista con los eventos que coinciden con el/los criterio/s de búsqueda, según el método de búsqueda utilizado
8	El cliente pulsa el botón donde se habilita la posibilidad de ver las entradas del evento en cuestión
9	El sistema muestra una lista con las entradas disponibles para ese evento
10	El cliente selecciona las entradas requeridas y acciona el botón de compra
11	El sistema recibe los datos del cliente
12	El sistema muestra la información acerca de la compra realizada
13	El cliente acciona el botón de pago con PayPal
14	El sistema redirige al usuario a la página de PayPal, donde se muestra el resumen de la compra realizada y el formulario de acceso para que el cliente la pague
15	El cliente introduce sus credenciales para PayPal
16	PayPal recibe los datos del cliente
17	El cliente accede a PayPal y autoriza el pago
18	PayPal redirige al cliente a la aplicación
19	El sistema muestra la confirmación del pago y los datos de la compra
Alternativas	
3.A	¿Son correctos los datos?

3.A.1	Si sí, continuar
3.A.2	Si no, el sistema mostrará un mensaje de error y volver a 2
6.A	¿Son correctos los datos?
6.A.1	Si sí, continuar
6.A.2	Si no, el sistema mostrará un mensaje de error y volver a 5
11.A	¿Son correctos los datos?
11.A.1	Si sí, continuar
11.A.2	Si no, el sistema mostrará un mensaje de error y volver a 10
12.A	¿Son correctos los datos?
12.A.1	Si sí, continuar
12.A.2	Si no, el cliente podrá utilizar el botón de cancelación de compra y volver a 5
16.A	¿Son correctos los datos?
16.A.1	Si sí, continuar
16.A.2	Si no, el sistema mostrará un mensaje de error y volver a 15

Tabla XVI: Caso de uso "Pagar entradas"

- **Caso de uso 12: Ver compras realizadas**

Caso de uso	Ver compras realizadas
Actor primario	Cliente
Sistema	Sistema de Ventas
Participantes	Cliente, Sistema
Condición previa	El cliente tiene que estar registrado en el sistema
Operaciones básicas	
1	Acceso a la aplicación por parte del cliente
2	Introducción de credenciales
3	El sistema recibe los datos del cliente
4	El cliente accede al contenido de la aplicación
5	El cliente selecciona la opción habilitada en el menú para visualizar las compras realizadas
6	El sistema muestra al cliente una lista con las compras realizadas por éste
Alternativas	
3.A	¿Son correctos los datos?
3.A.1	Si sí, continuar
3.A.2	Si no, el sistema mostrará un mensaje de error y volver a 2

Tabla XVII: Caso de uso "Ver compras realizadas"

- **Caso de uso 13: Imprimir entradas**

Caso de uso	Imprimir entradas
Actor primario	Cliente
Sistema	Sistema de Ventas
Participantes	Cliente, Sistema
Condición previa	El cliente tiene que estar registrado en el sistema y haber comprado alguna entrada
Operaciones básicas	
1	Acceso a la aplicación por parte del cliente
2	Introducción de credenciales
3	El sistema recibe los datos del cliente
4	El cliente accede al contenido de la aplicación
5	El cliente selecciona la opción habilitada en el menú para visualizar las compras realizadas
6	El sistema muestra al cliente una lista con las compras realizadas por éste
7	El cliente acciona el botón preparado para imprimir las entradas de una determinada compra
8	El sistema redirige al cliente a la página de impresión de la compra, donde se muestra la información relacionada con la misma, las entradas y el botón para imprimirlas
9	El cliente pulsa sobre el botón "Imprimir entradas"
10	El sistema presenta la ventana de impresión
11	El cliente acciona el botón de imprimir

12	El sistema imprime las entradas
Alternativas	
3.A	¿Son correctos los datos?
3.A.1	Si sí, continuar
3.A.2	Si no, el sistema mostrará un mensaje de error y volver a 2

Tabla XVIII: Caso de uso "Imprimir entradas"

2.9. Modelo de dominio

Para finalizar el análisis del sistema y a partir de los requisitos ([apartado 2.5](#)) y casos de uso descritos anteriormente ([apartado 2.8](#)), procederemos a realizar el modelo de dominio del sistema.

El modelo de dominio muestra a los desarrolladores un conjunto de clases conceptuales significativas en un dominio concreto del problema. Se utiliza con frecuencia como fuente de inspiración para el diseño de los objetos software [21].

Puede considerarse como el punto de partida para el diseño del sistema. Proporciona al analista una visión estructural del prototipo de negocio, con el objetivo de desarrollar el sistema requerido por el cliente.

El modelo de dominio se representa mediante un conjunto de clases u objetos conceptuales en los que no se define ninguna operación. Generalmente suelen presentar los siguientes elementos:

- Objetos del dominio o clases conceptuales
- Asociaciones entre las clases conceptuales
- Atributos de las clases conceptuales

A continuación se muestra el modelo de dominio de la aplicación desde un punto de vista conceptual:

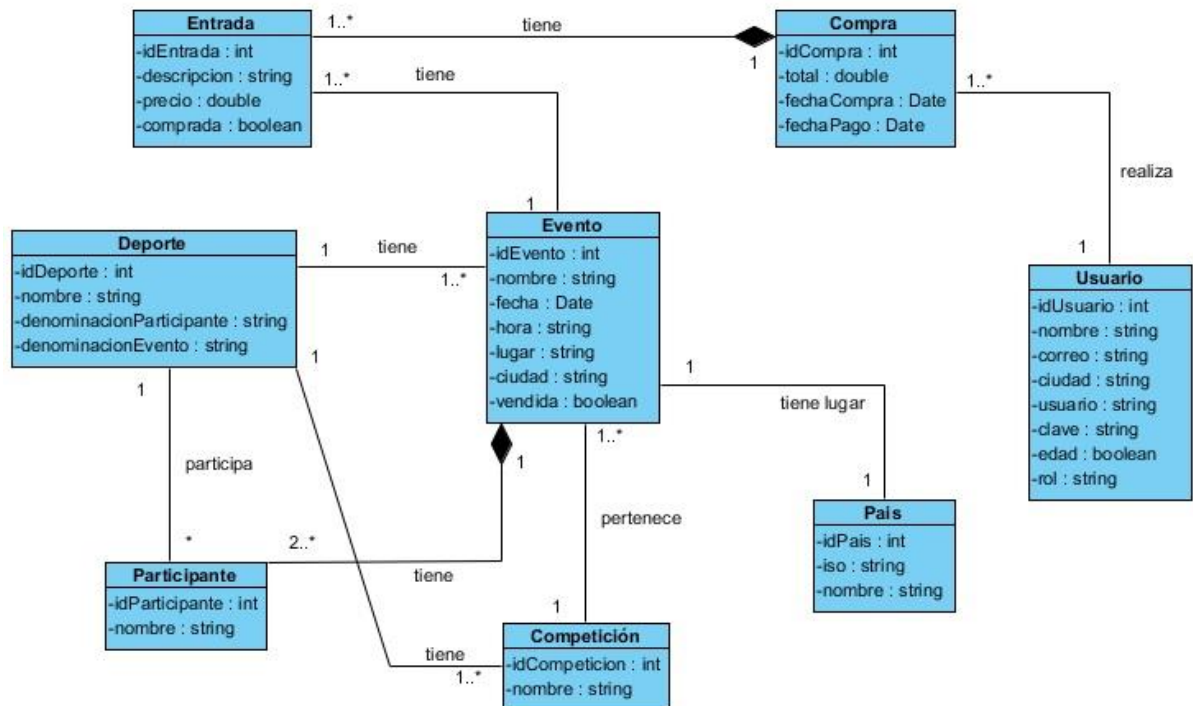


Ilustración 20: Modelo de Dominio

Como puede observarse, el modelo de dominio está compuesto por varias entidades. La más relevante es la clase evento, que se asocia con otras entidades para formar una clase conceptual fuerte, albergando el deporte, la competición, los participantes y el país pertenecientes a un evento.

Por otro lado, un evento tiene muchas entradas y cada una de ellas pertenece exclusivamente a un evento.

Por último, cada compra es realizada por un usuario y tiene asociadas una cantidad determinada de entradas.

3. Diseño

Esta fase se centra en proporcionar la funcionalidad al sistema a través de sus componentes. Para la mayoría de los sistemas existen muchos diseños posibles que cumplen los requerimientos. Éstos comprenden una amplia gama de soluciones que combinan hardware, software y operaciones humanas. Debería elegirse la solución técnica más apropiada que cumpla los requerimientos [22].

En los siguientes apartados se verán cada una de las fases del diseño del sistema:

3.1. Diseño de la representación de la información

El objetivo de este apartado consiste en determinar las características de los datos de la aplicación, proporcionando las estructuras adecuadas. En el proceso, se incluye la identificación y la definición de los tipos de datos, así como los mecanismos de almacenamiento concretos.

Los elementos de información del sistema son los siguientes:

- **Usuarios:** se almacena un identificador de usuario, nombre, correo, ciudad, nombre de usuario, clave, edad y rol.
- **Deportes:** se almacena un identificador de deporte, nombre, denominación del participante y denominación del evento.
- **Competiciones:** se almacena un identificador de competición, nombre y el deporte al que pertenecen.
- **Participantes:** se almacena un identificador de participante, nombre y el deporte en el que participan.
- **Eventos:** se almacena un identificador de evento, nombre, fecha, hora, lugar, si las entradas están puestas a la venta, país, deporte, competición y participantes.
- **Entradas:** se almacena un identificador de entrada, descripción, precio, si está comprada y el evento al que pertenece.
- **Compras:** se almacena un identificador de compra, total, usuario que la ha realizado, las entradas asociadas, fecha de compra y fecha de pago.
- **Países:** se almacena un identificador de país, código iso y nombre.

Una vez especificados los elementos de información del sistema, es el momento de realizar el modelo Entidad-Relación, para posteriormente obtener las tablas de la base de datos.

3.1.1. Modelo Entidad-Relación

El modelo entidad-relación (MER) describe el “mundo real” como un conjunto de entidades y de relaciones entre ellas. Es la técnica de modelado de datos más extendida y se representa de forma gráfica a través del diagrama Entidad-Relación (DER) [23].

El DER define todos los datos que se introducen, se almacenan, se transforman y se producen dentro de una aplicación [2]. Está formado por entidades, representadas mediante un círculo, atributos caracterizados mediante elipses que se definen como propiedades o características que describen a una entidad y relaciones que se corresponden con asociaciones o vínculos entre entidades y se escenifican por medio de rombos.

Un concepto importante dentro de una relación es la cardinalidad. Puede definirse como la especificación del número de ocurrencias de un objeto que se relaciona con ocurrencias de otro objeto. Dos objetos se pueden relacionar como uno a uno, uno a muchos o muchos a muchos [2].

Existen dos aspectos a tener en cuenta a la hora de construir un diagrama Entidad-Relación:

- **Clave primaria:** atributo o conjunto de atributos que identifican de forma unívoca a una entidad.
- **Clave foránea:** atributo o conjunto de atributos de una entidad que son clave primaria en otra entidad con la cual se relaciona.

Una vez conocidos los elementos que forman parte de un diagrama Entidad-Relación podemos desarrollar el modelo Entidad-Relación.

Cabe destacar la importancia de trabajar en nuestro sistema con la técnica de mapeado objeto-relacional (ORM), que proporciona persistencia en una base de datos relacional a una aplicación orientada a objetos. Concretamente, se ha utilizado

Por lo tanto, a partir del modelo de dominio definido en el [capítulo 2.9](#), se ha generado de manera automática el modelo Entidad-Relación. Es importante especificar que el modelo de dominio desarrollado se ha realizado teniendo en cuenta la obtención de la tercera forma normal, para asegurar que no hay duplicidad de datos. Seguidamente se adjunta la ilustración correspondiente al modelo Entidad-Relación de nuestro sistema:

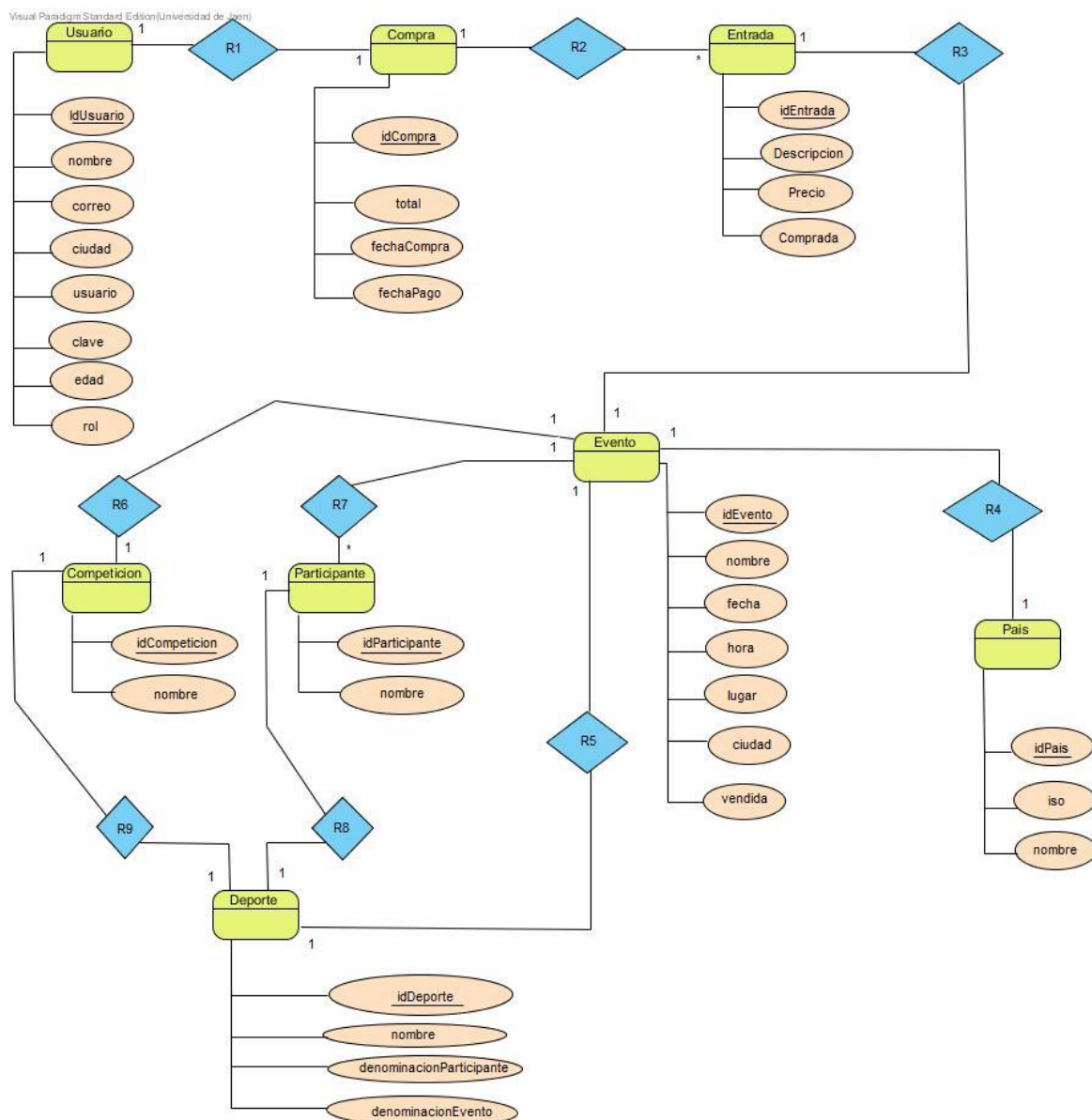


Ilustración 21: Modelo Entidad-Relación

La siguiente tabla define las relaciones especificadas en el modelo Entidad-Relación anterior:

Relaciones	
R1	Es realizada
R2	Contiene
R3	Pertenece
R4	Tiene lugar
R5	Contiene
R6	Contiene
R7	Contiene
R8	Participa
R9	Pertenece

Tabla XIX: Relaciones

3.1.2. Tablas de la aplicación

Una vez realizado el mapeado objeto relacional anteriormente descrito, se han obtenido las siguientes tablas:

- **Usuario**

Tabla que contiene información sobre los usuarios del sistema. Contiene los siguientes campos:

CAMPO	TIPO	DESCRIPCIÓN
IdUsuario	Int	Identificador unívoco del usuario. Será la clave primaria y de tipo auto incremental
Nombre	String	Nombre y apellidos del usuario
Correo	String	Correo del usuario
Ciudad	String	Ciudad del usuario
Usuario	String	Nombre de usuario
Clave	String	Contraseña del usuario
Edad	Boolean	Mayoría de edad del usuario
Rol	String	Rol del usuario

Tabla XX: Tabla de usuarios

- **Deporte**

Tabla que almacena los deportes registrados en el sistema. Está formada por los siguientes campos:

CAMPO	TIPO	DESCRIPCIÓN
IdDeporte	Int	Identificador unívoco del deporte. Será la clave primaria y de tipo auto incremental
Nombre	String	Nombre del deporte
DenominaciónParticipante	String	Término utilizado para la denominación del participante de ese deporte (Equipos, Tenistas, Pilotos, etc.)
DenominaciónEvento	String	Término utilizado para la denominación del evento de ese deporte (Partidos, Carreras, etc.)

Tabla XXI: Tabla de deportes

- **Competición**

Tabla que alberga las competiciones existentes en el sistema. Abarca los siguientes campos:

CAMPO	TIPO	DESCRIPCIÓN
IdCompetición	Int	Identificador unívoco de la competición. Será la clave primaria y de tipo auto incremental
Nombre	String	Nombre de la competición
IdDeporte	Int	Deporte al que pertenece la competición

Tabla XXII: Tabla de competiciones

- **Participante**

Tabla que incluye información sobre los participantes de los distintos deportes del sistema. Comprende los campos siguientes:

CAMPO	TIPO	DESCRIPCIÓN
IdParticipante	Int	Identificador unívoco del participante. Será la clave primaria y de tipo auto incremental
Nombre	String	Nombre del participante
IdDeporte	Int	Deporte que practica el participante

Tabla XXIII: Tabla de participantes

- **País**

Tabla que contiene los países del mundo.

CAMPO	TIPO	DESCRIPCIÓN
IdPais	Int	Identificador unívoco del país. Será la clave primaria y de tipo auto incremental
Iso	String	Codificación del país
Nombre	String	Nombre del país

Tabla XXIV: Tabla de países

- **Evento**

Tabla que almacena los datos de los eventos registrados en el sistema. Está formada por los siguientes campos:

CAMPO	TIPO	DESCRIPCIÓN
IdEvento	Int	Identificador unívoco del evento. Será la clave primaria y de tipo auto incremental
Nombre	String	Nombre del evento
Fecha	Date	Fecha del evento
Hora	String	Hora del evento
Lugar	String	Lugar del evento
Ciudad	String	Ciudad del evento
Vendida	Boolean	Indica si las entradas de un evento están puestas a la venta
IdPais	Int	País del evento
IdDeporte	Int	Deporte correspondiente al evento
IdCompeticion	Int	Competición perteneciente al evento

Tabla XXV: Tabla de eventos

- **Participa**

Tabla que contiene los eventos asociados a cada uno de los participantes

CAMPO	TIPO	DESCRIPCIÓN
IdEvento	Int	Identificador unívoco del evento
IdParticipante	Int	Identificador unívoco del participante

Tabla XXVI: Tabla "participa"

- **Entrada**

Tabla que incluye información relacionada con las entradas de los eventos. Abarca los campos siguientes:

CAMPO	TIPO	DESCRIPCIÓN
IdEntrada	Int	Identificador unívoco de la entrada. Será la clave primaria y de tipo auto incremental.
Descripcion	String	Descripción de la entrada.
Precio	Double	Precio de la entrada.
Comprada	Boolean	Indica si una entrada está comprada.
IdEvento	Int	Evento al que pertenece la entrada.

Tabla XXVII: Tabla de entradas

- **Compra**

Tabla que alberga las compras realizadas a través del sistema por parte de los usuarios. Contiene los campos siguientes:

CAMPO	TIPO	DESCRIPCIÓN
IdCompra	Int	Identificador unívoco de la compra. Será la clave primaria y de tipo auto incremental.
Total	Double	Precio total de la compra.
IdUsuario	Int	Usuario que ha realizado la compra.
FechaCompra	Date	Fecha en la que se efectuó la compra.
FechaPago	Date	Fecha de pago de la compra.

Tabla XXVIII: Tabla de compras

- **Compras_entradas**

Tabla que incluye las entradas pertenecientes a una compra

CAMPO	TIPO	DESCRIPCIÓN
IdCompra	Int	Identificador unívoco de la compra
IdEntrada	Int	Identificador unívoco de la entrada

Tabla XXIX: Tabla “compras_entradas”

3.2. Diagrama de Clases

El diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos (datos almacenados), operaciones o métodos y las relaciones entre los objetos [2].

Una clase se representa mediante un triángulo dividido en tres secciones: la primera muestra el nombre de la clase, la segunda los atributos y la tercera y última las operaciones.

Por otro lado, las relaciones existentes en un diagrama de este tipo nos indican cómo se interrelacionan los objetos de dos o más clases. Los mensajes “navegan” a través de las relaciones definidas entre éstas.

Una vez definidos los elementos de un diagrama de clases es el momento de realizarlo. La comunicación entre las distintas partes del sistema se va a producir siguiendo el modelo-vista-controlador (MVC), que se define como un estilo de arquitectura software que determina la separación del código en tres componentes diferentes, según su responsabilidad, denominados modelos, vistas y controladores [24]:

- El Modelo constituye la representación la información con la que trabaja la aplicación, es decir, su lógica de negocio y los mecanismos de persistencia. Se encarga de enviar a la vista la información solicitada por el usuario.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La aplicación web realiza, por tanto, una separación de responsabilidades con el objetivo de que el acoplamiento sea mínimo. La estructuración llevada a cabo consiste en definir distintos paquetes que permiten aislar los componentes principales. A continuación se muestra un primer diagrama de paquetes relacionados representando el modelo-vista-controlador:

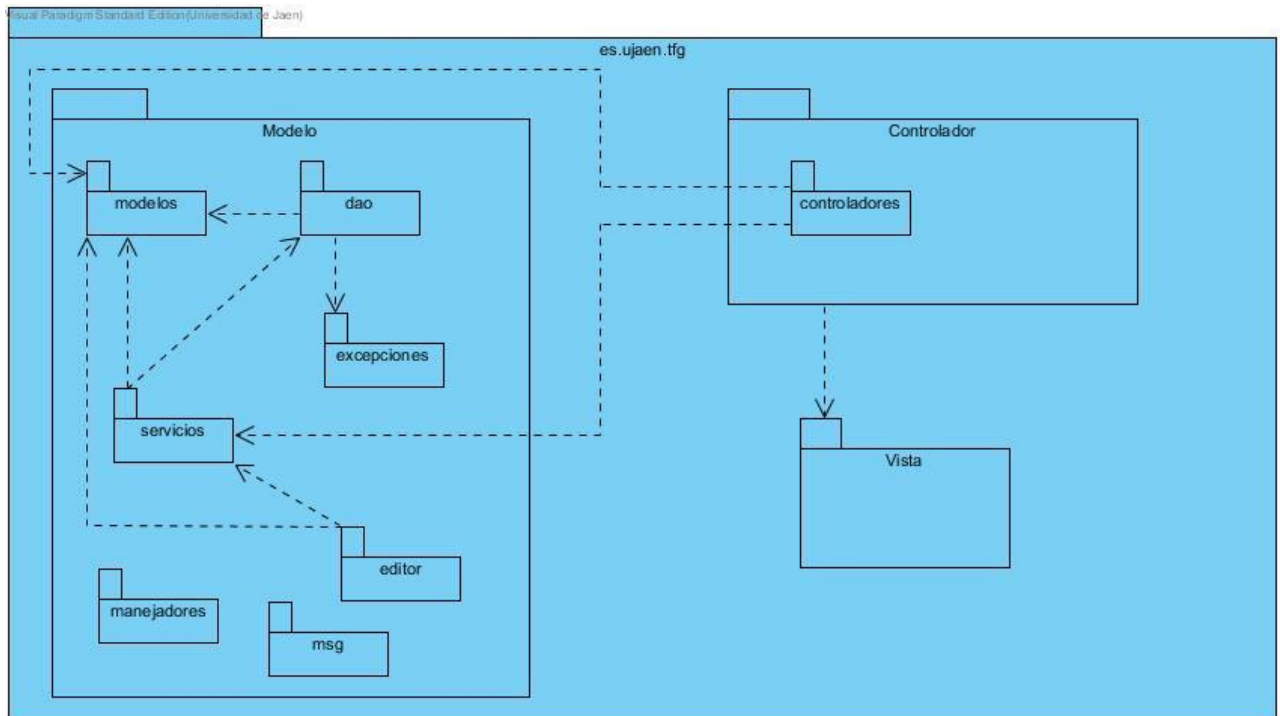


Ilustración 22: Diagrama de paquetes MVC

Como puede observarse, algunos paquetes del diagrama contienen a su vez paquetes concretos, por lo que pueden ser detallados en mayor profundidad:

- **Paquete modelos:** contiene la entidades que modelan la información de la lógica de negocio de la aplicación, es decir, el almacenamiento de los datos y los métodos necesarios para manipularlos.
- **Paquete dao:** posee los métodos necesarios para consultar, insertar, modificar y eliminar información de la base de datos.
- **Paquete servicios:** facilita la interacción entre el controlador y los daos, definiendo métodos que proporcionan los datos persistentes utilizados por la lógica de negocio.
- **Paquete excepciones:** se utiliza para controlar las excepciones producidas en la ejecución de una operación relacionada con la base de datos.
- **Paquete editor:** proporciona métodos de utilidad para convertir un String proveniente de la interfaz de usuario a un tipo dado, por medio del registro de un PropertyEditor utilizando InitBinder y WebDataBinder [25].
- **Paquete controladores:** engloba a todos los controladores, que actuarán de intermediarios entre el modelo y la vista. Contiene métodos

para comunicarse con el modelo con el fin de obtener los datos generados tras la interacción del usuario con la interfaz.

El paquete **manejadores** contiene la clase que permite redirigir los diferentes tipos de usuarios (administrador y cliente) a diferentes páginas después de iniciar sesión, mientras que el paquete **msg** permite almacenar todos los mensajes de error de validación referentes a un formulario.

Seguidamente se materializa un ejemplo concreto de la representación del Modelo-Vista-Controlador de la aplicación para dar de alta un usuario:

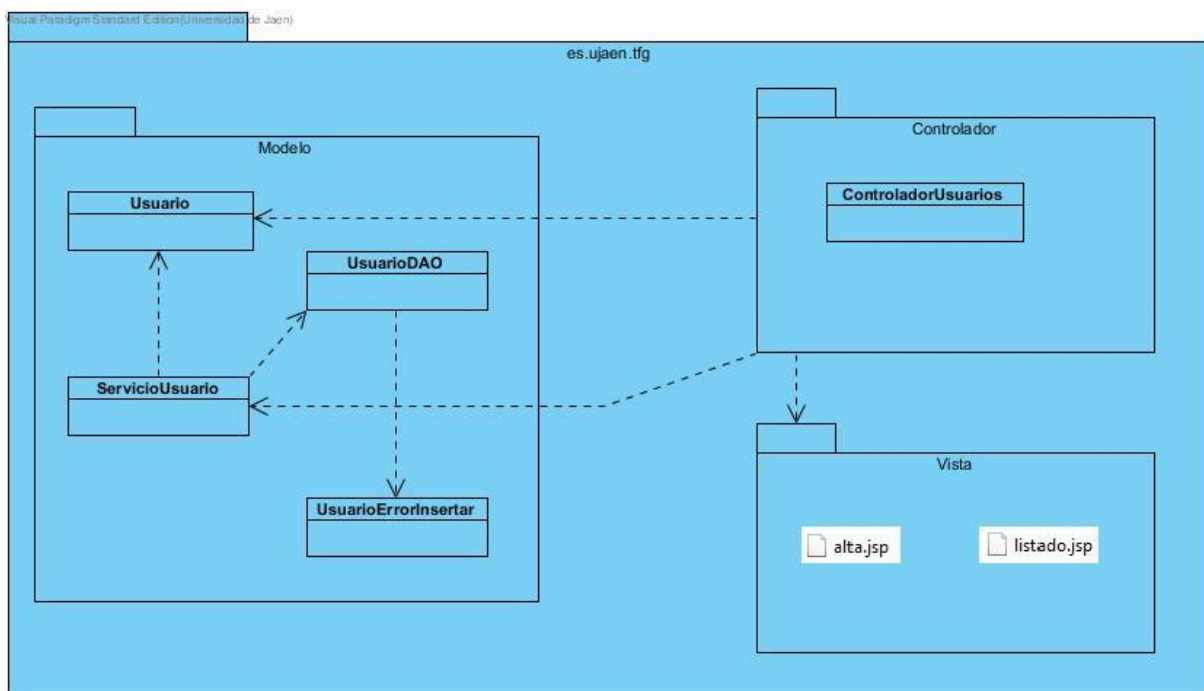


Ilustración 23: Ejemplo MVC “Alta de usuario”

El controlador redirige al administrador al listado de usuarios. Para ello, invoca a una función del servicio, que obtiene una lista con los usuarios del sistema a través de un método del dao correspondiente y finalmente pasa la información encontrada a la vista. Una vez en el listado, el administrador selecciona la opción habilitada para dar de alta un nuevo usuario y el controlador lo conduce a la vista del alta, enviando un objeto de tipo **Usuario** a dicha vista. A continuación, el administrador rellena los campos del formulario, enviándose los datos al servicio. Si son correctos, la clase **UsuarioDAO** se encargará de insertar el nuevo usuario en la base de datos y el controlador redirigirá al administrador nuevamente al listado. En caso contrario, el

sistema lanzará una excepción y el administrador será conducido a la vista del alta, donde se mostrarán los errores.

Cabe destacar que el resto de controladores contienen una organización similar.

A continuación se definirán nuevos diagramas que permitan visualizar las clases que forman parte de cada uno de los paquetes del diagrama de paquetes MVC anterior:

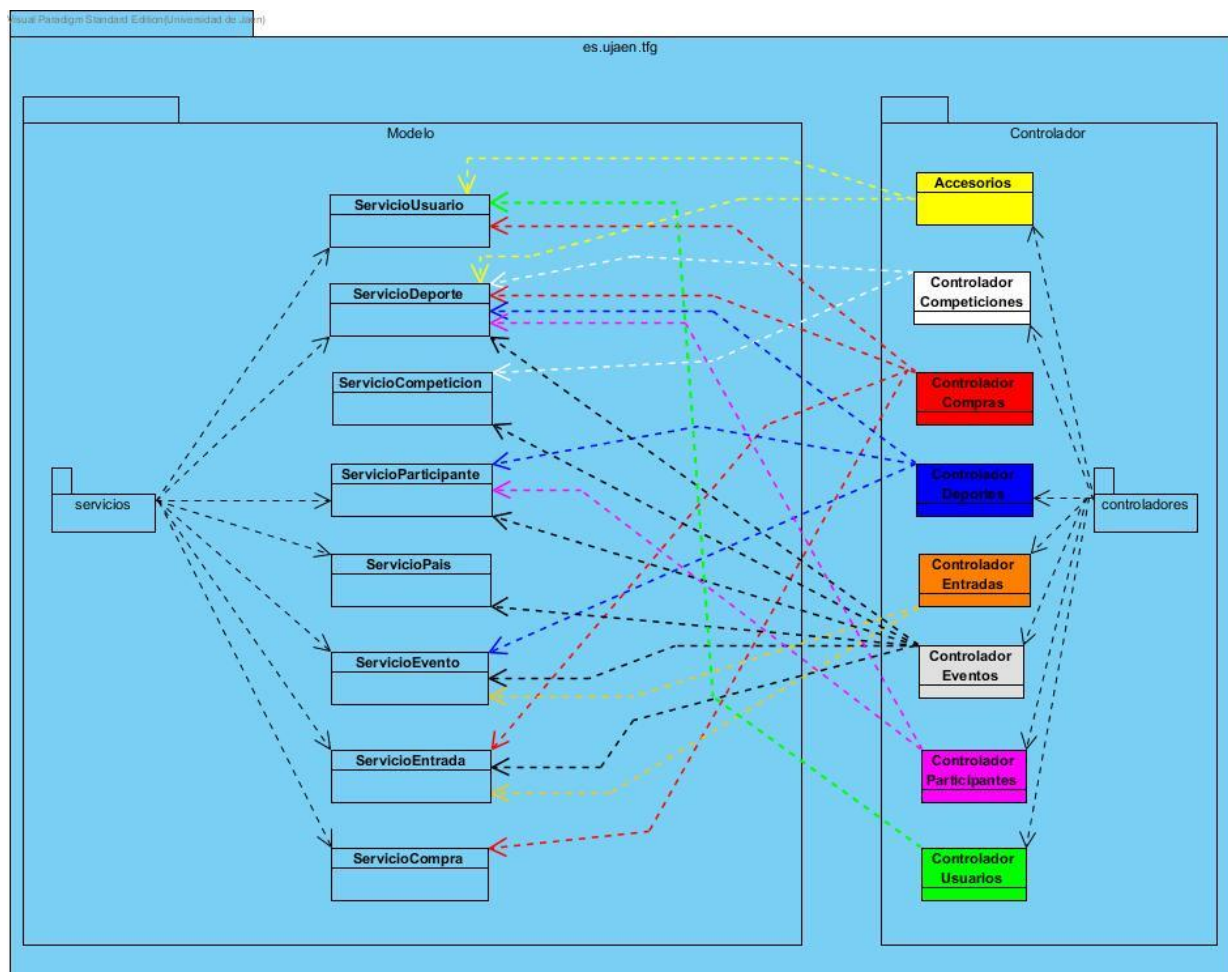


Ilustración 24: Controladores & Servicios

La ilustración anterior muestra las relaciones existentes entre los controladores y los servicios del sistema, estableciéndose así un vínculo entre el paquete controlador y el paquete modelo.

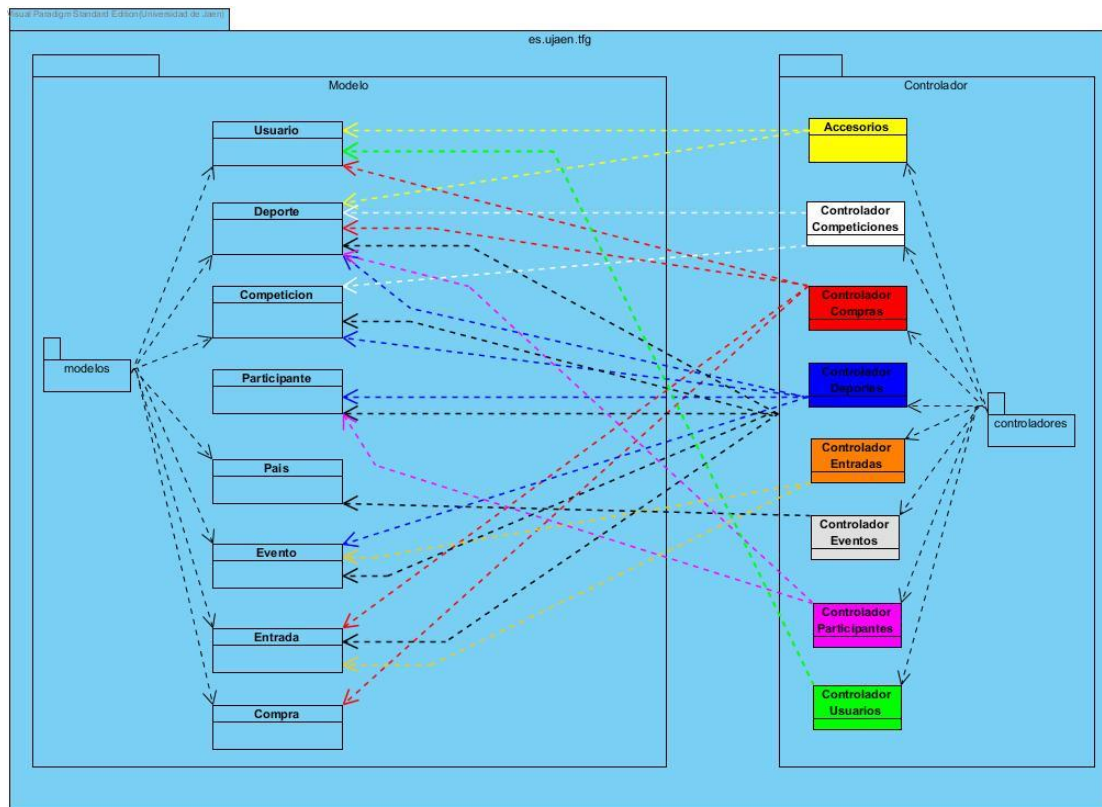


Ilustración 25: Controladores & Modelos

En la anterior imagen puede apreciarse las relaciones existentes entre los controladores y los modelos del sistema, estableciéndose así un nexo entre el paquete controlador y el paquete modelo.

Por ejemplo, el controlador de compras se relaciona con los modelos usuario, deporte, entrada y compra.

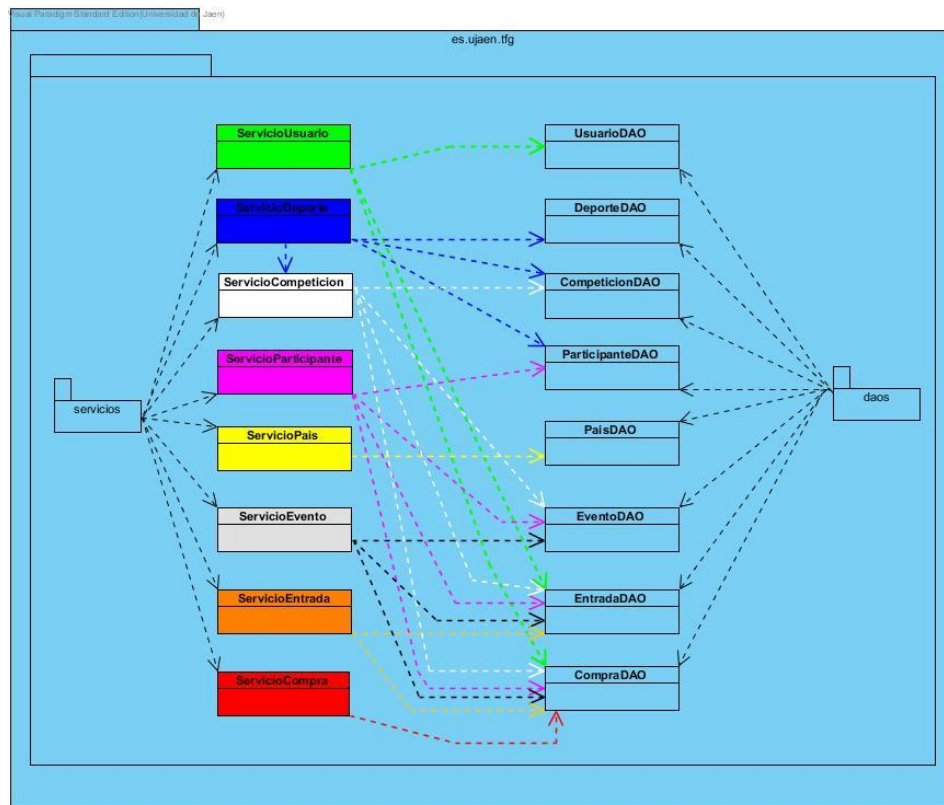


Ilustración 26: Servicios & Daos

La ilustración anterior presenta las relaciones existentes entre los servicios y los daos del sistema, ambos pertenecientes al paquete modelo.

Por citar un ejemplo, el servicio correspondiente al deporte se relaciona con los daos de deporte, competición y participante.

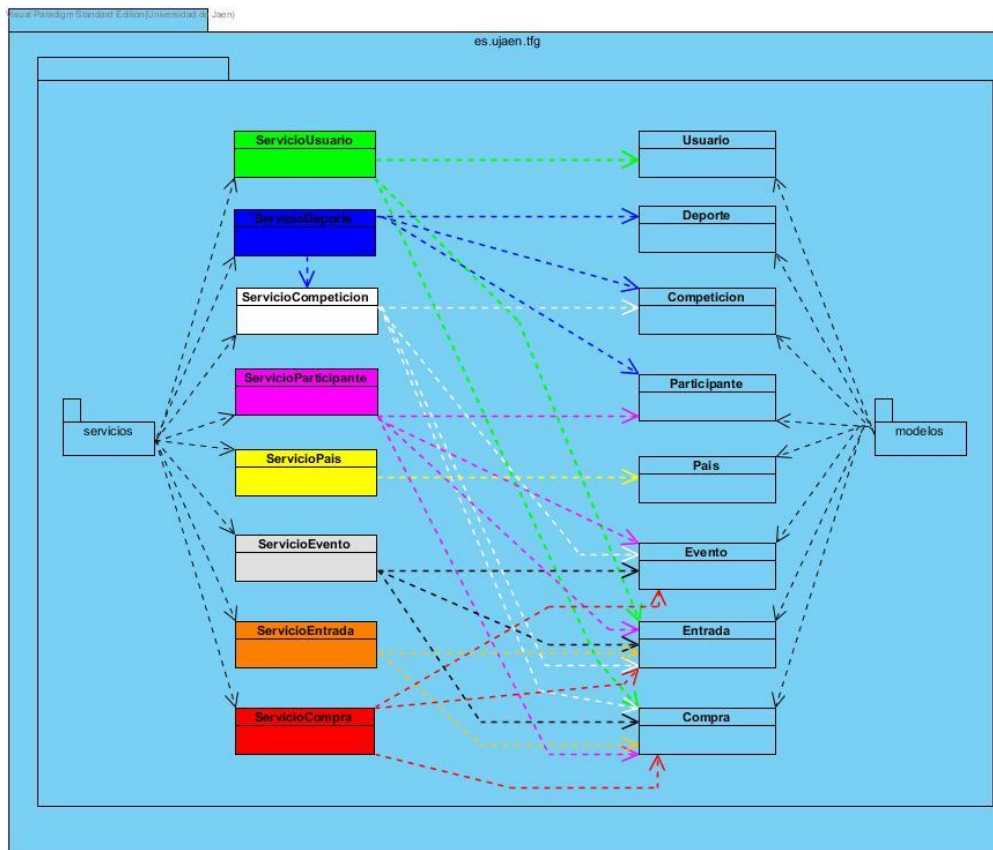


Ilustración 27: Servicios & Modelos

Puede apreciarse en la imagen las relaciones existentes entre los servicios y los modelos del sistema, ambos pertenecientes al paquete modelo.

Un ejemplo de relación sería la establecida entre el servicio de entrada y los modelos entrada y compra.

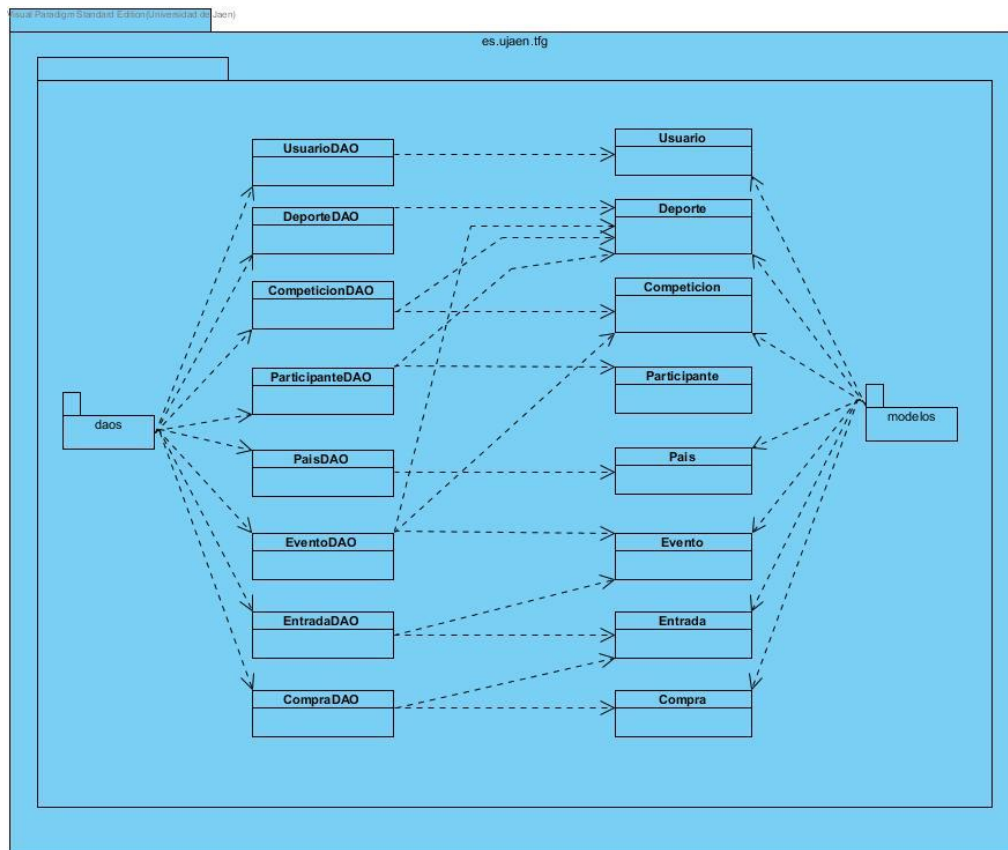


Ilustración 28: Daos & Modelos

La ilustración anterior muestra las relaciones existentes entre los daos y los modelos del sistema, ambos situados en el paquete modelo.

Por ejemplo, el dao de participante se relaciona con los modelos del participante y de la competición.

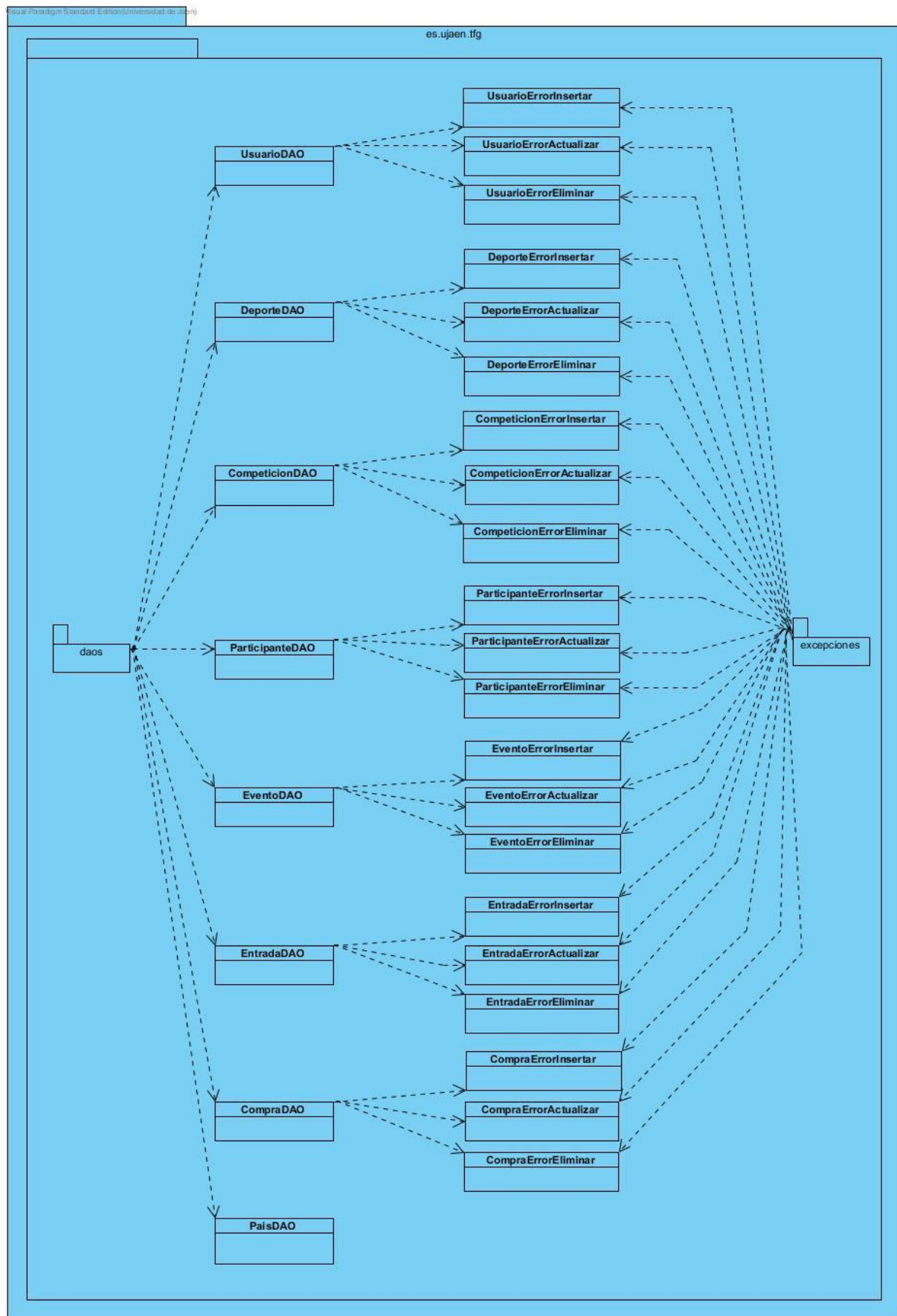


Ilustración 29: Daos & Excepciones

Se observa en la ilustración anterior las relaciones existentes entre los daos y las excepciones del sistema, ambos pertenecientes al paquete modelo.

Un ejemplo de relación sería la establecida entre el dao de compra y las excepciones correspondientes a la inserción, actualización y eliminación de la misma.

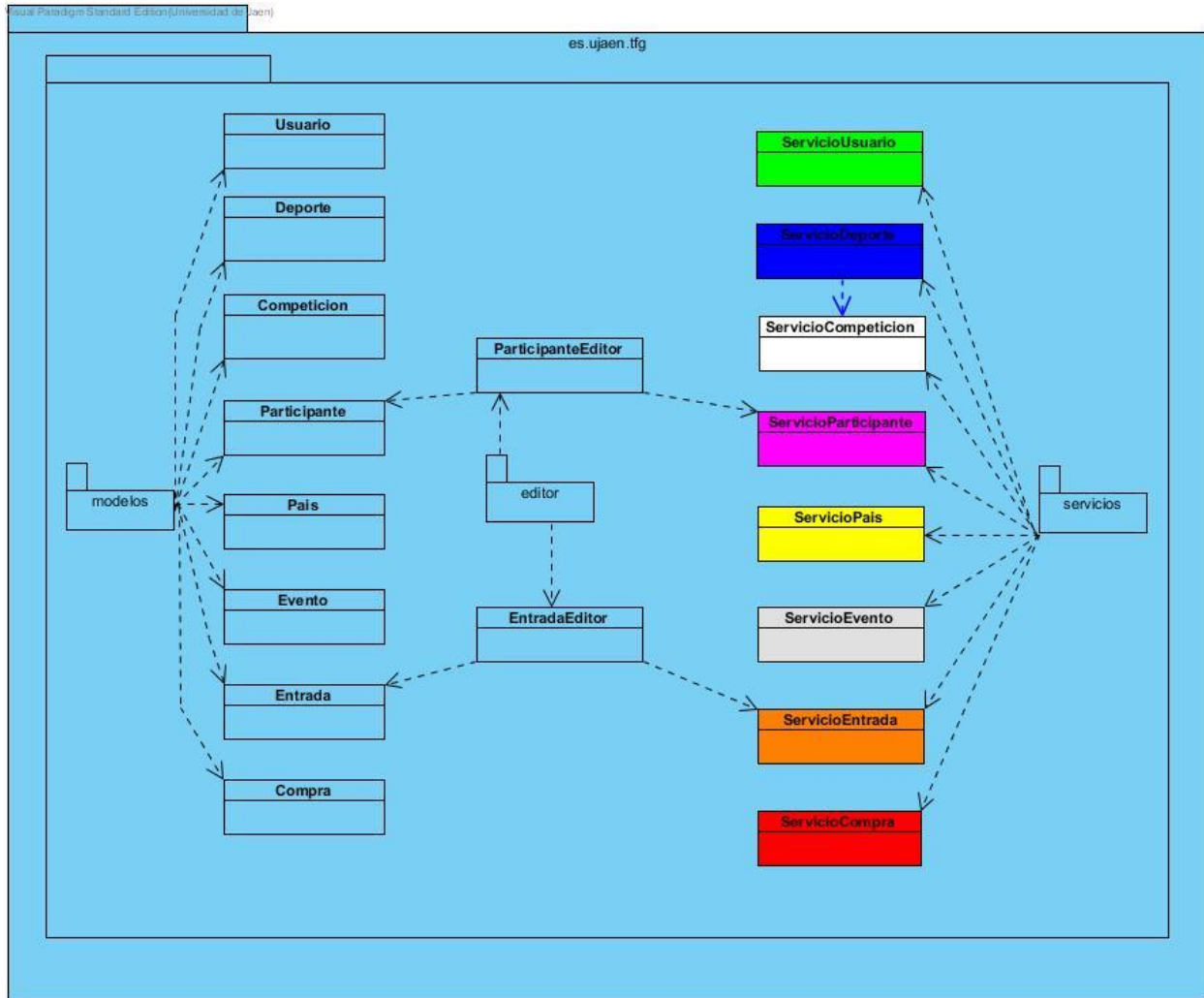


Ilustración 30: Editor

La ilustración anterior muestra las relaciones existentes entre los modelos, los servicios y el paquete editor.

Por citar un ejemplo, el editor del participante se relaciona con el modelo y el servicio del mismo.

3.3. Diagrama de Secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el modelado de una vista del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos [2].

La representación general de un diagrama de secuencia se lleva a cabo de la siguiente manera: la dimensión vertical simboliza el tiempo, mientras que los objetos involucrados en la interacción están distribuidos horizontalmente en el diagrama. Esta ordenación horizontal de los objetos se realiza de manera arbitraria, aunque mejora la legibilidad si la interacción progresa de izquierda a derecha. Cada objeto está representado por una línea de vida, que se define por una línea vertical discontinua y con un símbolo de objeto en su parte superior. Por último, los mensajes se representan con flechas horizontales que van desde una línea de vida a otra. Se pueden numerar secuencialmente a partir del valor 1 y es posible utilizar una numeración compuesta [22].

Un fragmento combinado es una sección dentro de una interacción que incluye bucles u otras alternativas. Para ejecutarlas dentro de un fragmento combinado se ha utilizado el operador **alt**, mientras que para la representación de interacciones complejas se ha empleado el operador **ref**, el cual permite simplificar el diagrama de secuencia correspondiente, referenciando desde el fragmento combinado otros fragmentos de interacción [22].

A continuación se muestran los diagramas de secuencia correspondientes a los casos de uso más importantes de los bloques funcionales definidos en la especificación de requerimientos del sistema:

- **Diagrama de secuencia: Identificación**

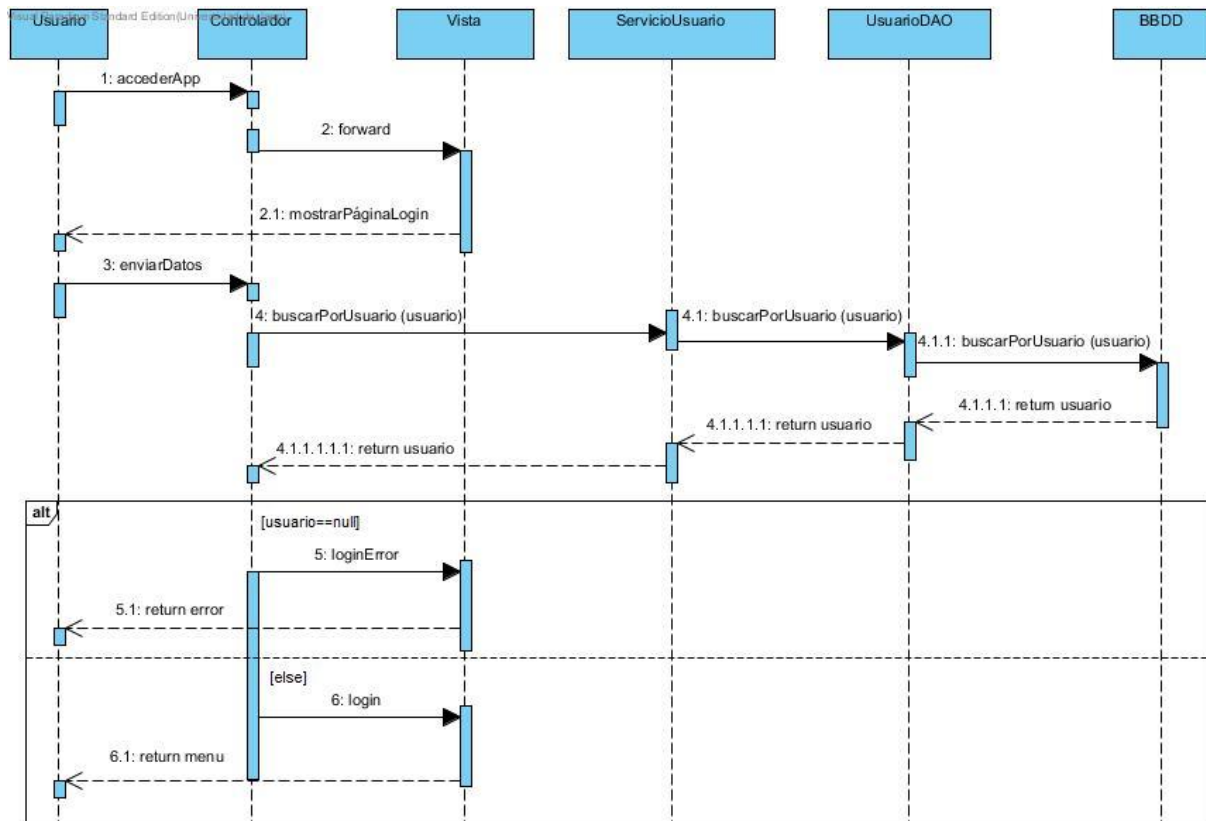


Ilustración 31: DS "Identificación"

El usuario accede a la aplicación para identificarse. Una vez ha introducido sus credenciales, el sistema las comprueba. Si no existe un usuario con esas credenciales, muestra un error. Si por el contrario existe, redirige al usuario al menú de inicio.

- **Diagrama de secuencia: Registrarse**

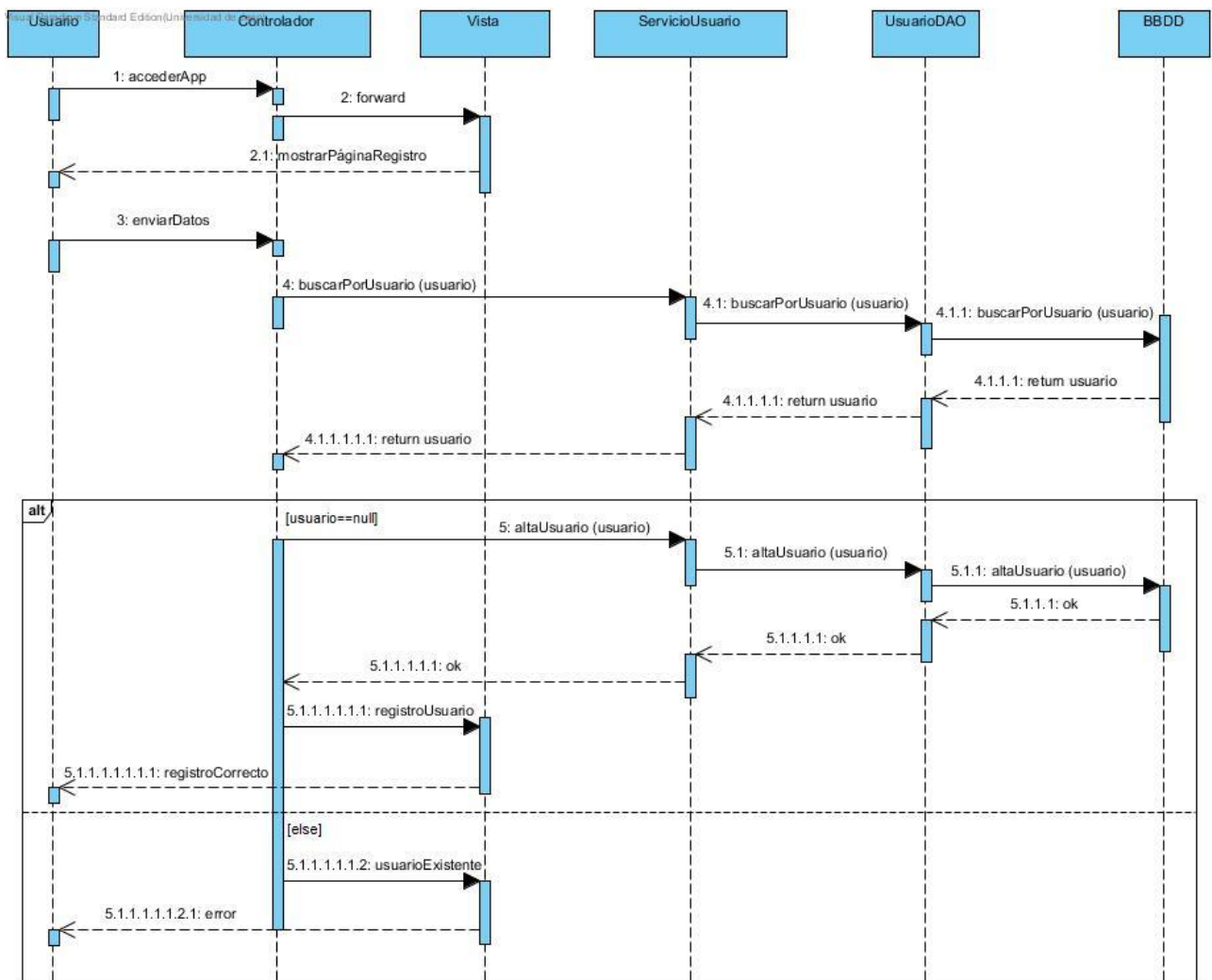


Ilustración 32: DS "Registrarse"

El usuario accede a la aplicación para registrarse. Una vez ha rellenado todos los campos, el sistema los comprueba. Si no existe un usuario cuyo nombre de usuario coincida con el nombre de usuario introducido, se registra en el sistema. Si por el contrario existe, el sistema muestra un mensaje indicando que ya existe un usuario con ese nombre de usuario.

- **Diagrama de secuencia: Listado de usuarios**

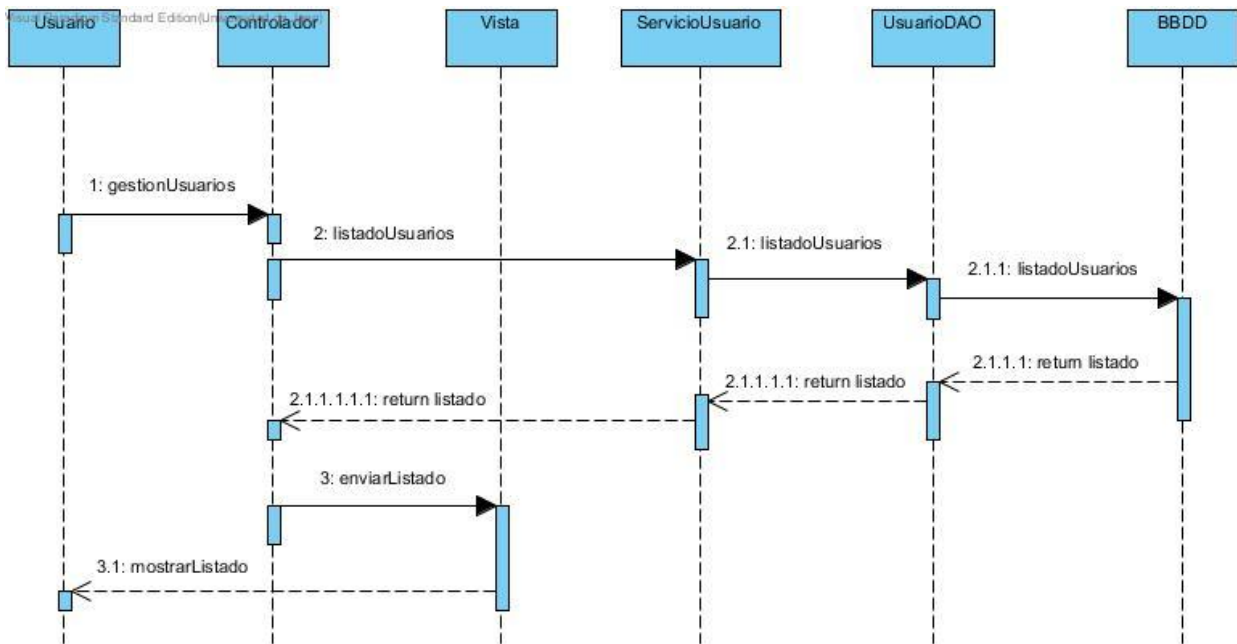


Ilustración 33: DS "Listado de usuarios"

Una vez logueado, el usuario administrador accede a la opción del menú habilitada para gestionar los usuarios. El sistema busca los usuarios y devuelve un listado de los mismos.

- **Diagrama de secuencia: Editar deporte**

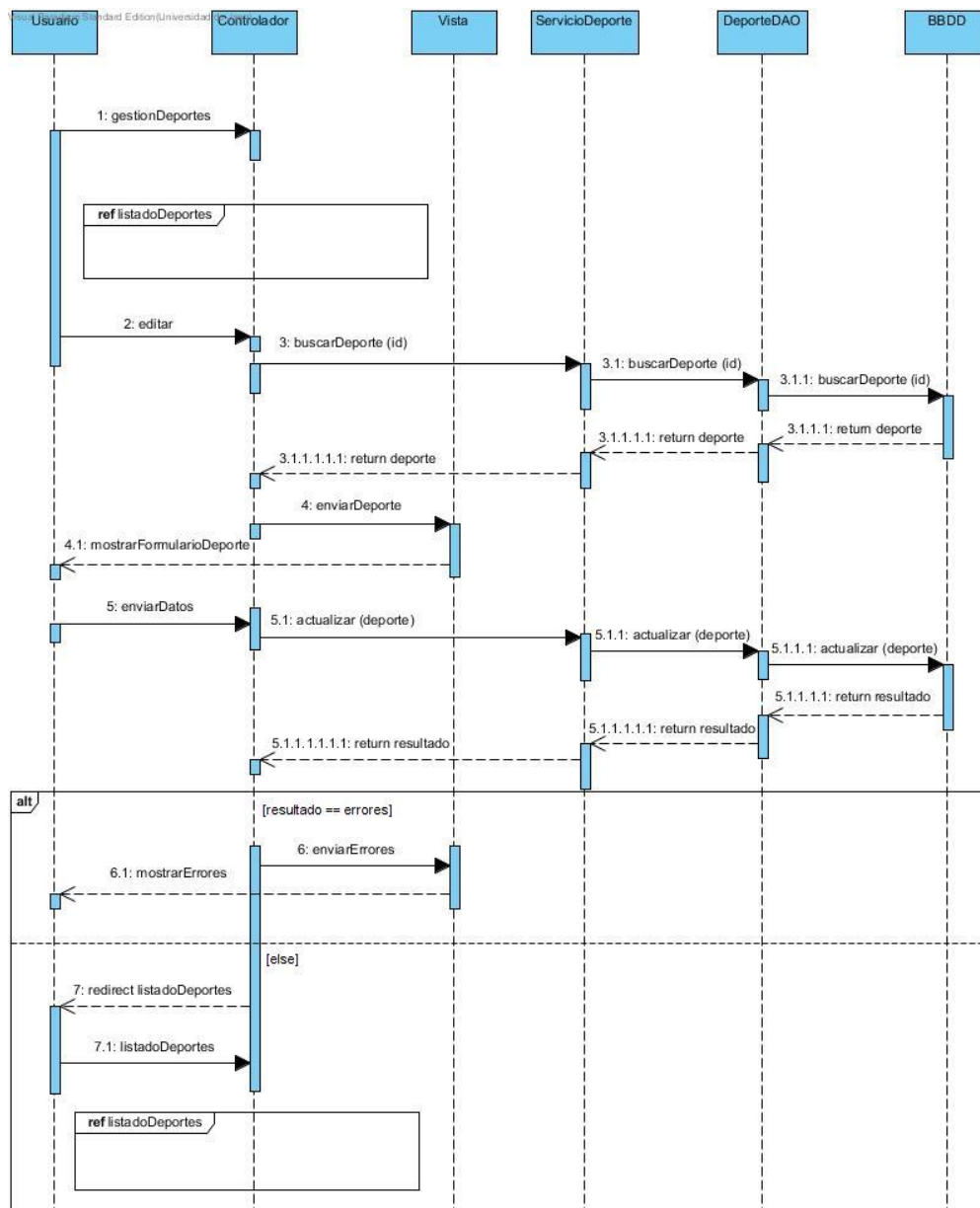


Ilustración 34: DS "Editar deporte"

Una vez logueado, el usuario administrador accede a la opción del menú habilitada para gestionar los deportes. El sistema busca los deportes y devuelve un listado de los mismos, ejecutándose otro diagrama de secuencia (etiqueta ref). A continuación, el usuario administrador selecciona el deporte que quiere editar y el sistema se lo muestra. El siguiente paso consiste en editar los campos correspondientes. Finalmente, si existen errores en la edición, el sistema los indicará. Si no, el sistema redirigirá al administrador al listado de deportes, referenciándose de nuevo a otra interacción.

- **Diagrama de secuencia: Eliminar competición**

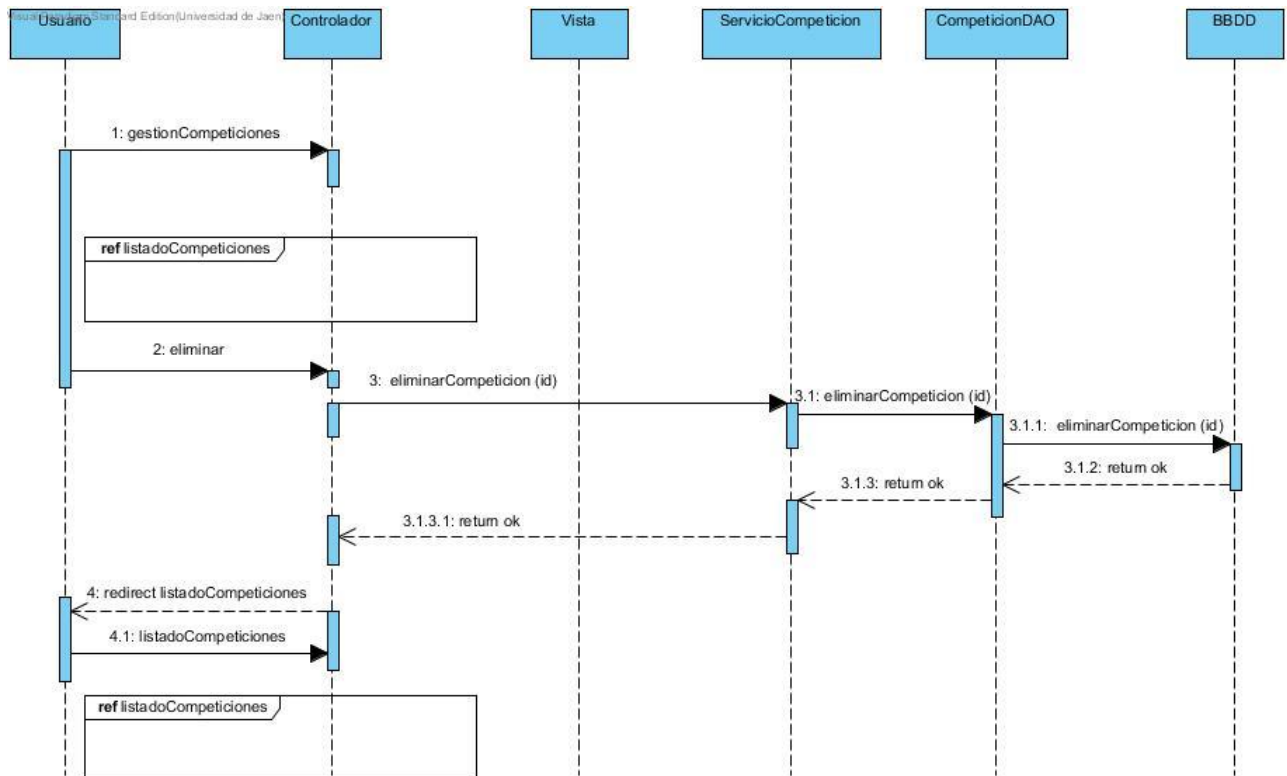


Ilustración 35: DS "Eliminar competición"

Una vez logueado, el usuario administrador accede a la opción del menú habilitada para gestionar las competiciones. El sistema busca las competiciones y devuelve un listado de las mismas, referenciándose a otro diagrama de secuencia (ref). A continuación, el usuario administrador selecciona el deporte que quiere editar y el sistema se lo muestra. El siguiente paso consiste en editar los campos correspondientes. Finalmente, si existen errores en la edición, el sistema los indicará. Si no, el sistema redirigirá al administrador al listado de deportes, ejecutándose otro diagrama de secuencia.

- **Diagrama de secuencia: Alta de evento**

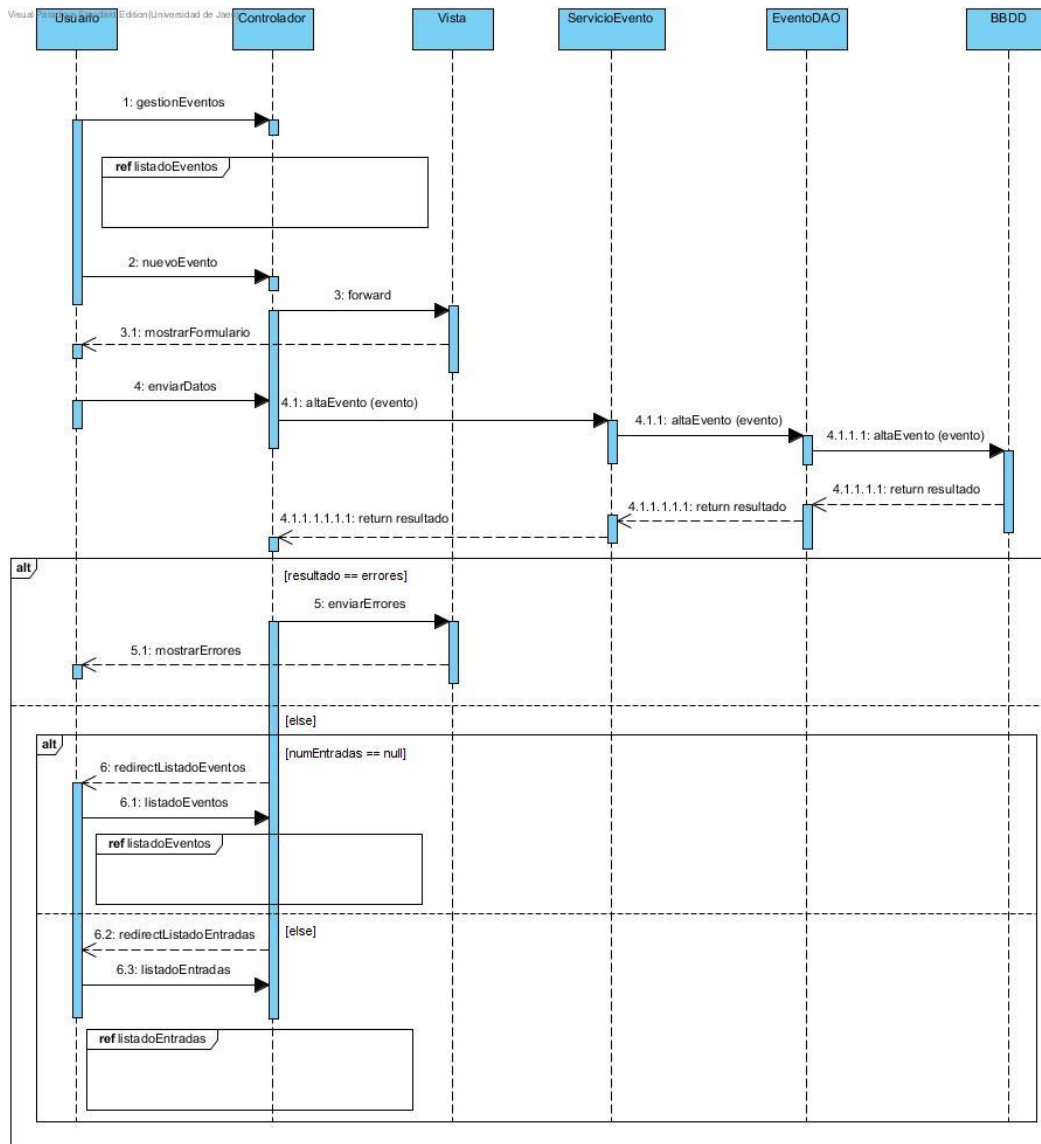


Ilustración 36: DS "Alta de evento"

Una vez logueado, el usuario administrador accede a la opción del menú habilitada para gestionar los eventos. El sistema busca los eventos y devuelve un listado de los mismos, referenciándose a otro diagrama de secuencia (ref). A continuación, el usuario administrador selecciona el botón proporcionado para dar de alta un nuevo evento y el sistema muestra el formulario. El siguiente paso consiste en rellenar los campos correspondientes. Finalmente, si existen errores en el formulario, el sistema los indicará. Si por el contrario no contiene ningún error, el sistema redirigirá al administrador al listado de eventos si el número de entradas para el evento es cero o al listado de entradas si el número de entradas es mayor que cero, todo ello efectuándose en diagramas de secuencia diferentes.

- **Diagrama de secuencia: Puesta en venta**

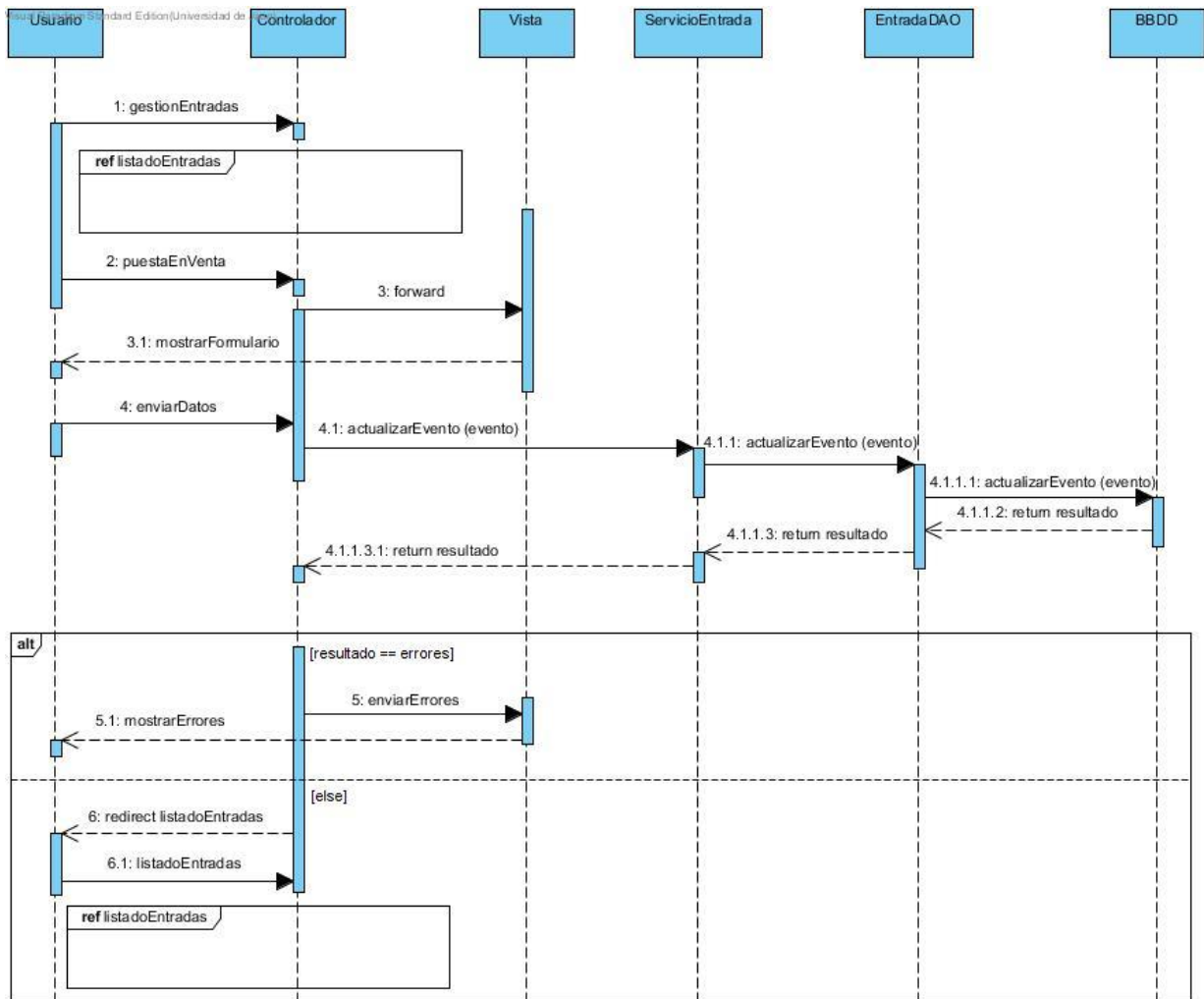


Ilustración 37: DS "Puesta en venta"

Una vez logueado, el usuario administrador accede a la opción del menú habilitada para gestionar las entradas. El sistema busca las entradas y devuelve un listado de las mismas, referenciándose a otro diagrama de secuencia (ref). A continuación, el usuario administrador selecciona el botón proporcionado para poner en venta las entradas de un evento y el sistema muestra el formulario. El siguiente paso consiste en seleccionar el evento cuyas entradas serán puestas en venta. Finalmente, si existen errores en el formulario, el sistema los indicará. Si por el contrario no contiene ningún error, el sistema redirigirá al administrador al listado de entradas, realizándose esta interacción en un diagrama de secuencia diferente.

- **Diagrama de secuencia: Visualizar compra**

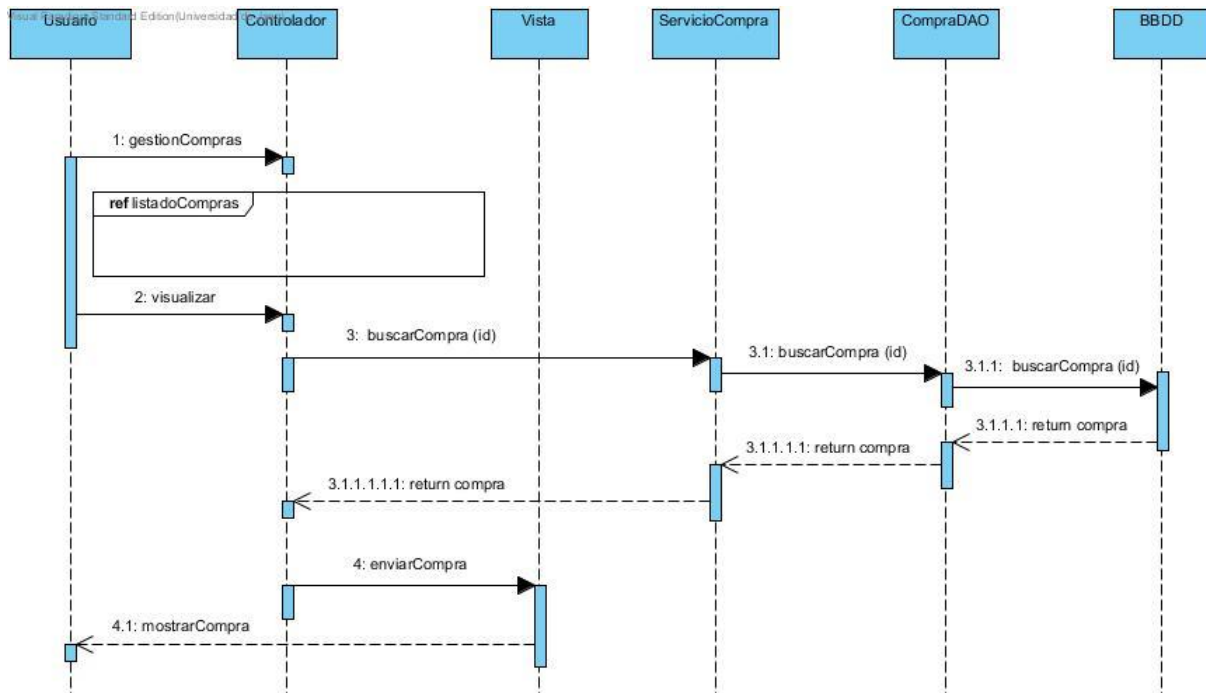


Ilustración 38: DS "Visualizar compra"

Una vez logueado, el usuario administrador accede a la opción del menú habilitada para gestionar las compras. El sistema busca las compras existentes y devuelve el listado de las mismas, incluyéndose esta interacción en otro diagrama de secuencia (ref). A continuación, el usuario administrador selecciona el botón proporcionado para visualizar una compra determinada y el sistema presenta la información relacionada con esta.

- **Diagrama de secuencia: Buscar eventos**

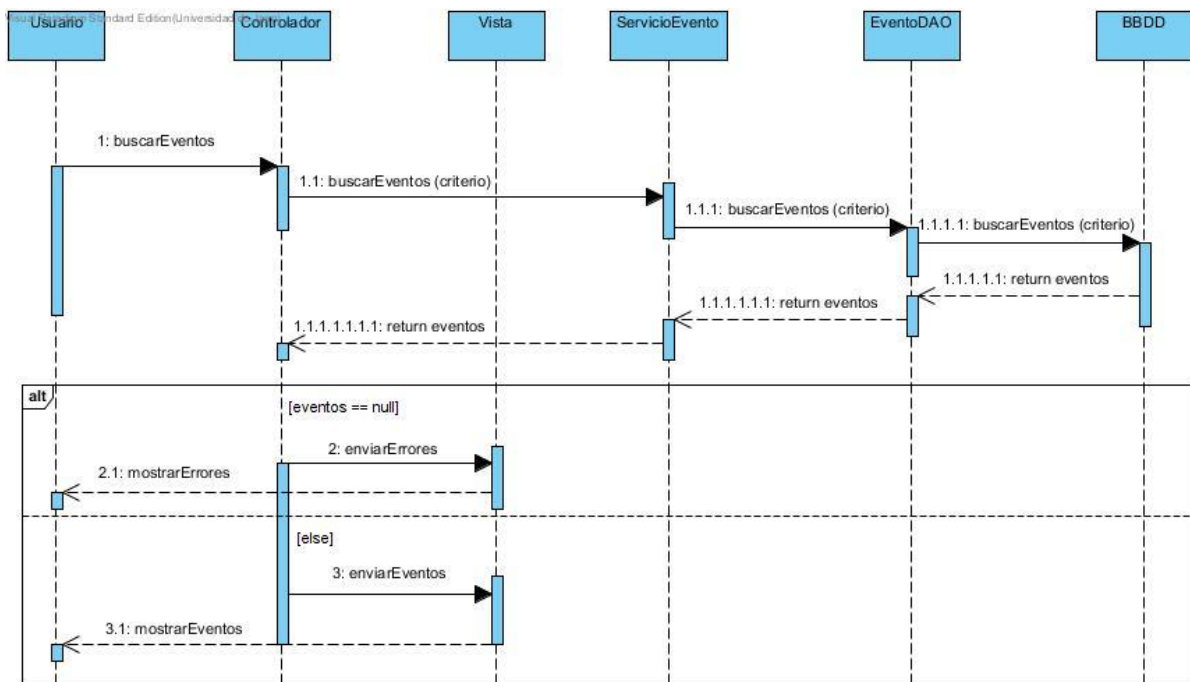


Ilustración 39: DS "Buscar eventos"

Una vez logueado, el usuario cliente accede a la opción del menú habilitada para localizar un evento o utiliza el buscador. El sistema busca los eventos que coinciden con los criterios de búsqueda introducidos y los devuelve. En caso de no existir ningún evento que se ajuste a los criterios dados, devolverá un mensaje de error.

- **Diagrama de secuencia: Comprar entradas**

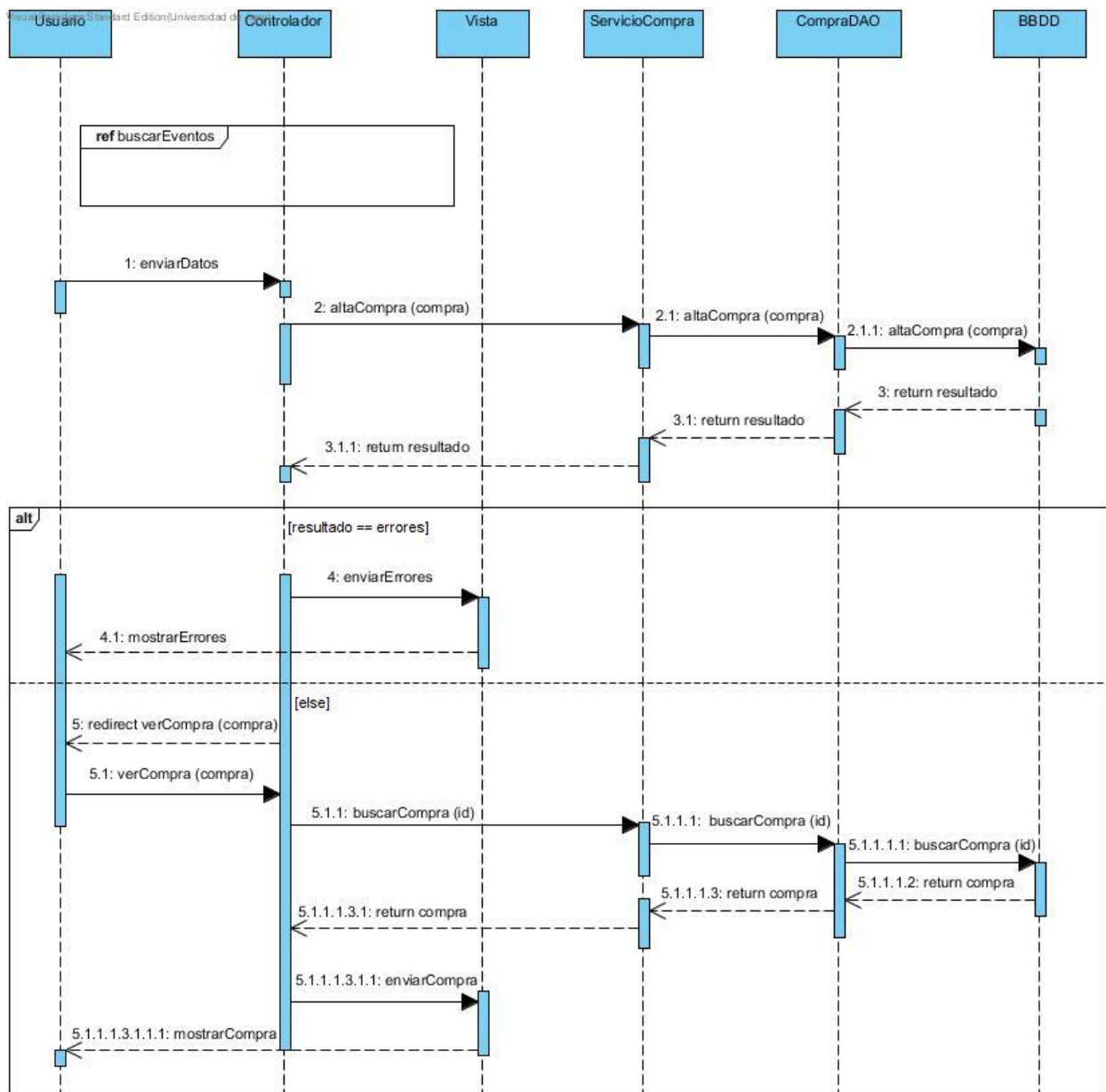


Ilustración 40: DS "Comprar entradas"

Una vez logueado, el usuario cliente localiza el evento deseado, proceso llevado a cabo en otro diagrama de secuencia (ref). A continuación seleccionará las entradas y realizará la compra. Si se ha producido algún error, el sistema informará al cliente al respecto. En caso contrario, mostrará la información de la compra realizada.

- **Diagrama de secuencia: Pagar entradas**

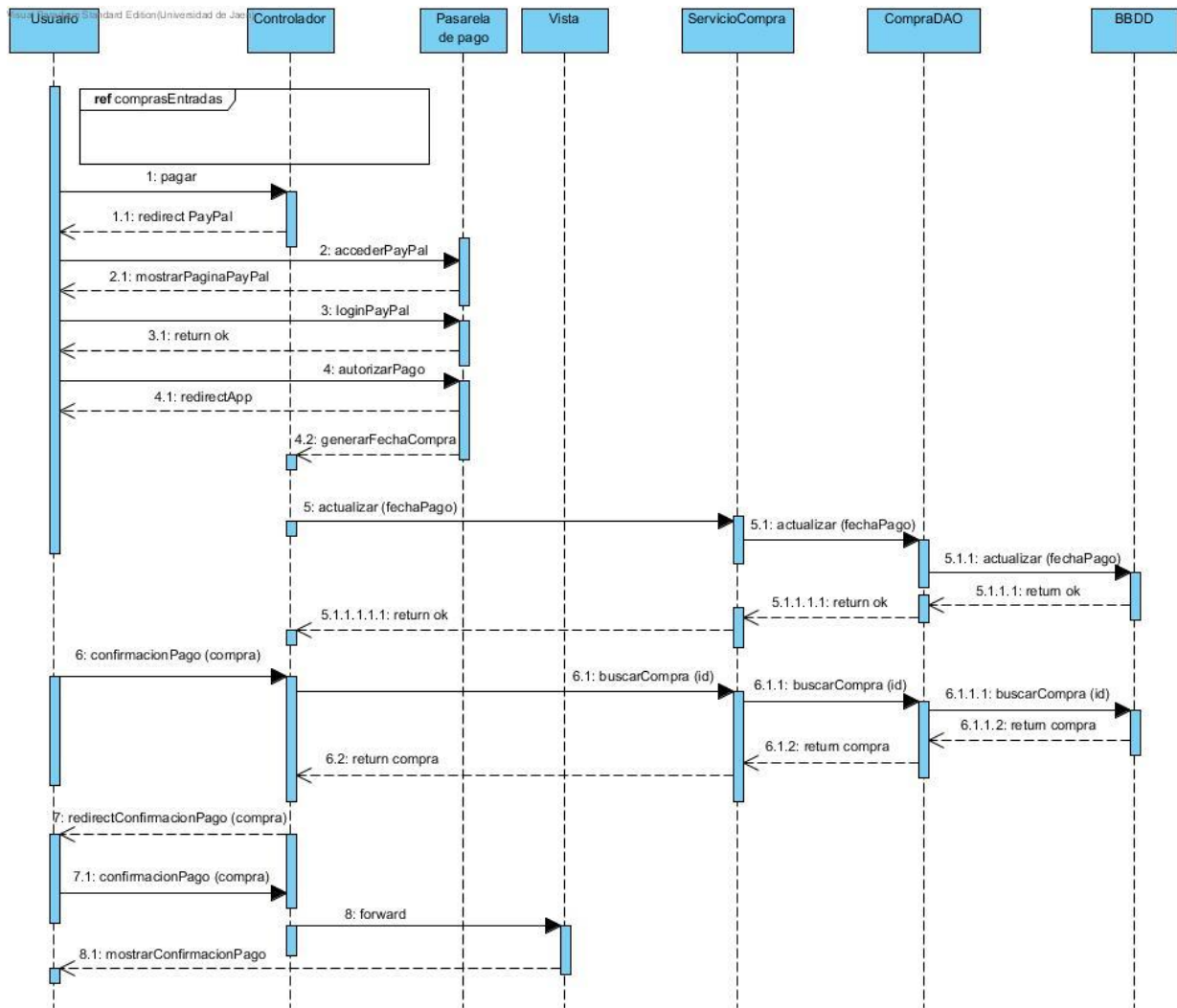


Ilustración 41: DS "Pagar entradas"

Una vez logueado, el usuario compra una serie de entradas para un evento concreto, cuyo proceso se realizará en un diagrama de secuencia diferente (ref). A continuación, accionará el botón de pago habilitado y será conducido a PayPal, donde le aparecerá el resumen de la compra efectuada y el formulario de identificación. Tras introducir sus credenciales y acceder a la plataforma, el cliente autoriza el pago de la compra. En este momento, PayPal lanza dos operaciones de manera concurrente: la primera es redirigir al cliente a la aplicación web, donde le aparecerá la confirmación de que se ha abonado el precio total de la compra, y la segunda consiste en la generación de la fecha de compra, de manera que PayPal indica que el pago ha sido realizado.

- **Diagrama de secuencia: Ver compras realizadas**

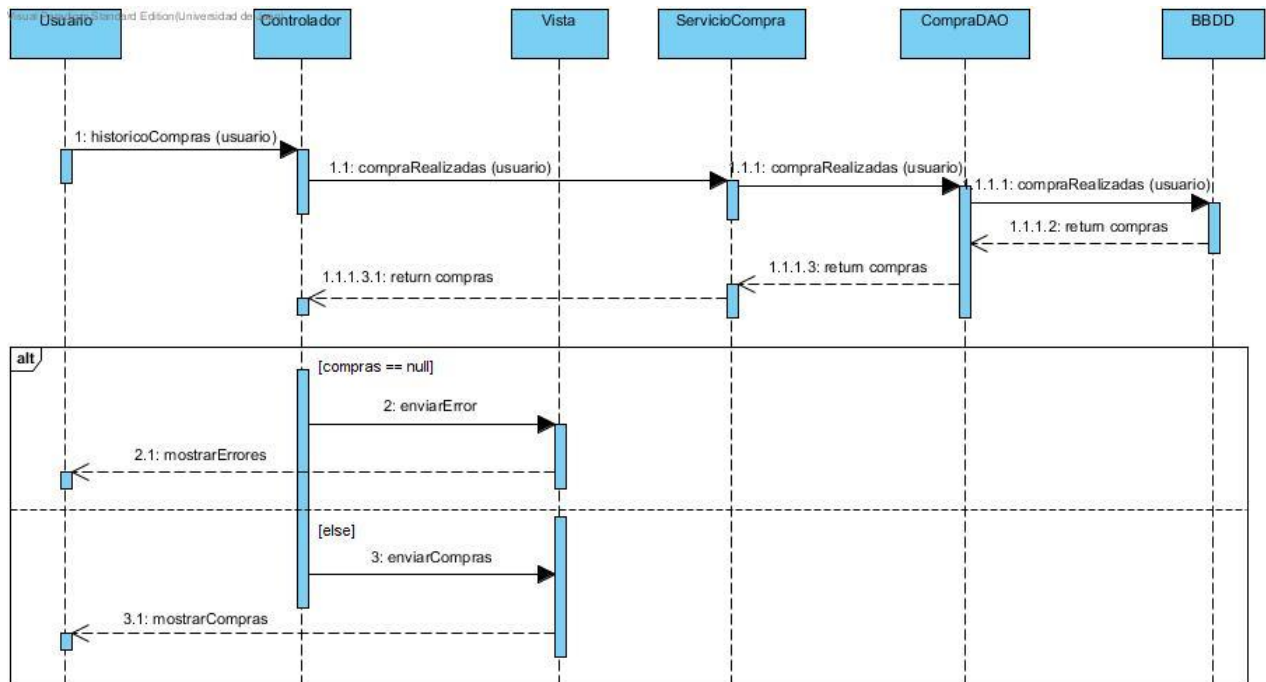


Ilustración 42: DS "Ver compras realizadas"

Una vez logueado, el usuario cliente accede a la opción del menú habilitada para ver el histórico de compras realizadas por este. El sistema busca las compras y las devuelve. En caso de no existir ninguna compra, devolverá un mensaje de error.

- **Diagrama de secuencia: Imprimir entradas**

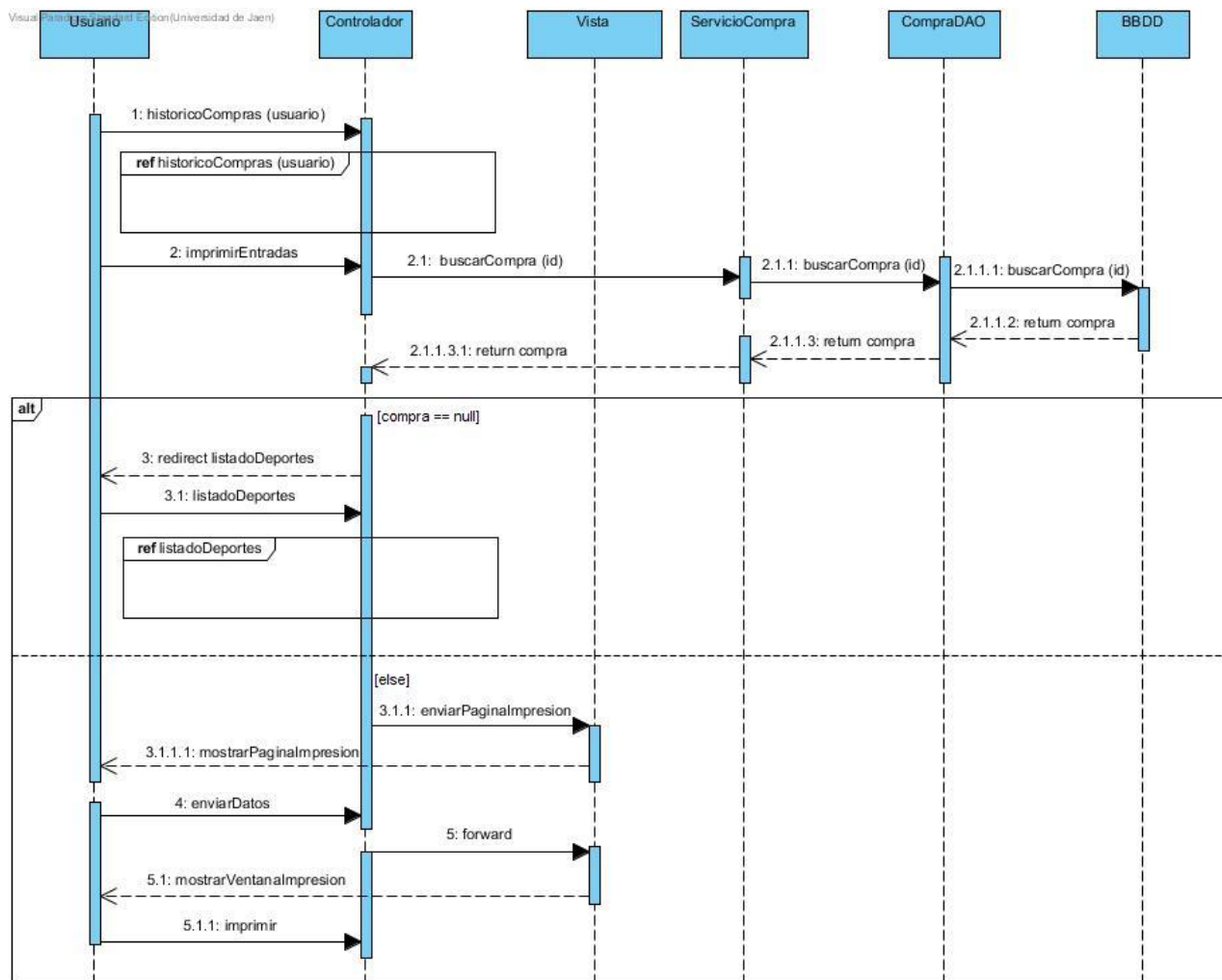


Ilustración 43: DS "Imprimir entradas"

Una vez logueado, el usuario cliente accede a la opción del menú habilitada para ver el histórico de compras realizadas por este, referenciándose dicho proceso en otro diagrama de secuencia (etiqueta ref). A continuación, el cliente selecciona el botón habilitado para imprimir las entradas referentes a una compra determinada. Si la compra no existe, el cliente será redirigido al listado de deportes, ejecutándose esta interacción en un diagrama de secuencia diferente. En caso contrario, el sistema mostrará la página de impresión, en la cual el cliente accionará el botón de imprimir.

3.4. Diseño de la Interfaz

Una vez realizado el análisis del sistema y definida la especificación de los distintos diagramas pertenecientes al diseño, procederemos a detallar el diseño de la interfaz. En esta etapa del diseño del sistema software se define la apariencia visual de la aplicación. El ingeniero del software es el encargado de diseñar una interfaz atractiva para el usuario pero al mismo tiempo tiene que ser inteligible y manejable.

Si el software es difícil de utilizar, si obliga a cometer errores, o causa frustración para conseguirlos objetivos, no será de agrado, independientemente de la potencia informática que demuestre o de la funcionalidad que ofrezca. Dado que la interfaz es la que da forma a la percepción del software por parte del usuario, tiene que estar bien diseñada [2].

A la hora de realizar el diseño de la interfaz de usuario deben tenerse en cuenta la usabilidad y sus principios, tal y como se desarrolla en el capítulo [2.5.2.1](#) de la memoria.

Tras una breve introducción acerca de la importancia del diseño de la interfaz y haber recalcado los principios en los que se apoya dicho diseño, procederemos a realizar el diseño de la interfaz mediante la puesta en marcha de las siguientes tareas:

- Estilo
- Metáforas
- Prototipos
- Notificaciones y mensajes de error

3.4.1. Estilo

El diseño es importante para la usabilidad de la interfaz. Existen reglas de fácil aplicación que pueden ayudar a un buen diseño [18]. Dichas reglas definen lo que se conoce como el estilo de la interfaz, es decir, un conjunto de directrices comunes relacionadas con la apariencia visual que se utilizan en el desarrollo de nuevas “pantallas” de la interfaz de usuario de una aplicación.

Para su definición por escrito, se utilizan las guías de estilo, que son documentos en los que se especifican patrones básicos relacionados con el “aspecto” de la interfaz. Resulta muy útil disponer de estas guías cuando el proceso de diseño va a ser responsabilidad de varias personas, de manera que se mantenga la consistencia del mismo.

Un término importante dentro del estilo es el concepto de layout. Se utiliza para nombrar al esquema de distribución de los elementos dentro de un diseño [26]. Construiremos el layout de nuestra aplicación a partir del modelo de rejilla o grid de Bootstrap, mediante la utilización de clases CSS dentro de las principales etiquetas de un documento HTML, que nos permitirán homogeneizar el diseño.

Aunque el usuario se encuentre en pantallas diferentes, se respetará la estructuración de componentes comunes, tales como menús, botones y barras de navegación, ya que es una forma de familiarizar al usuario con la aplicación. Sin embargo, la interfaz también estará dotada de zonas específicas, como por ejemplo tablas o carruseles de imágenes. A continuación se muestra el layout de la aplicación:

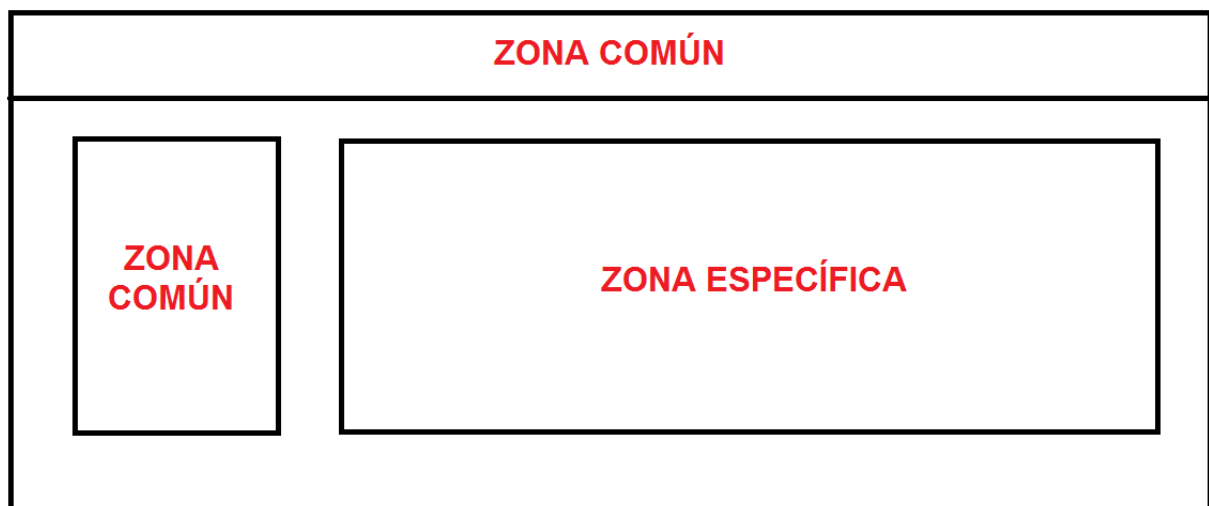


Ilustración 44: Layout

Respecto a los colores, todas las pantallas tendrán la misma tonalidad, respetando la asignación inicial. En caso de que hubiera que destacar algún tipo de información, se emplearán colores llamativos.

Por otro lado, el tipo y el tamaño de la letra también son aspectos importantes que hay que considerar. Se utilizarán los existentes por defecto en Bootstrap, ya que son idóneos para la legibilidad de la interfaz de la aplicación.

3.4.2. Metáforas

Una metáfora consiste en una asociación entre elementos que comparten alguna similitud de significado para sustituir a uno por el otro en una misma estructura [27]. Tienen un papel dominante en el diseño de interfaces, ya que estos elementos describen visualmente las tareas que la interfaz permite desarrollar.

Para que una metáfora alcance el objetivo deseado, debe existir una base cultural similar entre el desarrollador de la aplicación y el usuario final de ésta, ya que en caso contrario puede llevar a errores en la interacción.

Las metáforas utilizadas en el sistema pertenecen al conjunto de iconos predefinidos en el framework Bootstrap [28]. A continuación se muestran en una tabla las metáforas que se utilizarán en nuestra aplicación:

Icono	Nombre	Descripción
	Editar	Icono representativo de la acción para “editar una entidad”
	Eliminar	Icono representativo de la acción para “eliminar una entidad”
	Nuevo	Icono representativo de la acción para “dar de alta un usuario”. El icono para el resto de entidades se realizará de forma equivalente
	Visualiza	Icono representativo de la acción para “visualizar una entidad”
	Puesta en Venta	Icono representativo de la acción para “poner en venta”
Buscar eventos 	Buscar eventos	Icono representativo de la acción para “buscar eventos”

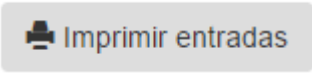
	Comprar	Icono representativo de la acción para “comprar entradas”
	Cancelar Compra	Icono representativo de la acción para “cancelar una compra”
	PayPal	Icono representativo de la acción para “pagar una compra”
	Imprimir entradas	Icono representativo de la acción para “imprimir una entrada”

Tabla XXX: Metáforas

3.4.3. Prototipos de interfaz

La mejor herramienta para definir la estructura de la interfaz con el usuario es el prototipo de papel. Este tipo de prototipo se basa en la utilización de instrumentos que sirvan para describir el diseño en un papel, tales como un lápiz o un bolígrafo. Este sistema nos permite una gran velocidad y flexibilidad [18].

Una de las principales ventajas de la utilización de esta técnica es el coste reducido que presenta y teniendo en cuenta que el principal objetivo es el análisis de la calidad del diseño, es absurdo perder tiempo y dinero utilizando una herramienta especializada.

La fusión de los distintos prototipos desarrollados proporciona una idea de lo que será el sistema, no el diseño final de éste, por lo que serán susceptibles de cambio durante la implementación.

Para poder reproducir las interacciones efectuadas con el sistema, se utilizará una hoja de papel para cada escenario.

El primer prototipo hace referencia a la pantalla principal de la aplicación, constituyendo la raíz del sitio web para usuarios sin identificar. A partir de ella el usuario puede o bien registrarse o identificarse:

Ilustración 45: Prototipo 1 "Pantalla principal"

El siguiente prototipo muestra la pantalla de inicio del administrador:

Ilustración 46: Prototipo 2 "Menú inicio administrador"

Si el administrador selecciona la opción “Gestión de usuarios” situada en el menú, el sistema muestra el listado de usuarios, con la funcionalidad ilustrada en la ilustración siguiente:

Administrador >>

Menú							
Gestión de Usuarios							
Gestión de Deportes							
Gestión de Competiciones							
Gestión de Participantes							
Gestión de Eventos							
Gestión de Entradas							
Gestión de Compras							

Listado de Usuarios

Id	Nombre	Correo	Ciudad	Usuario	Clave	Edad	Rol	
1	Administrador	admin@joven.es	Jaén	admin	admin	51	ADMIN	Visualizar Editar Eliminar
2	Ramón Bailén Sánchez	rb@joven.es	Jaén	rb00003	ramon	51	USUARIO	Visualizar Editar Eliminar
3	—	—	—	—	—	—	—	Visualizar Editar Eliminar

Ilustración 47: Prototipo 3 “Listado de usuarios”

Si el administrador desea dar de alta un nuevo usuario, tendrá que rellenar los campos de la imagen inferior:

Administrador >>

Menú
Gestión de Usuarios
Gestión de Deportes
Gestión de Competiciones
Gestión de Participantes
Gestión de Eventos
Gestión de Entradas
Gestión de Compras

Alta de Usuario

Nombre:

Correo:

Ciudad:

Usuario:

Clave:

Rol:

☐ Admin

☐ User

Ilustración 48: Prototipo 4 “Alta de usuario”

Si el administrador desea modificar un usuario, accionará el botón de editar situado en el listado, mostrando el sistema la siguiente pantalla:

Administrador >>

Menú
Gestión de Usuarios
Gestión de Deportes
Gestión de Competiciones
Gestión de Participantes
Gestión de Eventos
Gestión de Entradas
Gestión de Compras

Editar Usuario

Id Cliente: 1

Nombre:

Correo:

Ciudad:

Usuario:

Clave:

Ilustración 49: Prototipo 5 "Editar usuario"

Accediendo a la gestión de competiciones y al botón habilitado para dar de alta una nueva competición, el administrador podrá realizar dicho proceso:

Administrador >>

Menú
Gestión de Usuarios
Gestión de Deportes
Gestión de Competiciones
Gestión de Participantes
Gestión de Eventos
Gestión de Entradas
Gestión de Compras

Alta de Competición

Nombre:

Deporte:

Ilustración 50: Prototipo 6 "Alta de competición"

Para editar el participante de un deporte, el administrador modificará el formulario de la siguiente ilustración:

Administrador >> [Salir](#)

Menú
Gestión de Usuarios
Gestión de Deportes
Gestión de Competiciones
Gestión de Participantes
Gestión de Eventos
Gestión de Entradas
Gestión de Compras

Editar Participante

Id. Participante: 3

Nombre:

Deporte:

 ▼

Ilustración 51: Prototipo 7 "Editar participante"

El administrador accederá a la gestión de eventos y accionará el botón habilitado para el alta. A continuación rellenará el siguiente formulario:

Administrador >> [Salir](#)

Menú
Gestión de Usuarios
Gestión de Deportes
Gestión de Competiciones
Gestión de Participantes
Gestión de Eventos
Gestión de Entradas
Gestión de Compras

Alta de Evento

Nombre:

Fecha:

Hora:

Lugar:

Ciudad:

Deporte:

 ▼

Competición:

 ▼

Participantes:

☐ Real Madrid

☐ Barcelona

☐ Rafael Nadal

[Guardar](#) [Restablecer](#) [Volver](#)

Ilustración 52: Prototipo 8 "Alta de evento"

Para visualizar una compra, el administrador entrará en la gestión de compras y seleccionará la opción existente para ver una compra determinada. El sistema mostrará la siguiente pantalla:

Administrador >> Salir

Menú
Gestión de Usuarios
Gestión de Deportes
Gestión de Competiciones
Gestión de Participantes
Gestión de Eventos
Gestión de Entradas
Gestión de Compras

Visualizar Compra

Id Compra: 21

Usuario: Ramón Bailén Sánchez

Fecha de Compra: 15-01-2015

Fecha de Pago: 15-01-2015

Entradas:

Entrada 1		23 €
Entrada 2		23 €

Total: 46 €

Volver

Ilustración 53: Prototipo 9 “Visualizar compra”

La imagen siguiente muestra la pantalla de inicio del cliente:

Ramón Bailén Sánchez >> Salir

Menú
Datos Personales
Ver Eventos

Bienvenido/a a la aplicación de venta de localidades deportivas

Ilustración 54: Prototipo 10 “Menú inicio cliente”

Si el cliente selecciona la opción “Datos personales” del menú, el sistema presenta la ilustración inferior:

Hand-drawn prototype of the "Visualizar Datos Personales" screen. The interface includes a menu on the left with options: "Menú", "Datos Personales", and "Ver Eventos". The main content area displays the title "Visualizar Datos Personales" followed by the following information:

- Id. Cliente: 2
- Nombre: Ramón Bailén Sánchez
- Correo: rb@ujaen.es
- Ciudad: Jaén
- Usuario: contraseña
- Contraseña: raman

At the bottom of the main area are three buttons: "Editar", "Eliminar", and "Volver". The top right corner shows the user name "Ramón Bailén Sánchez >>" and a "Salir" button.

Ilustración 55: Prototipo 11 “Visualizar datos personales cliente”

A través de la opción “Ver Eventos” del menú, el cliente visualiza los deportes habilitados en el sistema que contienen eventos con entradas a la venta:

Hand-drawn prototype of the "Ver deportes eventos" screen. The interface includes a menu on the left with options: "Menú", "Datos Personales", and "Ver Eventos". The main content area displays the title "Entradas Deportes" above a box labeled "Imagen". To the right of the image box are three horizontal lines representing event entries. Below the image box is a list of categories:

- Categorías
 - Fútbol
 - Tenis
 - Baloncesto
 - Golf

The top right corner shows the user name "Ramón Bailén Sánchez >>" and a "Salir" button.

Ilustración 56: Prototipo 12 “Ver deportes eventos”

Cuando el cliente selecciona un deporte determinado, el sistema muestra 3 categorías, con el objetivo de facilitar la búsqueda de eventos relacionados con dicho deporte:

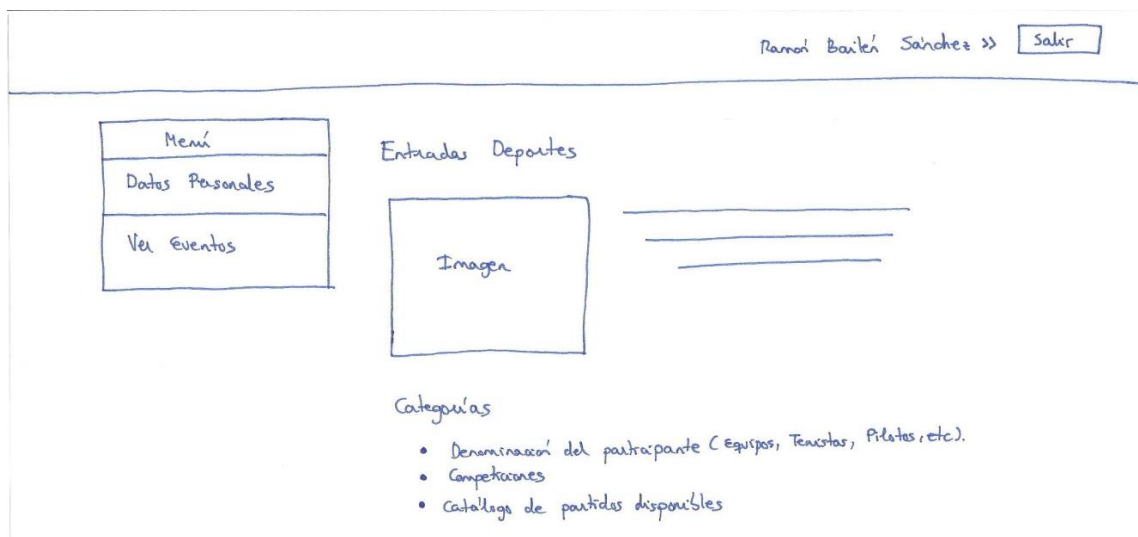


Ilustración 57: Prototipo 13 “Categorías deporte”

Una vez que el cliente ha seleccionado una categoría, el sistema presenta los eventos relacionados con ésta:

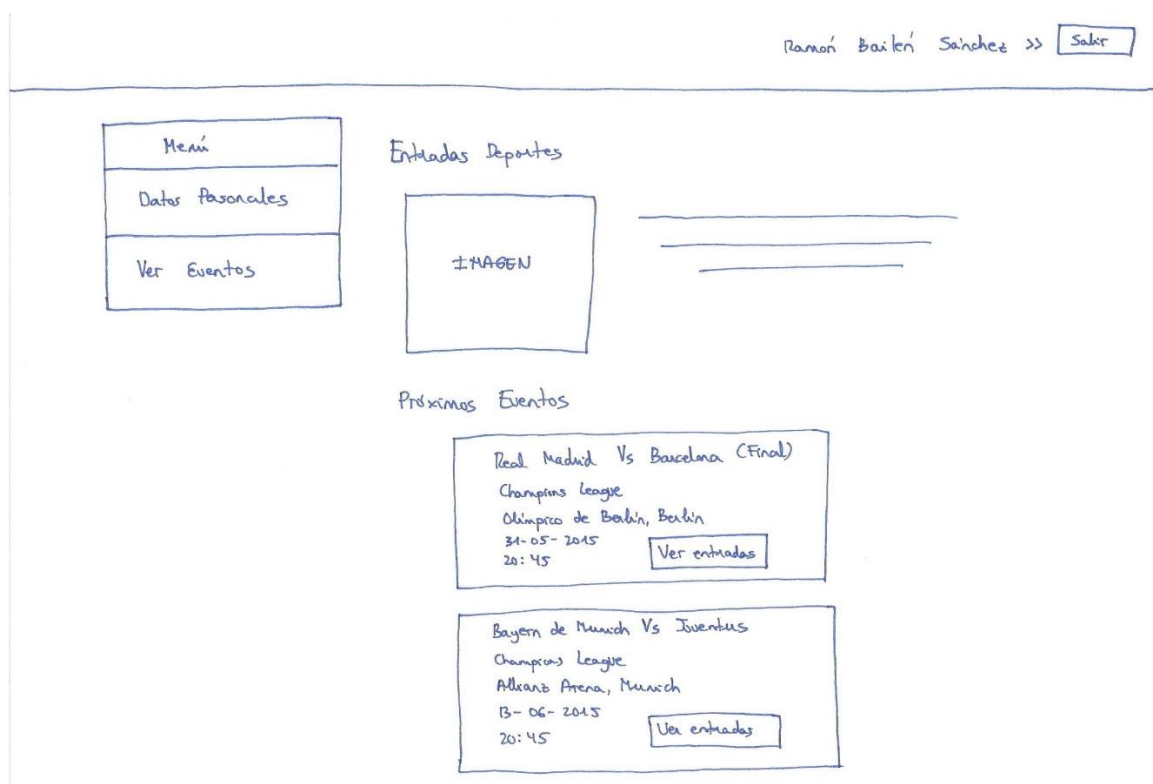


Ilustración 58: Prototipo 14 “Eventos deporte”

El cliente accede a visualizar las entradas de un evento concreto, con la posibilidad de comprar algunas de ellas:

Ramón Bailén Sánchez >> Salir

Menú
Datos Personales
Ver Eventos

Entradas Deportes

Seleccione las entradas que desea comprar

☐ Entrada 1 para evento : Real Madrid Vs Barcelona (Final) 30€

☐ Entrada 2 para evento : Real Madrid Vs Barcelona (Final) 45€

☐ Entrada 3 para evento : Real Madrid Vs Barcelona (Final) 25€

Comprar

Ilustración 59: Prototipo 15 “Comprar entradas”

Tras realizar una compra, el sistema mostrará al cliente la información de la misma y habilitará la funcionalidad de cancelación o pago.

Ramón Bailén Sánchez >> Salir

Menú
Datos Personales
Ver Eventos

Entradas Deportes

Confirmación de Compra

Compra nº: 28

Cliente: Ramón Bailén Sánchez

Fecha de Compra: 26-02-2015

Total: 70€

Entradas:

Entrada 2 para evento: Real Madrid Vs Barcelona (Final) Deporte: Fútbol Precio: 45€
Entrada 3 para evento: Real Madrid Vs Barcelona (Final) Deporte: Fútbol Precio: 25€

Cancelar Compra Pagar

Ilustración 60: Prototipo 16 “Confirmación de compra”

3.4.4. Notificaciones y mensajes de error

Una vez conocido el funcionamiento del sistema y tras diseñar los diferentes prototipos de la interfaz, es necesario informar al usuario de determinadas circunstancias y errores. Por lo tanto, el siguiente paso consiste en definir tanto las notificaciones como los mensajes de error que se muestran al usuario en su interacción con la aplicación.

3.4.4.1. Notificaciones

Las notificaciones permiten un incremento de la confianza por parte del usuario tras la realización de alguna operación. Generalmente, deben ser breves y claras, con el objetivo de que el usuario las lea y no las descarte, además de contener colores que llamen la atención de éste.

Algunos mensajes de alerta utilizados en el sistema son los siguientes:

- **Registro**

Tras finalizar el registro en la plataforma de un nuevo usuario, el sistema muestra el mensaje de confirmación siguiente:

Se ha registrado correctamente

Ilustración 61: Notificación "Registro correcto"

- **Eliminar**

Si un usuario del sistema desea eliminar algún tipo de información, el sistema presenta una notificación con el objetivo de verificar la realización de dicha operación:

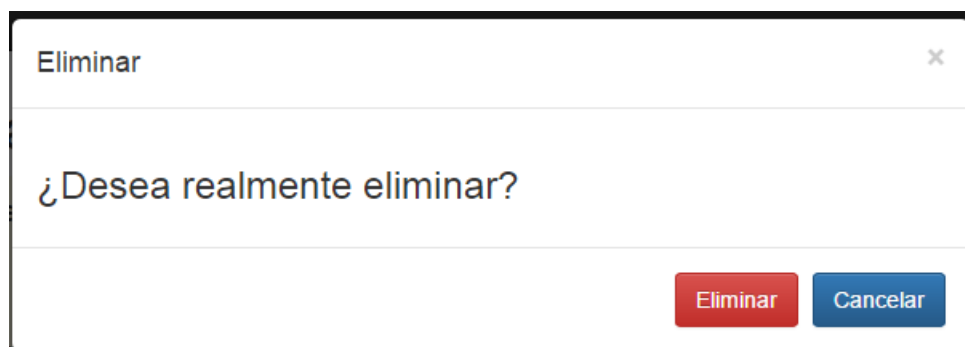


Ilustración 62: Notificación "Eliminar"

- **Salir**

Si un usuario desea salir del sistema, el sistema muestra una alerta indicando la confirmación de la operación:



Ilustración 63: Notificación "Salir"

- **Cancelar compra**

El sistema expone una confirmación de anulación de una operación de compra por si el cliente desea cancelarla antes de pagarla:

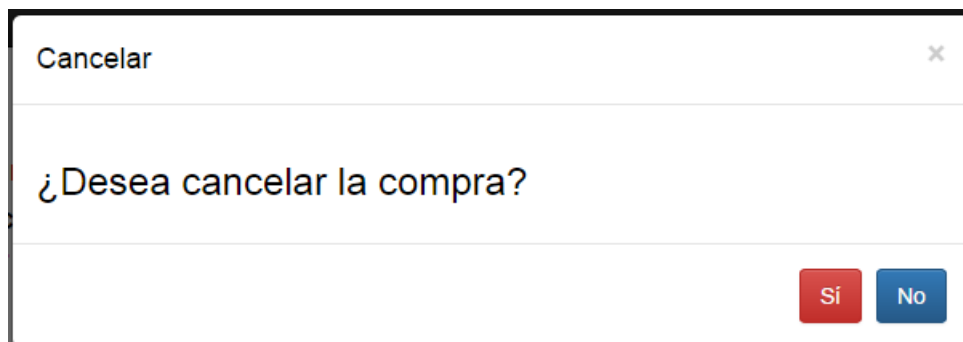


Ilustración 64: Notificación "Cancelar compra"

- **Existencia de deportes**

Si el usuario administrador intenta crear una nueva competición, un nuevo participante o un nuevo evento sin existir deportes previamente, el sistema muestra las siguientes notificaciones, respectivamente:

Para agregar competiciones, debe previamente disponer de deportes

Ilustración 65: Notificación "Alta de competición"

Para agregar participantes, debe previamente disponer de deportes

Ilustración 66: Notificación "Alta de participante"

Para agregar un evento, debe previamente disponer de deportes

Ilustración 67: Notificación "Alta de evento"

- **Alta de entrada**

El sistema muestra una notificación si el usuario administrador intenta crear entradas sin la presencia de eventos:

Para agregar una entrada, debe previamente disponer de eventos

Ilustración 68: Notificación "Alta entrada"

3.4.4.2. Mensajes de error

Los mensajes de error son de vital importancia para la usabilidad de la aplicación, ya que representan la vía por la que el sistema comunica al usuario que se ha producido un error durante la interacción.

Al igual que las notificaciones, los mensajes de error deben ser breves y específicos, de manera que el usuario entienda el motivo que ha llevado al sistema a una situación errónea. Igualmente, debe establecerse una apariencia adecuada que facilite la comprensión y localización de éstos.

Teniendo en cuenta las características descritas, pueden aparecer los siguientes mensajes de error:

- **Identificación**

Si un usuario no introduce las credenciales correctas (nombre de usuario y contraseña) para acceder a la aplicación, el sistema muestra el mensaje de error siguiente:

Los datos suministrados no son válidos. Vuelva a introducirlos

Ilustración 69: Mensaje de error "Identificación"

- **Formulario**

Los campos de todos los formularios están validados. Las ilustraciones siguientes son ejemplos representativos de errores tras rellenar algunos campos de los formularios existentes en la aplicación:

Nombre:

Campo requerido

Inserta un nombre válido entre 5 y 30 caracteres

Ilustración 70: Mensaje de error "Formulario 1"

Deporte:

Seleccione un deporte

Ilustración 71: Mensaje de error "Formulario 2"

Usuario:

Ya existe un usuario con ese nombre de usuario

Ilustración 72: Mensaje de error "Formulario 3"

Participantes:

- ☐ Real Madrid C.F
- ☐ F.C Barcelona
- ☐ Roger Federer

Seleccione al menos dos participantes

Ilustración 73: Mensaje de error "Formulario 4"

Hora:

Introduzca la hora

Ilustración 74: Mensaje de error "Formulario 5"

Fecha:

Introduzca la fecha

Ilustración 75: Mensaje de error "Formulario 6"

- **Búsqueda de eventos**

Un cliente que utilice el buscador para localizar un evento e introduzca un criterio de búsqueda erróneo recibe el siguiente mensaje por parte del sistema:

NO HAY EVENTOS DISPONIBLES CON ESE CRITERIO DE BÚSQUEDA

Ilustración 76: Mensaje de error "Búsqueda de eventos"

- **Selección de entradas**

El sistema muestra el siguiente mensaje a los clientes que pulsen el botón comprar sin haber seleccionado previamente ninguna entrada:

Seleccione las entradas que desee

Ilustración 77: Mensaje de error "Selección de entradas"

4. Implementación

La implementación es la actividad final de la ingeniería del software. En ella, el modelo obtenido en las fases anteriores se traduce a código fuente. Dichas fases previas se han desarrollado en los capítulos 2 ([análisis del sistema](#)) y 3 ([diseño del sistema](#)) respectivamente.

A continuación, se detallan algunos aspectos relacionados con esta etapa, como son el tipo de arquitectura de la aplicación, información destacada acerca de la implementación y por último la realización de pruebas.

4.1. Arquitectura

Este proyecto se basa en el desarrollo de una aplicación web con una arquitectura cliente/servidor. El funcionamiento de este tipo de arquitectura es el siguiente: a través de un software cliente (navegador web), el usuario realiza peticiones a un servidor, el cual aloja la aplicación. Tras la recepción de la solicitud, el servidor la procesa y envía una respuesta al cliente.

Para ello, se apoya en la utilización de las siguientes tecnologías:

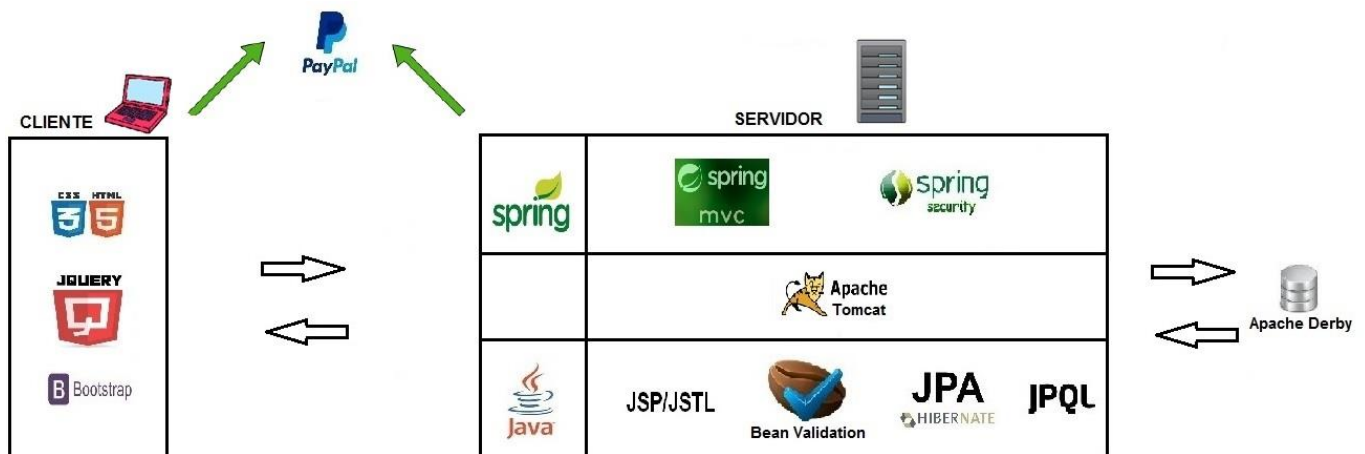


Ilustración 78: Arquitectura

Como puede observarse, se han separado las diferentes tecnologías según los componentes típicos de una aplicación web centrada en el servidor. Las funciones de cliente y servidor pueden estar en plataformas separadas o en la misma. El servidor funciona como sistema gestor de base de datos (SGBD). Ambos componentes interaccionan con Pasarela Integral de PayPal.

Para el desarrollo del proyecto se ha aplicado el patrón de arquitectura MVC [24] por medio de uno de los frameworks más conocido y demandado de la plataforma Java: Spring Framework [11]. Tras el análisis realizado en el [capítulo 2.3](#), se llegó a la conclusión de que JSF [13] está más orientado a aplicaciones de interfaz enriquecida que requieren una mayor interacción con el cliente. Por el contrario, para el desarrollo de aplicaciones web convencionales con un enfoque MVC puro, utilizar Spring MVC es una forma dinámica de implementar aplicaciones Java hoy en día, dentro del paradigma de la programación orientación a objetos.

El sistema presenta dos tipos de usuarios: por un lado, se encuentran los usuarios con perfil de administración, encargados del desarrollo y mantenimiento del mismo y por otro los clientes potenciales que utilizarán los servicios de la aplicación para la adquisición de las entradas. Para ello, ambos tipos tendrán que estar registrados en el sistema.

La aplicación hará uso de páginas JSPs [29] para crear las vistas, presentando la estructura básica de un documento HTML [30], que incluye hojas de estilo CSS [31] y etiquetas JSTL [32], además de entidades JPA [33] que representan el modelo, todo ello sobre un servidor web Tomcat, elegido debido a su sencillez y extenso uso. En cuanto a la base de datos, se hará uso de una base de datos Derby, ya que tiene un tamaño reducido y al implementar el proyecto con tecnología Java hace que la aplicación sea portable.

En relación al diseño de una interfaz web ágil y flexible de cara al cliente, se ha tomado la decisión de emplear Bootstrap [15], el cual proporciona un resultado rápido y muy profesional, además de ofrecer componentes JavaScript, tales como modales o menú desplegados, por medio de la utilización de la librería jQuery [34], la cual posee grandes volúmenes de documentación y un gran equipo de desarrolladores a cargo de su actualización.

Por otro lado, debido al auge de los dispositivos móviles, a la hora de desarrollar una aplicación web es interesante pensar en el diseño y funcionamiento de ésta en móviles y tablets. De ahí la elección de un framework cuya principal característica es el uso de la técnica responsive, que permite la correcta visualización de una interfaz web en dispositivos diferentes.

4.2. Detalles sobre implementación

En los siguientes apartados se definen las nociones que proporcionan mayor relevancia sobre la implementación de la aplicación web.

4.2.1. Entorno de desarrollo

En este apartado se determinan las herramientas y equipos utilizados para el desarrollo de la aplicación.

En este proyecto se va a desarrollar una aplicación web utilizando la tecnología Java y con acceso a una base de datos que trabaja a nivel local. Debido a que las funciones MVC se dividen entre cliente y servidor, se diferenciará entre los requerimientos del equipo necesarios para el cliente y los necesarios para el servidor.

Los requisitos imprescindibles para el cliente son muy sencillos. Solo le hace falta un equipo conectado a Internet y tener instalado un navegador que permita la correcta visualización e interacción con la aplicación web (se recomienda la utilización Google Chrome).

Análogamente, para que la aplicación funcione correctamente, los requerimientos del equipo informático en el que se vaya a utilizar se dividen en dos tipos: requerimientos de hardware y requerimientos de software.

1) Hardware

- Monitor: debe soportar una resolución de 1024x768 y superiores.
- Memoria: el equipo debe proporcionar la memoria RAM suficiente para ejecutar la funcionalidad ofrecida por la aplicación. Se recomienda disponer de 2GB de RAM.
- Disco duro: la capacidad de almacenamiento del equipo debe ser idónea como para guardar la base de datos con la que trabaja la aplicación y tolerar la ejecución de cualquier operación.
- Velocidad: es imprescindible para el buen rendimiento. El equipo tiene que ser lo suficientemente rápido en la ejecución de la aplicación.

- Tarjeta gráfica: cualquier equipo actual contiene una tarjeta gráfica con características adecuadas para la utilización de la aplicación.
- Impresora: para la impresión de las entradas, se recomendaría disponer de una impresora básica, pero no es necesario establecer un requerimiento en este aspecto.

2) Software

- Sistema operativo: Windows 8.1.
- Sistema Gestor de Bases de Datos: es necesario que el equipo disponga del sistema gestor de bases de datos Apache Derby.
- Máquina virtual de Java (Java Virtual Machine, JVM): debe disponer de ella para ejecutar la aplicación.
- Navegador: como se ha comentado anteriormente, se requiere de la disponibilidad de un navegador web actual y se recomienda la utilización de Google Chrome.

La herramienta de desarrollo empleada para la implementación del proyecto ha sido Netbeans en su versión 8.0.2. Netbeans es un entorno de desarrollo integrado (IDE) libre y gratuito, creado principalmente para el lenguaje de programación Java, que permite a los programadores escribir, compilar, depurar y ejecutar programas.

El funcionamiento de Netbeans no es a nivel de archivo, sino a nivel de proyecto. Físicamente, un proyecto no es otra cosa que un directorio que tiene una organización especial, la cual incluye todos los recursos necesarios para la construcción de un programa, como pueden ser los archivos con el código o algunas bibliotecas externas.

Además del Netbeans, se requiere disponer del contenedor web Apache Tomcat, junto con una base de datos Derby que acepte conexiones en el puerto 1527.

Para crear el proyecto donde hemos trabajado en el desarrollo de la aplicación web, se han seguido los siguientes pasos:

- 1) Abrir Netbeans.
- 2) Hacer click en el menú Archivo y seleccionar Proyecto Nuevo.
- 3) Seleccionar la categoría Java Web.
- 4) Escogemos el proyecto Web Application.
- 5) Introducir el Nombre del Proyecto ("Localidades").
- 6) Optamos por Apache Tomcat 8.0.9.0 como Servidor.
- 7) Introducir la ruta de contexto de la aplicación web ("/localidades").
- 8) Seleccionamos el framework que vamos a usar en la aplicación ("Spring Web MVC").
- 9) Accionamos el botón Terminar.

4.2.2. Lenguajes de programación

La implementación de la aplicación se ha llevado a cabo mediante la utilización del lenguaje de programación Java, además del lenguaje de consulta orientado a objetos JPQL a través el empleo de la API de persistencia de Java JPA.

Java es un lenguaje de programación orientado a objetos. El principal objetivo de este lenguaje es permitir que los desarrolladores de aplicaciones escriban el programa una vez y puedan ejecutarlo en cualquier dispositivo sin tener que ser recompilado. Un aspecto importante de este lenguaje es la máquina virtual de Java (Java Virtual Machine), que ayuda a ejecutar las aplicaciones Java sin importar la arquitectura de la computadora, convirtiendo el bytecode a código nativo del dispositivo final.

JPQL (Java Persistence Query Language) es un lenguaje de consulta orientado a objetos independiente de la plataforma definido como parte de la especificación de la API de persistencia de Java (JPA). Es usado para hacer consultas sobre entidades almacenadas en una base de datos relacional. Está inspirado por SQL y la sintaxis de sus consultas se asemeja a las de este lenguaje, pero opera con objetos entidad de JPA en lugar de hacerlo directamente con las tablas de la base de datos. Soporta tanto consultas de recuperación de objetos (Select), como de actualización (Update) y borrado (Delete) [33].

JPA (Java Persistence API) es la API de persistencia desarrollada para la plataforma Java. El objetivo que persigue es no perder las ventajas de la orientación

a objetos al interactuar con una base de datos y permitir la utilización de POJOs (clase simple que no extiende ni implementa nada en especial). El mapeo objeto-relacional (ORM), es decir, la relación entre entidades Java y tablas de la base de datos, se realiza mediante anotaciones en las propias clases de entidad [33]. Seguidamente se ilustra un ejemplo:

```
@Entity
@Table(name = "entradas")
public class Entrada implements Serializable {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private int idEntrada;

    private String descripcion;

    private double precio;

    private boolean comprada;

    @OneToOne
    @JoinColumn(name = "idEvento")
    private Evento evento;

}
```

Ilustración 79: JPA Anotaciones

Apoyándonos en la especificación de JPA, hemos utilizado Hibernate para construir una capa de persistencia. La integración de JPA con implementación de Hibernate en nuestro sistema se lleva a cabo en el fichero *applicationContext.xml* ubicado en el directorio */web/WEB-INF* del proyecto:

```
<bean id="jpaAdapter" class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
    <property name="database" value="DERBY" />
    <property name="databasePlatform" value="org.hibernate.dialect.DerbyTenSevenDialect" />
    <property name="generateDdl" value="true" />
</bean>
```

Ilustración 80: JPA Integración

Para crear las tablas de la base de datos de manera automática, en primer lugar se indica la ubicación del paquete donde se encuentran las clases persistentes anotadas con `@Entity`, a continuación se inyecta la base de datos y el adaptador específico para el ORM empleado (Hibernate) y por último se activa la generación automática de esquemas. La ilustración siguiente muestra el proceso descrito:

```
<bean id="emf" class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
  <property name="packagesToScan" value="es.ujaen.tfg.modelos" />
  <property name="dataSource" ref="dataSource" />
  <property name="jpaVendorAdapter" ref="jpaAdapter" />

  <!-- Generación de tablas -->
  <property name="jpaProperties">
    <props>
      <prop key="hibernate.hbm2ddl.auto">update</prop>
    </props>
  </property>
</bean>
```

Ilustración 81: Creación de las tablas

4.2.3. Tecnologías utilizadas

Como ya se ha especificado anteriormente, para llevar a cabo la implementación del proyecto se ha utilizado **Spring Framework**. Es una herramienta para el desarrollo de aplicaciones web en Java que permite una división limpia entre el modelo, la vista y el controlador (MVC), proporcionando un enfoque ligero y modular. Además es flexible, empleando solamente la funcionalidad necesaria, evitando tener la carga de los módulos que no utiliza.

Se ha añadido la versión 4.0.1 de manera automática mediante la creación del proyecto, tal y como se ha descrito en el apartado del [entorno de desarrollo](#).

Spring posibilita el uso de anotaciones, que nos permiten la especificación de dependencias de manera flexible. Actualmente, es el tipo de configuración más usado, por delante del empleo del archivo de configuración XML, ya que supone un ahorro de código considerable. Una anotación es una línea de código insertada en el código fuente cuya función principal es indicar el comportamiento del software.

Además, el framework soporta la programación orientada a aspectos. Es un enfoque de programación cuyo objetivo principal es la separación de funcionalidades dentro del sistema. Por un lado, se encuentran las funcionalidades comunes de la aplicación y por otro las funcionalidades propias de cada clase. En nuestra aplicación, tiene utilidad dentro del soporte de transacciones declarativo y en el cacheado de operaciones.

Se denominada transacción a un conjunto de operaciones todo o nada. La implementación de una operación en la capa de lógica de negocio puede implicar una serie de pasos delicados. El fallo de cualquiera de estos pasos debe deshacer

las acciones realizadas hasta el estado estable anterior a la operación [12]. Spring provee soporte de transacciones programáticas y declarativas. El primer tipo es la forma tradicional de crear una transacción. Permite definir los límites de la transacción en el código. Sin embargo, el soporte de transacciones declarativo es la forma más limpia y potente de gestionar las transacciones en una aplicación. Se ayuda de la programación orientada a aspectos (AOP) para la inyección del soporte de transacciones [12]. Su implementación, integrada en los DAOs, puede realizarse mediante la anotación `@Transactional`. Primero es necesario activar el soporte de transacciones por anotación en el fichero de contexto de la aplicación [12], tal y como muestra la siguiente ilustración:

```
<tx:annotation-driven transaction-manager="transactionManager" proxy-target-class="true"/>
<context:component-scan base-package="es.ujaen.tfg.dao"/>
<bean id="transactionManager" class="org.springframework.orm.jpa.JpaTransactionManager">
    <property name="entityManagerFactory" ref="emf"/>
</bean>
```

A continuación se muestra un ejemplo:

```
@Transactional(propagation = Propagation.REQUIRED, readOnly = false,
                rollbackFor = es.ujaen.tfg.excepciones.UsuarioErrorInsertar.class)
public void insertar(Usuario usuario) throws UsuarioErrorInsertar {
    try {
        em.persist(usuario);
    } catch (Exception e) {
        throw new UsuarioErrorInsertar();
    }
}
```

Ilustración 82: Transacción

Respecto al cacheado, consiste en una optimización de operaciones cuyo tiempo de procesamiento es largo y con acceso frecuente, lo que producirá una mejora en el rendimiento de la aplicación. Usa la programación orientada a aspectos para interceptar a llamada a la operación y devolver el resultado directamente si los datos se encuentran en la caché [12].

El soporte de caché utilizado se corresponde con la implementación basada en EHCACHE. Es una caché de alto rendimiento, escalable, persistente y distribuida [12]. Se ha creado un fichero denominado *ehCache.xml* dentro del directorio *WEB-INF* con la configuración básica. Seguidamente se adjunta un sencillo ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<ehcache xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://ehcache.org/ehcache.xsd">

    <cache name="usuarios" maxElementsInMemory="10000" eternal="true"
overflowToDisk="false" />
</ehcache>
```

Ilustración 83: EHCACHE

El fichero anterior se incluye dentro del archivo `/WEB-INF/applicationContext.xml`:

```
<bean id="cacheManager" class="org.springframework.cache.ehcache.EhCacheCacheManager">
    <property name="cacheManager" ref="ehcache"/>
</bean>
<bean id="ehcache" class="org.springframework.cache.ehcache.EhCacheManagerFactoryBean">
    <property name="configLocation" value="/WEB-INF/ehCache.xml" />
</bean>
```

Ilustración 84: EHCACHE Import

Al igual que las transacciones, el cacheado puede realizarse mediante anotaciones (`@Cacheable`), indicando que los resultados de la operación serán cacheados. La siguiente llamada con los mismos argumentos es resuelta por la caché [12]. A continuación se muestra un ejemplo:

```
@Cacheable(value = "usuarios", key = "#idUsuario")
public Usuario buscar(int idUsuario) {
    return em.find(Usuario.class, idUsuario);
}
```

Ilustración 85: Cacheado

Una operación de escritura puede modificar previamente cacheados [12]. Se ha empleado la anotación `@CacheEvict` para invalidar parcialmente una caché. En la siguiente ilustración se presenta un ejemplo de su utilización:

```
@CacheEvict(value = "usuarios", key = "#usuario.getIdUsuario()")
@Transactional(propagation = Propagation.REQUIRED, readOnly = false,
    rollbackFor = es.ujaen.tfg.excepciones.UsuarioErrorActualizar.class)
public void actualizar(Usuario usuario) throws UsuarioErrorActualizar {
    try {
        em.merge(usuario);
    } catch (Exception e) {
        throw new UsuarioErrorActualizar();
    }
}
```

Ilustración 86: CacheEvict

Spring Framework está diseñado entorno a un controlador principal, denominado *DispatcherServlet*, que gestiona todas las peticiones. Este controlador trata de localizar los controladores que atiendan cada petición utilizando la anotación `@RequestMapping`. Los controladores son clases que usan `@Controller`. Sus principales funciones son: interactuar con el modelo, enviar información a la vista por medio de objetos `ModelMap` e identificar cuál de ellas generará la salida HTML [10].

Spring realiza la configuración de los componentes de la aplicación y gestiona el ciclo de vida de los objetos mediante inyección de dependencias (`@Autowired`) [10]. Una clase ya no crea los objetos que necesita, sino que los recibe de otra clase.

A continuación se muestra la estructura básica de un controlador:

```
@Controller
@RequestMapping("/usuarios")
@Component
public class ControladorUsuarios {

    @Autowired
    private ServicioUsuario servicio;

    public ControladorUsuarios() {

    }

    /**
     *
     * @param model
     * @return
     */
    @RequestMapping(value = {"", "/listadoClientes"}, method = RequestMethod.GET)
    public String listadoUsuarios(ModelMap model) {
        List<Usuario> clientes = servicio.listadoUsuarios();
        model.addAttribute("clientes", clientes);
        return "admin/usuarios/listado";
    }

}
```

Ilustración 87: Ejemplo controlador

El método “listadoUsuarios” utiliza el servicio para obtener todos los usuarios del sistema. Esta información es pasada a la vista a través de un objeto de tipo `ModelMap`. Finalmente, se devuelve la vista adecuada.

Para poder utilizar las anotaciones de Spring, hay que añadir la ruta de clases del proyecto en el fichero `/WEB-INF/dispatcher-servlet.xml`:

```
<mvc:annotation-driven />
<context:component-scan base-package="es.ujaen.tfg" />
```

Ilustración 88: Configuración anotaciones

Adicionalmente, se ha usado **Bootstrap**, proporcionando la realización de un diseño responsive que se ajusta a las características del dispositivo desde el que se ejecuta la aplicación. La forma más sencilla de utilizarlo es descargando el código CSS y JavaScript compilado. Para ello, accedemos a <http://getbootstrap.com/> y pulsamos el botón Download Bootstrap. Ha sido descargada la versión 3.3.1. Después de descomprimir el fichero descargado, creamos una carpeta dentro del directorio “web” de nuestro proyecto, denominada *resources* y copiamos dentro el contenido del archivo. El siguiente paso consiste en añadir la existencia de esos recursos al proyecto, agregándolos nuevamente en el fichero */WEB-INF/dispatcher-servlet.xml*:

```
<mvc:resources mapping="/resources/**" location="/resources/" />
```

Ilustración 89: Recursos

El último paso radica en incorporar dichos recursos dentro de los archivos jsp, con el objetivo de poder comenzar a utilizar las clases CSS ofrecidas por Bootstrap:

```
<head>
  <title>Venta de Localidades</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="<c:url value='/resources/css/bootstrap.min.css'/>" rel="stylesheet">
  <link href="<c:url value='/resources/css/bootstrap-theme.min.css'/>" rel="stylesheet">
  <script src="<c:url value='/resources/js/bootstrap.min.js'/>"></script>
</head>
```

Ilustración 90: Recursos JSP

El empleo de dichas clases CSS nos permitirán estandarizar el diseño. Antes de comenzar, es necesario realizar algunos preparativos importantes:

- **doctype de HTML5:** debido a que Bootstrap utiliza elementos HTML y propiedades CSS, hay que incluir en todas las páginas el siguiente código:

```

<!DOCTYPE html>
<html lang="es">
...
</html>

```

Ilustración 91: Estilo Doctype HTML5

- **responsive:** para que las páginas se adapten a los dispositivos móviles, hay que añadir la siguiente etiqueta dentro de la cabecera <head> de las páginas:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Ilustración 92: Estilo Responsive

Básicamente, lo que hace Bootstrap con la rejilla citada anteriormente es determinar un modelo de filas y en cada una de ellas definir de 1 hasta 12 columnas, con un ancho relativo al número de columnas indicado. Presenta la ventaja de que en cada pantalla podemos fijar las columnas que necesitamos.

Las filas se definen dentro de un contenedor de tipo *.container*. Para indicar el comienzo de cada una de ellas, se utiliza la clase “row”. Dentro de cada fila colocaremos tantos div como columnas tenga la fila. La sintaxis empleada para establecer cada columna es *col-md-**, donde el asterisco representa un valor entre 1 y 12, con el que se especifica cuántas de las 12 columnas de la fila ocupan [35]. A continuación se muestra un ejemplo de una de las pantallas de nuestra aplicación:

```

<div class="container">
  <div class="row">
    <div class="col-md-2">
      <ul class="nav nav-pills nav-stacked">
        <br>
        <br>
        <br>
        <li class="active"><a href="<c:url value='main'/>">Menú</a></li>
        <li><a href="<c:url value='/usuarios/listadoClientes'/>">Gestión de Usuarios</a></li>
        <li><a href="<c:url value='/deportes/listadoDeportes'/>">Gestión de Deportes</a></li>
        <li><a href="<c:url value='/competiciones/listadoCompeticiones'/>">Gestión de Competiciones</a></li>
        <li><a href="<c:url value='/participantes/listadoParticipantes'/>">Gestión de Participantes</a></li>
        <li><a href="<c:url value='/eventos/listadoEventos'/>">Gestión de Eventos</a></li>
        <li><a href="<c:url value='/entradas/listadoEntradas'/>">Gestión de Entradas</a></li>
        <li><a href="<c:url value='/compras/listadoCompras'/>">Gestión de Compras</a></li>
      </ul>
    </div>
    <div class="col-md-5">

```

Ilustración 93: Grid de Bootstrap

Prácticamente todas las pantallas de la aplicación están compuestas por una fila y dos columnas. La primera de ellas utiliza normalmente 2 unidades de columna, ya que se corresponde con el menú de funcionalidades (clase *nav nav-pills nav-stacked*) y la segunda puede variar según la pantalla donde nos encontremos.

Sin embargo, es común a todas ellas la aparición de una barra fija en la parte superior de la página. Dicho elemento se denomina barra de navegación. En los dispositivos móviles se muestran inicialmente minimizadas y al pulsar sobre ellas, se despliegan todas sus opciones [36]. Para definir la barra de navegación se ha empleado la clase *navbar navbar-inverse navbar-fixed-top*.

En el caso de la página principal de la aplicación, la barra de navegación contiene el formulario de identificación. Para ello, se incluye la etiqueta *form* junto con *navbar-form navbar-right* dentro de la clase *navbar-collapse collapse*, que permite extender o contraer la barra de navegación, en función del tamaño de la pantalla, además de sendos botones que posibilitan el acceso al sistema (clase *btn btn-success*) o eliminar los datos introducidos (clase *btn btn-info*).

Una vez dentro del sistema, la barra de navegación contiene el nombre del usuario logueado y un botón para salir del mismo (clase *btn btn-success*). Cuando el usuario pulsa sobre dicho botón, el sistema muestra una notificación indicando que se confirme la acción. Para ello, se utiliza una ventana secundaria, denominada modal, que utiliza una clase genérica denominada *modal fade* en cuyo interior aparecen las clases *modal-header*, *modal-body* y *modal-footer*, permitiendo formar dicha ventana [37]. Además, si el usuario que ha accedido al sistema contiene el rol cliente (ROL_USER), la barra de navegación incluye también un buscador, que permite localizar los eventos del sistema, sirviéndose de la clase *navbar-form navbar-left*.

Por otro parte, se han utilizado los iconos habilitados por Bootstrap en distintas páginas de la aplicación [28]. Para añadirlos, se ha usado el siguiente código:

```
<span class="glyphicon nombre del icono "></span>
```

Ilustración 94: Estilo iconos

Respecto a los botones principales, se han aplicado las siguientes clases:

- Crear una nueva entidad: *btn btn-success*.
- Editar una entidad: *btn btn-primary*.
- Eliminar una entidad: *btn btn-danger*. Al igual que ocurría con el botón de salir, este botón también dispone de una ventana secundaria cuando se acciona, denominada modal, que contiene las clases *modal-header*, *modal-body* y *modal-footer*.
- Visualizar una entidad: *btn btn-warning*.
- Poner en venta las entradas: *btn btn-warning*.
- Ver entradas: *btn btn-warning*.
- Comprar: *btn btn-danger*.
- Cancelar compra: *btn btn-danger*. El comportamiento de este botón es homólogo al de los botones habilitados para salir del sistema y eliminar una entidad, citados anteriormente.
- Imprimir entradas: *btn btn-inver*.

Además, las tablas que contienen el listado de las entidades presentan el siguiente esquema:

```
<table class="table table-striped">  
  ...  
</table>
```

Ilustración 95: Estilo tablas

Son tablas cuyas filas alternan su color de fondo para mejorar la legibilidad de los contenidos [38]. En inglés se denominan "*striped tables*". Todas ellas están incluidas dentro de la clase *table-responsive*, que permite ajustar la tabla al dispositivo desde donde se esté visualizando.

Por último, los formularios emplean la clase "error" para validar los datos introducidos.

```
.error {  
  color: red;  
}
```

Ilustración 96: Estilo error

Por otra parte, también se han manejado las siguientes tecnologías en el desarrollo del proyecto:

- **CSS**

La hoja de estilo en cascada o CSS es un lenguaje de marcas utilizado para describir la apariencia de un documento HTML. Los principales objetivos del desarrollo de CSS son la separación de la estructura de un documento de su presentación, el favorecimiento de la creación del contenido y su procesamiento y la adaptación del contenido al dispositivo donde se va a utilizar [10].

Las reglas de estilo pueden ubicarse en un documento separado (fichero específico referenciado) o en el mismo documento HTML (atributo *style* en etiquetas).

A continuación se muestran ambos ejemplos, utilizados en la aplicación:

```
<link rel="stylesheet" type="text/css" href="<c:url value='/resources/css/common.css' />">
```

Ilustración 97: CSS Ejemplo 1

```
<style type="text/css">

.carousel.vertical .active {
  top: 0;
}

.carousel.vertical .next {
  top: 100%;
}

.carousel.vertical .prev {
  top: -100%;
}

</style>
```

Ilustración 98: CSS Ejemplo 2

- **HTML**

HTML (HyperText Markup Language) es el lenguaje de marcado predominante para la elaboración de páginas web. Es un estándar que especifica un código,

denominado código HTML, que permite definir el contenido de una página web, como por ejemplo texto, imágenes o vídeos.

La estructura básica de un documento HTML es la siguiente:

```
<!DOCTYPE html>
<html>
  <head>

    <title>Venta de Localidades</title>

  </head>

  <body>

    <p>Ejemplo</p>

  </body>
</html>
```

Ilustración 99: HTML

- **Spring Security**

La seguridad en las aplicaciones constituye un tema muy importante y complejo. Caben destacar tres aspectos de la seguridad [12]:

- **Autenticación:** identificación de quién quiere realizar una acción en nuestra aplicación.
- **Autorización:** se utiliza la información validada durante la autenticación para determinar si a un usuario se le permite realizar una determinada acción en nuestra aplicación.
- **Encriptación:** se protege la información confidencial intercambiada con el servicio, de manera que no pueda utilizarse aunque sea interceptada.

Por lo tanto, para controlar los accesos a las vistas de la aplicación, se han añadido un conjunto de paquetes adicionales a los básicos de Spring Framework, constituyendo un módulo orientado a seguridad de aplicaciones web en Java denominado Spring Security. Dicho módulo se encuentra en el directorio “Localidades/dependencias/security” del proyecto.

Para llevar a cabo el proceso, es necesario incluir el siguiente filtro en el fichero */WEB-INF/web.xml* e indicar las URLs a proteger:

```

<filter>
  <filter-name>springSecurityFilterChain</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<filter-mapping>
  <filter-name>springSecurityFilterChain</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

```

Ilustración 100: Filtro Spring Security

El último paso consiste en comprobar las credenciales de los usuarios que intentan acceder a la aplicación y permitirles el acceso a las vistas definidas, según el rol del usuario. Para ello, se obtiene la información de la autenticación utilizando la tabla “usuarios” de la base de datos, realizando la siguiente consulta:

```

<authentication-manager>
  <authentication-provider>
    <jdbc-user-service data-source-ref="dataSource"
      users-by-username-query="select usuario,clave, true from usuarios where usuario=?"
      authorities-by-username-query="select usuario, rol from usuarios where usuario=?"
    />
  </authentication-provider>
</authentication-manager>

```

Ilustración 101: Autenticación

Posteriormente, especificamos patrones que contienen la dirección de acceso a los controladores habilitados, los cuales mostrarán las vistas a los roles asignados. A continuación se muestran algunos ejemplos:

```

<security:intercept-url pattern="/usuarios/listadoClientes" access="hasRole('ROLE_ADMIN')" />
<security:intercept-url pattern="/usuarios/visualizaCliente/*" access="hasAnyRole('ROLE_USER')" />
<security:intercept-url pattern="/usuarios/borraCliente/*" access="hasAnyRole('ROLE_ADMIN', 'ROLE_USER')" />

```

Ilustración 102: Seguridad URLs

Todo este proceso se lleva a cabo en el fichero */WEB-INF/security.xml* y posteriormente se incluye en el archivo */WEB-INF/applicationContext.xml*:

```
<import resource="security.xml" />
```

Ilustración 103: Import security.xml

- **JSP/JSTL**

JavaServer Pages (JSP) es una tecnología que permite crear páginas web dinámicas basadas en HTML o XML, entre otros tipos de documentos. Se requiere de un servidor web como Apache Tomcat para desplegar y ejecutar JSP [29].

Por otro lado, existe un componente, denominado JavaServer Pages Standard Tag Library (JSTL), que extiende los JSP citados anteriormente, proporcionando librerías de etiquetas, llamadas Tag Libraries, que ofrecen funciones muy útiles, tales como iteraciones o condicionales [32]. Para poder utilizar este componente, se necesita declarar la siguiente directiva en la parte superior del fichero jsp:

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
```

Ilustración 104: Directiva JSTL

A continuación se muestra un ejemplo sencillo del uso de JSTL dentro de un fichero JSP:

```
<c:if test="${empty eventos}">
  <center>NO HAY EVENTOS DISPONIBLES</center>
</c:if>

<c:if test="${not empty eventos}">
  <div class="table-responsive">
    <table class="table table-striped">

      <tr class="text-capitalize">
        <th>Id</th>
        <th>Nombre</th>
      </tr>

      <c:forEach var="c" items="${eventos}">
        <tr class="table-striped">
          <td>${c.idEvento}</td>
          <td>${c.nombre}</td>
        </tr>
      </c:forEach>
    </table>
  </div>
</c:if>
```

Ilustración 105: Ejemplo JSTL

- jQuery

Es una biblioteca que ofrece una serie de funcionalidades basadas en JavaScript (JS). Proporciona mejoras en la interfaz de usuario y páginas web dinámicas, mediante la incorporación de efectos, animaciones, ventanas con notificaciones, etc. [34].

jQuery consiste en un único fichero JavaScript que contiene las funcionalidades. Su utilización se ha llevado a cabo mediante el empleo de una red de distribución de contenido (CDN, *content delivery network*), ofreciendo de esta manera el alojamiento de archivos JavaScript y la posterior descarga de dichos recursos. Este sistema proporciona una velocidad de acceso más rápida.

Las ventanas modales de la aplicación han sido creadas usando esta tecnología.

- **Apache Derby**

Apache Derby es un sistema gestor de bases de datos relacional implementado en Java y disponible bajo la licencia Apache 2.0 [39]. Además de ser de código abierto, es compatible con SQL estándar.

Al igual que JPA, la integración de la base de datos en nuestro sistema se realiza en el fichero *applicationContext.xml* situado en el directorio */web/WEB-INF* del proyecto:

```
<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName" value="org.apache.derby.jdbc.ClientDriver" />
    <property name="url" value="jdbc:derby://localhost:1527/tfg" />
    <property name="username" value="tfg" />
    <property name="password" value="tfg" />
</bean>
```

Ilustración 106: Integración BB.DD

- **Bean Validation**

Es una especificación de Java que proporciona declaración de restricciones y metadatos de validación. Utiliza anotaciones para indicar restricciones en propiedades predefinidas o personalizadas [10]. A continuación se muestran ejemplos de restricciones:

```
public class Usuario implements Serializable{
    @NotEmpty @Size(min = 5, max = 30)
    private String nombre;
}
```

Ilustración 107: Bean Validation

La personalización de los mensajes se lleva a cabo en un fichero denominado *message.properties*, ubicado en el paquete “es.ujaen.tfg.msg”. Un ejemplo sería el siguiente:

```
NotEmpty.cliente.nombre=Campo requerido  
Size.cliente.nombre=Inserta un nombre válido entre 5 y 30 caracteres
```

Ilustración 108: Ejemplo mensaje

El siguiente paso consiste en configurar el fichero *dispatcher-servlet.xml*:

```
<bean class="org.springframework.context.support.ResourceBundleMessageSource"  
      id="messageSource">  
    <property name="basename" value="es.ujaen.tfg.msg.messages" />  
</bean>
```

Ilustración 109: Bean Validation Configuración

Para la visualización de errores en la vista se añade la etiqueta inferior:

```
<form:error path="propiedad" cssClass="estilo">
```

Ilustración 110: Etiqueta error

Las librerías relacionadas con esta especificación se ubican en el directorio “Localidades/dependencias/validation” del proyecto.

- **Apache Tomcat**

Apache Tomcat es un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Oracle Corporation [40].

Tomcat es un servidor libre, de código abierto y está escrito en Java, lo que permite que pueda funcionar en cualquier sistema operativo que disponga de una máquina virtual de Java [40].

Al utilizar Spring Framework, no es necesario identificar el recurso, ya que el framework se encarga del proceso de formación.

La versión utilizada en este proyecto es Apache Tomcat 8.0.9.

4.2.4. Interacción con PayPal

Para abonar las entradas adquiridas por el cliente en cada una de las compras se ha utilizado la opción de pago estándar de PayPal. El flujo del proceso se describe a continuación [9]:



Ilustración 111: Pago estándar

- 1) Una vez que el cliente ha seleccionado las entradas deseadas para un evento en concreto, pulsa sobre el botón habilitado en la aplicación web para el pago a través de PayPal.
- 2) El cliente es dirigido a PayPal.
- 3) Paypal muestra el resumen de la compra realizada y proporciona un formulario al cliente para que inicie sesión en su cuenta. Una vez identificado, puede revisar la información de envío y facturación. El siguiente paso consiste en hacer en click en **Pagar** para realizar el pago, el cual se lleva a cabo de manera instantánea.
- 4) El cliente revisa la confirmación de pago de PayPal.
- 5) El cliente es redirigido a la aplicación web.
- 6) El cliente accede a la pantalla de confirmación de la compra.

Respecto a la implementación, para integrar en la aplicación web el sistema de pago anteriormente descrito, en primer lugar hay que identificar un punto concreto donde hay que colocar un botón en el que el comprador deberá pulsar para comenzar el pago. El botón, etiquetado como **Pagar**, ejecuta un Form POST a PayPal. Al hacer click, se redirige al cliente a la página de pago de PayPal, desde donde podrá pagar con su cuenta de PayPal. [9]. A continuación se muestra en color verde el botón habilitado en la aplicación para el pago:

The screenshot shows a web application interface for purchasing sports tickets. At the top, there is a dark header bar with the user's name 'Ramón Bailén Sánchez' and a green 'Salir' button. On the left, a blue sidebar menu contains 'Menú', 'Datos Personales', 'Ver Eventos', 'Ver Compras Realizadas', and 'Entradas Deportes'. The main content area is titled 'Entradas Deportes' and 'Confirmación de Compra'. It displays the following details: 'Compra nº 16', 'Cliente Ramón Bailén Sánchez', 'Fecha de Compra 2015-05-23 18:44:53.73', and 'Total 40.0 €'. Below this, a green box titled 'Tipo de Entrada' contains the details: 'Descripción Entrada 13 para evento: Real Madrid Vs FC Barcelona', 'Deporte Fútbol', and 'Precio 40.0 €'. At the bottom, there is a red 'X Cancelar Compra' button and a green 'PayPal PAGAR' button.

Ilustración 112: Botón de pagar

El Form POST contiene un conjunto de variables HTML que describen la transacción. El carro de la compra estándar incluye algunas de manera obligatoria, como por ejemplo [9]:

- **subtotal:** importe de la transacción
- **business:** identificador del vendedor o dirección de correo electrónico a su cuenta PayPal
- **paymentaction:** indica si la transacción es para un pago de una venta final o una autorización para una venta final

Seguidamente se incluye una ilustración acerca del código fuente utilizado para la integración de la funcionalidad de pago y se detalla el funcionamiento de éste:

```

<!-- Formulario de Pago con Paypal -->
<form action="https://www.sandbox.paypal.com/cgi-bin/webscr" method="post">

  <% int var=1;%>

  <c:forEach var="entrada" items="{entradasCompra}" varStatus="i">
    <input type="hidden" name="cmd" value="_cart">
    <input type="hidden" name="upload" value="1">
    <input type="hidden" name="lc" value="ES">
    <input type="hidden" name="currency_code" value="EUR">
    <input type="hidden" name="business" value="rb389-facilitator@hotmail.com">

    <input type="hidden" name="item_name" value="{out.println(var);%}" value="{entrada.descripcion}">
    <input type="hidden" name="amount_<out.println(var);%>" value="{entrada.precio}">

    <input type="hidden" name="paymentaction" value="sale">

    <% var = var + 1;%>
  </c:forEach>

  <input width="90" height="32" type="image" src="https://www.paypal.com/es/ES/i/btn/x-click-but02.gif" border="0" name="submit">

</form>

```

Ilustración 113: Código fuente PayPal

El procesamiento de pago de PayPal ha sido integrado mediante la utilización de un carro de la compra de terceros, en lugar del carro de la compra estándar definido anteriormente, ya que permite transferir a PayPal los detalles de los artículos seleccionados, en vez de el importe total del carro completo. Para ello, se han definido las variables **cmd** con value="_cart" y **upload** con value="1".

La variable **lc** hace referencia al idioma de visualización de la página de inicio de sesión o creación de cuenta, cuyo valor es ES (español), mientras que **currency_code** determina la divisa del pago y su valor es EUR (euro).

Por otro lado, las variables **business** y **paymentaction**, definidas anteriormente, contienen la dirección de correo electrónico del vendedor (rb389-facilitator@hotmail.com) y el valor correspondiente para el pago de una venta final (sale), respectivamente.

Para especificar los detalles de los artículos, se han utilizado las variables **item_name** y **amount**. La primera define el nombre del artículo y la segunda el precio. Para cada uno de los parámetros anteriores, se fija un nuevo grupo de valores que corresponde a cada artículo comprado. Como puede observarse en la imagen, dentro del bucle se ha añadido "_var" a los nombres de las dos variables anteriores, siendo **var** el número del artículo, empezando por 1 e incrementando en uno cada artículo que se añade antes de la finalización del bucle.

Por último, el botón que redirige al cliente a la página de PayPal contiene una imagen añadida justo antes de cerrar la etiqueta del formulario.

4.2.4.1. Entorno de pruebas de pago

Una vez que un usuario cliente ha comprado un número determinado de entradas, el siguiente paso consiste en abonar el precio de estas, por lo que deberá pulsar sobre el botón de pago habilitado en la aplicación. Así, el sistema redirigirá al usuario a PayPal, donde se mostrará el resumen de la compra y el formulario de identificación. Una vez que el cliente ha introducido sus credenciales, el último paso radica en hacer click en el botón “Pagar” situado en la página de confirmación de pagos.

PayPal aplica un retorno automático a todas las funcionalidades de pago estándar, de manera que una vez se ha realizado el desembolso, el usuario es redirigido de nuevo a su sitio rápidamente.

Para configurar el retroceso automático, debe proporcionarse en el formulario que contiene el carro de la compra de terceros la URL de devolución, donde se hará saber al comprador que el pago se ha realizado satisfactoriamente y que la transacción ha finalizado. Debido a que localhost no es una URL válida, existen una serie de servicios, denominados hostings, que están conectados a Internet, permitiendo a cualquier internauta el acceso a la aplicación. Sin embargo, la mayoría de estos hostings proporcionan soporte para PHP y MySQL. Además, cada vez que se realiza una prueba en el servidor remoto, habría que compilar la aplicación y hacer un despliegue, con la lentitud que ello conlleva. Teniendo en cuenta que el proyecto ha sido desarrollado con Java y que el coste de ofrecer un alojamiento con soporte para esta tecnología suele ser caro, se ha tomado la decisión de montar un entorno de desarrollo donde se simula la redirección de PayPal a nuestra aplicación.

A continuación se describe el procedimiento realizado:

- 1) Se ha habilitado un formulario donde el usuario cliente introduce el identificador de la compra que acaba de realizar. Realmente, PayPal devolvería un identificador único de la transacción realizada, redirigiendo directamente al usuario a la página de confirmación de la compra, con su respectiva información. Seguidamente se ilustra dicho formulario:

Formulario de Prueba

Identificador de transacción

Ilustración 114: Entorno de desarrollo PayPal

- 2) Una vez que los datos han sido enviados a través del formulario anterior, se crea automáticamente la fecha asociada al pago y se redirige al usuario a la vista que muestra la confirmación de la compra realizada. En ella se muestran los datos relacionados con la misma, las entradas correspondientes y un botón que permite al usuario imprimir las entradas si lo desea, además de ofrecer una serie de recomendaciones deportivas, tal y como se muestra en el ejemplo de la siguiente ilustración:

Ramón Bailén Sánchez » Salir

Confirmación de Pago

Su compra ha sido pagada

Dispones de 4 días desde la fecha de pago para imprimir las entradas

Id Compra: 35
Usuario: Ramón Bailén Sánchez
Fecha de Compra: 2015-06-05 13:37:23.535
Fecha de Pago: 2015-06-05 13:37:39.241
Precio Total: 2.0 €

ENTRADAS

Descripción
Entrada 2 para evento: Prueba

Deporte
Fútbol

Precio
2.0 €



Sugerencias



Compre entradas para alguno de nuestros deportes

Ilustración 115: Confirmación de pago

4.2.5. Generación de código QR

El usuario con perfil cliente dispone de unos días para imprimir las entradas adquiridas. En ellas aparece la siguiente información:

- Descripción
- Deporte al que pertenecen
- Precio
- **Código QR**

Un código QR <<Quick Response>> (respuesta rápida) es un sistema que permite almacenar información en una matriz de puntos o en un código de barras bidimensional. Presenta tres cuadrados en las esquinas que permiten detectar la posición del código al lector [41]. A través de este sistema, puede controlarse la entrada de personas a un recinto deportivo. Es mucho más rápido que comprobar la identidad de las mismas mediante la entrega de un carnet.

Se ha integrado de manera trivial en nuestra aplicación por medio del uso de un servicio externo generado con una API en PHP situado en la página <http://www.qrplanet.com>, debido a que se ha desarrollado un prototipo, aunque existen bibliotecas que lo hacen.

Cada entrada dispone de un código QR con la siguiente estructura:

```

```

Ilustración 116: Implementación QR

El cuadro rojo anterior contiene el identificador de la compra, el identificador de usuario que ha adquirido las entradas, el deporte al que pertenece el evento correspondiente y el precio de la entrada. A continuación se muestra un ejemplo:



Ilustración 117: Código QR

4.3. Pruebas

El objetivo de esta fase consiste en la realización de un conjunto de pruebas que permitan verificar y validar el sistema desarrollado, es decir, comprobar que el software realiza correctamente las tareas especificadas en los requerimientos funcionales ([capítulo 2.5.1](#)).

En este contexto, puede definirse el término prueba como el proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene un alta probabilidad de mostrar un error no descubierto hasta entonces [2].

El funcionamiento de un sistema software puede probarse de una de estas dos formas [2]:

- 1) Conociendo los requisitos funcionales, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa y al mismo tiempo buscando errores potenciales.
- 2) Conociendo el funcionamiento del producto, se pueden desarrollar pruebas que comprueben que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han probado de forma adecuada. En estas pruebas se comprueban los caminos lógicos del programa, además de bucles y condiciones, examinando el estado del programa en varios puntos.

El primer enfoque se denomina prueba de caja negra y el segundo, prueba de caja blanca. Las pruebas de validación que se van a llevar a cabo sobre nuestro sistema son las de caja negra, ya que su realización sobre la interfaz del software resulta ideal.

A continuación se muestran los casos de prueba desarrollados y los resultados obtenidos para cada uno de ellos.

4.3.1. Casos de prueba**Prueba 1:** Identificación de un usuario

Requisito testado	RF-01
Acción	El usuario introduce sus credenciales en el formulario de acceso a la aplicación
Checkpoint 1	El sistema valida los datos y muestra al usuario el menú de inicio correspondiente

Tabla XXXI: Prueba 1

Prueba 2: Identificación incorrecta de un usuario

Requisito testado	RF-01
Acción	El usuario introduce unas credenciales no válidas en el formulario de acceso a la aplicación
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que los datos suministrados no son válidos

Tabla XXXII: Prueba 2

Prueba 3: Salir de la aplicación

Requisito testado	RF-02
Acción	Una vez logueado, el usuario acciona en cualquier momento el botón de salir, situado en la parte superior derecha
Checkpoint 1	El sistema muestra una notificación de confirmación de la operación
Acción	El usuario pulsa sobre el botón de salir
Checkpoint 2	El sistema conduce al usuario a la pantalla de inicio de la aplicación

Tabla XXXIII: Prueba 3

Prueba 4: Registro de usuario

Requisito testado	RF-03
Acción	El usuario rellena el formulario de registro situado en la pantalla de inicio de la aplicación
Checkpoint 1	El sistema recibe los datos del usuario y muestra un mensaje de confirmación

Tabla XXXIV: Prueba 4

Prueba 5: Registro de usuario con el campo del nombre vacío

Requisito testado	RF-03
Acción	Campo del nombre vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre no puede estar vacío

Tabla XXXV: Prueba 5

Prueba 6: Registro de usuario con el campo del nombre incorrecto

Requisito testado	RF-03
Acción	Longitud del campo del nombre incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre dispone de una longitud mínima y máxima

Tabla XXXVI: Prueba 6

Prueba 7: Registro de usuario con el campo del correo vacío

Requisito testado	RF-03
Acción	Campo del correo vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el correo no puede estar vacío

Tabla XXXVII: Prueba 7

Prueba 8: Registro de usuario con el campo del correo incorrecto

Requisito testado	RF-03
Acción	Longitud del campo del correo incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la dirección de correo no es válida

Tabla XXXVIII: Prueba 8

Prueba 9: Registro de usuario con el campo de la ciudad vacío

Requisito testado	RF-03
Acción	Campo de la ciudad vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la ciudad no puede estar vacía

Tabla XXXIX: Prueba 9

Prueba 10: Registro de usuario con el campo de la ciudad incorrecto

Requisito testado	RF-03
Acción	Longitud del campo de la ciudad incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la ciudad dispone de una longitud mínima y máxima

Tabla XL: Prueba 10

Prueba 11: Registro de usuario con el campo del nombre de usuario vacío

Requisito testado	RF-03
Acción	Campo del nombre de usuario vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre de usuario no puede estar vacío

Tabla XLI: Prueba 11

Prueba 12: Registro de usuario con el campo del nombre de usuario incorrecto

Requisito testado	RF-03
Acción	Longitud del campo del nombre de usuario incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre de usuario dispone de una longitud mínima y máxima

Tabla XLII: Prueba 12

Prueba 13: Registro de usuario con el campo del nombre de usuario existente en la base de datos

Requisito testado	RF-03
Acción	Campo del nombre de usuario existente
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que ya existe un usuario con ese nombre de usuario

Tabla XLIII: Prueba 13

Prueba 14: Registro de usuario con el campo de la clave vacío

Requisito testado	RF-03
Acción	Campo de la clave vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la clave no puede estar vacía

Tabla XLIV: Prueba 14

Prueba 15: Registro de usuario con el campo de la clave incorrecto

Requisito testado	RF-03
Acción	Longitud del campo de la clave incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la clave dispone de una longitud mínima y máxima

Tabla XLV: Prueba 15

Prueba 16: Registro de usuario con el campo de la edad vacío

Requisito testado	RF-03
Acción	Campo de la edad vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la edad no puede estar vacía

Tabla XLVI: Prueba 16

Prueba 17: Alta de usuario

Requisito testado	RF-04
Acción	El administrador rellena el formulario de alta de un nuevo usuario
Checkpoint 1	El sistema recibe los datos, crea el nuevo usuario y conduce al administrador al listado

Tabla XLVII: Prueba 17

Prueba 18: Alta de usuario con el campo del nombre vacío

Requisito testado	RF-04
Acción	Campo del nombre vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre no puede estar vacío

Tabla XLVIII: Prueba 18

Prueba 19: Alta de usuario con el campo del nombre incorrecto

Requisito testado	RF-04
Acción	Longitud del campo del nombre incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre dispone de una longitud mínima y máxima

Tabla XLIX: Prueba 19

Prueba 20: Alta de usuario con el campo del correo vacío

Requisito testado	RF-04
Acción	Campo del correo vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el correo no puede estar vacío

Tabla L: Prueba 20

Prueba 21: Alta de usuario con el campo del correo incorrecto

Requisito testado	RF-04
Acción	Longitud del campo del correo incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la dirección de correo no es válida

Tabla LI: Prueba 21

Prueba 22: Alta de usuario con el campo de la ciudad vacío

Requisito testado	RF-04
Acción	Campo de la ciudad vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la ciudad no puede estar vacía

Tabla LII: Prueba 22

Prueba 23: Alta de usuario con el campo de la ciudad incorrecto

Requisito testado	RF-04
Acción	Longitud del campo de la ciudad incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la ciudad dispone de una longitud mínima y máxima

Tabla LIII: Prueba 23

Prueba 24: Alta de usuario con el campo del nombre de usuario vacío

Requisito testado	RF-04
Acción	Campo del nombre de usuario vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre de usuario no puede estar vacío

Tabla LIV: Prueba 24

Prueba 25: Alta de usuario con el campo del nombre de usuario incorrecto

Requisito testado	RF-04
Acción	Longitud del campo del nombre de usuario incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre de usuario dispone de una longitud mínima y máxima

Tabla LV: Prueba 25

Prueba 26: Alta de usuario con el campo del nombre de usuario existente en la base de datos

Requisito testado	RF-04
Acción	Campo del nombre de usuario existente
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que ya existe un usuario con ese nombre de usuario

Tabla LVI: Prueba 26

Prueba 27: Alta de usuario con el campo de la clave vacío

Requisito testado	RF-04
Acción	Campo de la clave vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la clave no puede estar vacía

Tabla LVII: Prueba 27

Prueba 28: Alta de usuario con el campo de la clave incorrecto

Requisito testado	RF-04
Acción	Longitud del campo de la clave incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la clave dispone de una longitud mínima y máxima

Tabla LVIII: Prueba 28

Prueba 29: Editar usuario

Requisito testado	RF-05
Acción	El administrador selecciona la opción de editar un usuario del sistema
Checkpoint 1	El sistema recibe los nuevos datos del usuario y conduce al administrador al listado

Tabla LIX: Prueba 29

Prueba 30: Editar usuario con el campo del nombre vacío

Requisito testado	RF-05
Acción	Campo del nombre vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre no puede estar vacío

Tabla LX: Prueba 30

Prueba 31: Editar usuario con el campo del nombre incorrecto

Requisito testado	RF-05
Acción	Longitud del campo del nombre incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre dispone de una longitud mínima y máxima

Tabla LXI: Prueba 31

Prueba 32: Editar usuario con el campo del correo vacío

Requisito testado	RF-05
Acción	Campo del correo vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el correo no puede estar vacío

Tabla LXII: Prueba 32

Prueba 33: Editar usuario con el campo del correo incorrecto

Requisito testado	RF-05
Acción	Longitud del campo del correo incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la dirección de correo no es válida

Tabla LXIII: Prueba 33

Prueba 34: Editar usuario con el campo de la ciudad vacío

Requisito testado	RF-05
Acción	Campo de la ciudad vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la ciudad no puede estar vacía

Tabla LXIV: Prueba 34

Prueba 35: Editar usuario con el campo de la ciudad incorrecto

Requisito testado	RF-05
Acción	Longitud del campo de la ciudad incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la ciudad dispone de una longitud mínima y máxima

Tabla LXV: Prueba 35

Prueba 36: Editar usuario con el campo del nombre de usuario vacío

Requisito testado	RF-05
Acción	Campo del nombre de usuario vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre de usuario no puede estar vacío

Tabla LXVI: Prueba 36

Prueba 37: Editar usuario con el campo del nombre de usuario incorrecto

Requisito testado	RF-05
Acción	Longitud del campo del nombre de usuario incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre de usuario dispone de una longitud mínima y máxima

Tabla LXVII: Prueba 37

Prueba 38: Editar usuario con el campo del nombre de usuario existente en la base de datos

Requisito testado	RF-05
Acción	Campo del nombre de usuario existente
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que ya existe un usuario con ese nombre de usuario

Tabla LXVIII: Prueba 38

Prueba 39: Editar usuario con el campo de la clave vacío

Requisito testado	RF-05
Acción	Campo de la clave vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la clave no puede estar vacía

Tabla LXIX: Prueba 39

Prueba 40: Editar usuario con el campo de la clave incorrecto

Requisito testado	RF-05
Acción	Longitud del campo de la clave incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la clave dispone de una longitud mínima y máxima

Tabla LXX: Prueba 40

Prueba 41: Eliminar usuario

Requisito testado	RF-06
Acción	El administrador accede a la gestión de usuarios y selecciona la opción de eliminar un usuario
Checkpoint 1	El sistema muestra una notificación de confirmación de la operación
Acción	El administrador pulsa sobre el botón de eliminar
Checkpoint 2	El sistema elimina la información de dicho usuario

Tabla LXXI: Prueba 41

Prueba 42: Listado de usuarios

Requisito testado	RF-07
Acción	El administrador accede a la pantalla del listado de usuarios
Checkpoint 1	El sistema muestra un listado con todos los usuarios

Tabla LXXII: Prueba 42

Prueba 43: Alta de deporte

Requisito testado	RF-08
Acción	El administrador rellena el formulario de alta de un nuevo deporte
Checkpoint 1	El sistema recibe los datos, crea el nuevo deporte y conduce al administrador al listado

Tabla LXXIII: Prueba 43

Prueba 44: Alta de deporte con el campo del nombre vacío

Requisito testado	RF-08
Acción	Campo nombre vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre no puede estar vacío

Tabla LXXIV: Prueba 44

Prueba 45: Alta de deporte con el campo del nombre incorrecto

Requisito testado	RF-08
Acción	Longitud del campo nombre incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre dispone de una longitud mínima y máxima

Tabla LXXV: Prueba 45

Prueba 46: Alta de deporte con el campo de la denominación del participante vacío

Requisito testado	RF-08
Acción	Campo denominación del participante vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la denominación del participante no puede estar vacía

Tabla LXXVI: Prueba 46

Prueba 47: Alta de deporte con el campo de la denominación del participante incorrecto

Requisito testado	RF-08
Acción	Longitud del campo denominación participante incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la denominación del participante dispone de una longitud mínima y máxima

Tabla LXXVII: Prueba 47

Prueba 48: Alta de deporte con el campo de la denominación del evento vacío

Requisito testado	RF-08
Acción	Campo denominación del evento vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la denominación del evento no puede estar vacía

Tabla LXXVIII: Prueba 48

Prueba 49: Alta de deporte con el campo de la denominación del evento incorrecto

Requisito testado	RF-08
Acción	Longitud del campo denominación evento incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la denominación del evento dispone de una longitud mínima y máxima

Tabla LXXIX: Prueba 49

Prueba 50: Editar deporte

Requisito testado	RF-09
Acción	El administrador selecciona la opción de editar un deporte del sistema
Checkpoint 1	El sistema recibe los nuevos datos del deporte y conduce al administrador al listado

Tabla LXXX: Prueba 50

Prueba 51: Editar deporte con el campo del nombre vacío

Requisito testado	RF-09
Acción	Campo nombre vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre no puede estar vacío

Tabla LXXXI: Prueba 51

Prueba 52: Editar deporte con el campo del nombre incorrecto

Requisito testado	RF-09
Acción	Longitud del campo nombre incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre dispone de una longitud mínima y máxima

Tabla LXXXII: Prueba 52

Prueba 53: Editar deporte con el campo de la denominación del participante vacío

Requisito testado	RF-09
Acción	Campo denominación del participante vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la denominación del participante no puede estar vacía

Tabla LXXXIII: Prueba 53

Prueba 54: Editar deporte con el campo de la denominación del participante incorrecto

Requisito testado	RF-09
Acción	Longitud del campo denominación participante incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la denominación del participante dispone de una longitud mínima y máxima

Tabla LXXXIV: Prueba 54

Prueba 55: Editar deporte con el campo de la denominación del evento vacío

Requisito testado	RF-09
Acción	Campo denominación del evento vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la denominación del evento no puede estar vacía

Tabla LXXXV: Prueba 55

Prueba 56: Editar deporte con el campo de la denominación del evento incorrecto

Requisito testado	RF-09
Acción	Longitud del campo denominación evento incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la denominación del evento dispone de una longitud mínima y máxima

Tabla LXXXVI: Prueba 56

Prueba 57: Eliminar deporte

Requisito testado	RF-10
Acción	El administrador accede a la gestión de deportes y selecciona la opción de eliminar un deporte
Checkpoint 1	El sistema muestra una notificación de confirmación de la operación
Acción	El administrador pulsa sobre el botón de eliminar
Checkpoint 2	El sistema elimina la información de dicho deporte

Tabla LXXXVII: Prueba 57

Prueba 58: Listado de deportes

Requisito testado	RF-11
Acción	El administrador accede a la pantalla del listado de deportes
Checkpoint 1	El sistema muestra un listado con todos los deportes

Tabla LXXXVIII: Prueba 58

Prueba 59: Alta de competición

Requisito testado	RF-12
Acción	El administrador rellena el formulario de alta de una nueva competición
Checkpoint 1	El sistema recibe los datos, crea la nueva competición y conduce al administrador al listado

Tabla LXXXIX: Prueba 59

Prueba 60: Alta de competición con el campo del nombre vacío

Requisito testado	RF-12
Acción	Campo nombre vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre no puede estar vacío

Tabla XC: Prueba 60

Prueba 61: Alta de competición con el campo del nombre incorrecto

Requisito testado	RF-12
Acción	Longitud del campo nombre incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre dispone de una longitud mínima y máxima

Tabla XCI: Prueba 61

Prueba 62: Alta de competición sin seleccionar un deporte

Requisito testado	RF-12
Acción	Campo deporte vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el deporte no puede estar vacío

Tabla XCII: Prueba 62

Prueba 63: Editar competición

Requisito testado	RF-13
Acción	El administrador selecciona la opción de editar una competición del sistema
Checkpoint 1	El sistema recibe los nuevos datos de la competición y conduce al administrador al listado

Tabla XCIII: Prueba 63

Prueba 64: Editar competición con el campo del nombre vacío

Requisito testado	RF-13
Acción	Campo nombre vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre no puede estar vacío

Tabla XCIV: Prueba 64

Prueba 65: Editar competición con el campo del nombre incorrecto

Requisito testado	RF-13
Acción	Longitud del campo nombre incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre dispone de una longitud mínima y máxima

Tabla XCV: Prueba 65

Prueba 66: Eliminar competición

Requisito testado	RF-14
Acción	El administrador accede a la gestión de competiciones y selecciona la opción de eliminar una competición
Checkpoint 1	El sistema muestra una notificación de confirmación de la operación
Acción	El administrador pulsa sobre el botón de eliminar
Checkpoint 2	El sistema elimina la información de dicha competición

Tabla XCVI: Prueba 66

Prueba 67: Listado de competiciones

Requisito testado	RF-15
Acción	El administrador accede a la pantalla del listado de competiciones
Checkpoint 1	El sistema muestra un listado con todas las competiciones

Tabla XCVII: Prueba 67

Prueba 68: Alta de participante

Requisito testado	RF-16
Acción	El administrador rellena el formulario de alta de un nuevo participante
Checkpoint 1	El sistema recibe los datos, crea el nuevo participante y conduce al administrador al listado

Tabla XCVIII: Prueba 68

Prueba 69: Alta de participante con el campo del nombre vacío

Requisito testado	RF-16
Acción	Campo nombre vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre no puede estar vacío

Tabla XCIX: Prueba 69

Prueba 70: Alta de participante con el campo del nombre incorrecto

Requisito testado	RF-16
Acción	Longitud del campo nombre incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre dispone de una longitud mínima y máxima

Tabla C: Prueba 70

Prueba 71: Alta de participante sin seleccionar un deporte

Requisito testado	RF-16
Acción	Campo deporte vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el deporte no puede estar vacío

Tabla CI: Prueba 71

Prueba 72: Editar participante

Requisito testado	RF-17
Acción	El administrador selecciona la opción de editar un participante del sistema
Checkpoint 1	El sistema recibe los nuevos datos del participante y conduce al administrador al listado

Tabla CII: Prueba 72

Prueba 73: Editar participante con el campo del nombre vacío

Requisito testado	RF-17
Acción	Campo nombre vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre no puede estar vacío

Tabla CIII: Prueba73

Prueba 74: Editar participante con el campo del nombre incorrecto

Requisito testado	RF-17
Acción	Longitud del campo nombre incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre dispone de una longitud mínima y máxima

Tabla CIV: Prueba 74

Prueba 75: Eliminar participante

Requisito testado	RF-18
Acción	El administrador accede a la gestión de participantes y selecciona la opción de eliminar un participante
Checkpoint 1	El sistema muestra una notificación de confirmación de la operación
Acción	El administrador pulsa sobre el botón de eliminar
Checkpoint 2	El sistema elimina la información de dicho participante

Tabla CV: Prueba 75

Prueba 76: Listado de participantes

Requisito testado	RF-19
Acción	El administrador accede a la pantalla del listado de participantes
Checkpoint 1	El sistema muestra un listado con todos los participantes

Tabla CVI: Prueba 76

Prueba 77: Alta de evento

Requisito testado	RF-20
Acción	El administrador rellena el formulario de alta de un nuevo evento
Checkpoint 1	El sistema recibe los datos, crea el nuevo evento y conduce al administrador al listado

Tabla CVII: Prueba 77

Prueba 78: Alta de evento con el campo del nombre vacío

Requisito testado	RF-20
Acción	Campo del nombre vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre no puede estar vacío

Tabla CVIII: Prueba 78

Prueba 79: Alta de evento con el campo del nombre incorrecto

Requisito testado	RF-20
Acción	Longitud del campo del nombre incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre dispone de una longitud mínima y máxima

Tabla CIX: Prueba 79

Prueba 80: Alta de evento con el campo de la fecha vacío

Requisito testado	RF-20
Acción	Campo de la fecha vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la fecha no puede estar vacía

Tabla CX: Prueba 80

Prueba 81: Alta de evento con el campo de la hora vacío

Requisito testado	RF-20
Acción	Campo de la hora vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la hora no puede estar vacía

Tabla CXI: Prueba 81

Prueba 82: Alta de evento con el campo del lugar vacío

Requisito testado	RF-20
Acción	Campo del lugar vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el lugar no puede estar vacío

Tabla CXII: Prueba 82

Prueba 83: Alta de evento con el campo del lugar incorrecto

Requisito testado	RF-20
Acción	Longitud del campo del lugar incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el lugar dispone de una longitud mínima y máxima

Tabla CXIII: Prueba 83

Prueba 84: Alta de evento con el campo de la ciudad vacío

Requisito testado	RF-20
Acción	Campo de la ciudad vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la ciudad no puede estar vacía

Tabla CXIV: Prueba 84

Prueba 85: Alta de evento con el campo de la ciudad incorrecto

Requisito testado	RF-20
Acción	Longitud del campo de la ciudad incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la ciudad dispone de una longitud mínima y máxima

Tabla CXV: Prueba 85

Prueba 86: Alta de evento con el campo del país vacío

Requisito testado	RF-20
Acción	Campo del país vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el país no puede estar vacío

Tabla CXVI: Prueba 86

Prueba 87: Alta de evento con el campo del deporte vacío

Requisito testado	RF-20
Acción	Campo del deporte vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el deporte no puede estar vacío

Tabla CXVII: Prueba 87

Prueba 88: Alta de evento con el campo de la competición vacío

Requisito testado	RF-20
Acción	Campo de la competición vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la competición no puede estar vacía

Tabla CXVIII: Prueba 88

Prueba 89: Alta de evento con el campo del participante vacío

Requisito testado	RF-20
Acción	Campo del participante vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el participante no puede estar vacío

Tabla CXIX: Prueba 89

Prueba 90: Alta de evento seleccionando un participante

Requisito testado	RF-20
Acción	Un participante seleccionado
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que seleccione al menos dos participantes

Tabla CXX: Prueba 90

Prueba 91: Editar evento

Requisito testado	RF-21
Acción	El administrador selecciona la opción de editar un evento del sistema
Checkpoint 1	El sistema recibe los nuevos datos del evento y conduce al administrador al listado

Tabla CXXI: Prueba 91

Prueba 92: Editar evento con el campo del nombre vacío

Requisito testado	RF-21
Acción	Campo del nombre vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre no puede estar vacío

Tabla CXXII: Prueba 92

Prueba 93: Editar evento con el campo del nombre incorrecto

Requisito testado	RF-21
Acción	Longitud del campo del nombre incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre dispone de una longitud mínima y máxima

Tabla CXXIII: Prueba 93

Prueba 94: Editar evento con el campo de la fecha vacío

Requisito testado	RF-21
Acción	Campo de la fecha vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la fecha no puede estar vacía

Tabla CXXIV: Prueba 94

Prueba 95: Editar evento con el campo de la hora vacío

Requisito testado	RF-21
Acción	Campo de la hora vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la hora no puede estar vacía

Tabla CXXV: Prueba 95

Prueba 96: Editar evento con el campo del lugar vacío

Requisito testado	RF-21
Acción	Campo del lugar vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el lugar no puede estar vacío

Tabla CXXVI: Prueba 96

Prueba 97: Editar evento con el campo del lugar incorrecto

Requisito testado	RF-21
Acción	Longitud del campo del lugar incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el lugar dispone de una longitud mínima y máxima

Tabla CXXVII: Prueba 97

Prueba 98: Editar evento con el campo de la ciudad vacío

Requisito testado	RF-21
Acción	Campo de la ciudad vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la ciudad no puede estar vacía

Tabla CXXVIII: Prueba 98

Prueba 99: Editar evento con el campo de la ciudad incorrecto

Requisito testado	RF-21
Acción	Longitud del campo de la ciudad incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la ciudad dispone de una longitud mínima y máxima

Tabla CXXIX: Prueba 99

Prueba 100: Editar evento con el campo del participante vacío

Requisito testado	RF-21
Acción	Campo del participante vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el participante no puede estar vacío

Tabla CXXX: Prueba 100

Prueba 101: Editar evento seleccionando un participante

Requisito testado	RF-21
Acción	Un participante seleccionado
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que seleccione al menos dos participantes

Tabla CXXXI: Prueba 101

Prueba 102: Eliminar evento

Requisito testado	RF-22
Acción	El administrador accede a la gestión de eventos y selecciona la opción de eliminar un evento
Checkpoint 1	El sistema muestra una notificación de confirmación de la operación
Acción	El administrador pulsa sobre el botón de eliminar
Checkpoint 2	El sistema elimina la información de dicho evento

Tabla CXXXII: Prueba 102

Prueba 103: Listado de eventos

Requisito testado	RF-23
Acción	El administrador accede a la pantalla del listado de eventos
Checkpoint 1	El sistema muestra un listado con todos los eventos

Tabla CXXXIII: Prueba 103

Prueba 104: Visualizar evento

Requisito testado	RF-24
Acción	El administrador accede a la pantalla del listado de eventos
Checkpoint 1	El sistema muestra un listado con todos los eventos
Acción	El administrador pulsa sobre el botón de visualizar un evento determinado
Checkpoint 2	El sistema muestra la información perteneciente al evento

Tabla CXXXIV: Prueba 104

Prueba 105: Alta de entrada

Requisito testado	RF-25
Acción	El administrador rellena el formulario de alta de una nueva entrada
Checkpoint 1	El sistema recibe los datos, crea la nueva entrada y conduce al administrador al listado

Tabla CXXXV: Prueba 105

Prueba 106: Alta de entrada con el campo de la descripción vacío

Requisito testado	RF-25
Acción	Campo de la descripción vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la descripción no puede estar vacía

Tabla CXXXVI: Prueba 106

Prueba 107: Alta de entrada con el campo de la descripción incorrecto

Requisito testado	RF-25
Acción	Longitud del campo de la descripción incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la descripción dispone de una longitud mínima y máxima

Tabla CXXXVII: Prueba 107

Prueba 108: Alta de entrada con el campo del precio incorrecto

Requisito testado	RF-25
Acción	Valor incorrecto del campo precio
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el precio dispone de un valor mínimo

Tabla CXXXVIII: Prueba 108

Prueba 109: Alta de entrada con el campo del evento vacío

Requisito testado	RF-25
Acción	Longitud del campo evento vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el evento no puede estar vacío

Tabla CXXXIX: Prueba 109

Prueba 110: Editar entrada

Requisito testado	RF-26
Acción	El administrador selecciona la opción de editar una entrada del sistema
Checkpoint 1	El sistema recibe los nuevos datos de la entrada y conduce al administrador al listado

Tabla CXL: Prueba 110

Prueba 111: Editar entrada con el campo de la descripción vacío

Requisito testado	RF-26
Acción	Campo de la descripción vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la descripción no puede estar vacía

Tabla CXLI: Prueba 111

Prueba 112: Editar entrada con el campo de la descripción incorrecto

Requisito testado	RF-26
Acción	Longitud del campo de la descripción incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la descripción dispone de una longitud mínima y máxima

Tabla CXLII: Prueba 112

Prueba 113: Editar entrada con el campo del precio incorrecto

Requisito testado	RF-26
Acción	Valor incorrecto del campo precio
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el precio dispone de un valor mínimo

Tabla CXLIII: Prueba 113

Prueba 114: Eliminar entrada

Requisito testado	RF-27
Acción	El administrador selecciona la opción de eliminar una entrada
Checkpoint 1	El sistema elimina la información de dicha entrada

Tabla CXLIV: Prueba 114

Prueba 115: Listado de entradas

Requisito testado	RF-28
Acción	El administrador accede a la pantalla del listado de entradas
Checkpoint 1	El sistema muestra un listado con todas las entradas

Tabla CXLV: Prueba 115

Prueba 116: Puesta en venta de las entradas de un evento

Requisito testado	RF-29
Acción	El administrador acciona el botón habilitado para la puesta en venta y selecciona el evento correspondiente
Checkpoint 1	El sistema recibe los datos, pone las entradas del evento seleccionado a la venta y conduce al administrador al listado

Tabla CXLVI: Prueba 116

Prueba 117: Visualizar compra

Requisito testado	RF-30
Acción	El administrador accede a visualizar una compra determinada del listado
Checkpoint 1	El sistema muestra toda la información referente a la compra

Tabla CXLVII: Prueba 117

Prueba 118: Listado de compras

Requisito testado	RF-31
Acción	El administrador accede a la pantalla del listado de compras
Checkpoint 1	El sistema muestra un listado con todas las compras realizadas por los clientes

Tabla CXLVIII: Prueba 118

Prueba 119: Editar cliente

Requisito testado	RF-32
Acción	El cliente accede a datos personales, selecciona la opción de editar y modifica los campos habilitados
Checkpoint 1	El sistema recibe los nuevos datos del cliente y lo conduce a la pantalla de datos personales

Tabla CXLIX: Prueba 119

Prueba 120: Editar cliente con el campo del nombre vacío

Requisito testado	RF-32
Acción	Campo del nombre vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre no puede estar vacío

Tabla CL: Prueba 120

Prueba 121: Editar cliente con el campo del nombre incorrecto

Requisito testado	RF-32
Acción	Longitud del campo del nombre incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre dispone de una longitud mínima y máxima

Tabla CLI: Prueba 121

Prueba 122: Editar cliente con el campo del correo vacío

Requisito testado	RF-32
Acción	Campo del correo vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el correo no puede estar vacío

Tabla CLII: Prueba 122

Prueba 123: Editar cliente con el campo del correo incorrecto

Requisito testado	RF-32
Acción	Longitud del campo del correo incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la dirección de correo no es válida

Tabla CLIII: Prueba 123

Prueba 124: Editar cliente con el campo de la ciudad vacío

Requisito testado	RF-32
Acción	Campo de la ciudad vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la ciudad no puede estar vacía

Tabla CLIV: Prueba 124

Prueba 125: Editar cliente con el campo de la ciudad incorrecto

Requisito testado	RF-32
Acción	Longitud del campo de la ciudad incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la ciudad dispone de una longitud mínima y máxima

Tabla CLV: Prueba 125

Prueba 126: Editar cliente con el campo del nombre de usuario vacío

Requisito testado	RF-32
Acción	Campo del nombre de usuario vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre de usuario no puede estar vacío

Tabla CLVI: Prueba 126

Prueba 127: Editar cliente con el campo del nombre de usuario incorrecto

Requisito testado	RF-32
Acción	Longitud del campo del nombre de usuario incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el nombre de usuario dispone de una longitud mínima y máxima

Tabla CLVII: Prueba 127

Prueba 128: Editar cliente con el campo del nombre de usuario existente en la base de datos

Requisito testado	RF-32
Acción	Campo del nombre de usuario existente
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que ya existe un usuario con ese nombre de usuario

Tabla CLVIII: Prueba 128

Prueba 129: Editar cliente con el campo de la clave vacío

Requisito testeado	RF-32
Acción	Campo de la clave vacío
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la clave no puede estar vacía

Tabla CLIX: Prueba 129

Prueba 130: Editar cliente con el campo de la clave incorrecto

Requisito testeado	RF-32
Acción	Longitud del campo de la clave incorrecta
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que la clave dispone de una longitud mínima y máxima

Tabla CLX: Prueba 130

Prueba 131: Eliminar cliente

Requisito testado	RF-33
Acción	El cliente accede a datos personales y acciona el botón de eliminar
Checkpoint 1	El sistema muestra una notificación de confirmación de la operación
Acción	El cliente pulsa sobre el botón de eliminar
Checkpoint 2	El sistema elimina la información de dicho cliente y lo conduce a la pantalla de inicio de la aplicación

Tabla CLXI: Prueba 131

Prueba 132: Visualizar datos personales cliente

Requisito testado	RF-34
Acción	El cliente accede a datos personales para visualizar la información de su perfil
Checkpoint 1	El sistema muestra todos los datos referentes al perfil del cliente

Tabla CLXII: Prueba 132

Prueba 133: Buscar eventos

Requisito testado	RF-35
Acción	Una vez logueado, el cliente dispone de dos alternativas: por un lado puede acceder al buscador situado en la parte superior izquierda de cualquier pantalla de la aplicación e introducir uno de los cuatro criterios de búsqueda soportados por el sistema: ciudad, deporte, competición y participante; por otro lado puede pulsar sobre la opción “Ver Eventos” habilitada en el menú, en cuyo caso tendrá que seleccionar varias categorías hasta llegar al evento deseado
Checkpoint 1	El sistema muestra los eventos que coinciden con el criterio de búsqueda introducido por el cliente o con las categorías seleccionadas

Tabla CLXIII: Prueba 133

Prueba 134: Buscar eventos sin introducir ningún criterio de búsqueda

Requisito testado	RF-35
Acción	El cliente no introduce en el buscador ningún criterio de búsqueda y acciona el botón de buscar
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que se introduzca un criterio de búsqueda, además de señalar que no hay eventos disponibles con ese criterio

Tabla CLXIV: Prueba 134

Prueba 135: Introducir un criterio de búsqueda no válido

Requisito testado	RF-35
Acción	El cliente introduce en el buscador un criterio de búsqueda erróneo
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que no hay eventos disponibles con ese criterio

Tabla CLXV: Prueba 135

Prueba 136: Ver compras realizadas

Requisito testado	RF-36
Acción	El cliente accede a visualizar su histórico de compras
Checkpoint 1	El sistema muestra un listado con información parcial sobre las compras realizadas por el cliente, además de ofrecer a éste la posibilidad de visualizar toda la información acerca de cada una de ellas mediante la habilitación de los botones correspondientes

Tabla CLXVI: Prueba 136

Prueba 137: Comprar entradas

Requisito testado	RF-37
Acción	El usuario accede al buscador o a la opción "Ver Eventos" del menú con el objetivo de localizar un evento determinado
Checkpoint 1	El sistema muestra los eventos que coinciden con el criterio de búsqueda introducido por el cliente o con las categorías seleccionadas
Acción	El usuario pulsa sobre el botón habilitado para ver las entradas del evento
Checkpoint 2	El sistema presenta un listado que contiene las entradas que aún no han sido compradas por ningún cliente
Acción	El usuario selecciona las entradas que desea y acciona el botón de comprar
Checkpoint 3	El sistema muestra una pantalla con los datos de la compra realizada, además de incluir las opciones de cancelación o pago de la compra

Tabla CLXVII: Prueba 137

Prueba 138: Realizar una compra sin seleccionar ninguna entrada

Requisito testado	RF-37
Acción	El usuario accede al buscador o a la opción "Ver Eventos" del menú con el objetivo de localizar un evento determinado
Checkpoint 1	El sistema muestra los eventos que coinciden con el criterio de búsqueda introducido por el cliente o con las categorías seleccionadas
Acción	El usuario pulsa sobre el botón habilitado para ver las entradas del evento
Checkpoint 2	El sistema presenta un listado que contiene las entradas que aún no han sido compradas por ningún cliente
Acción	El usuario no selecciona ninguna entrada y acciona el botón de comprar
Checkpoint 3	El sistema debe mostrar un mensaje de error indicando que seleccione alguna entrada

Tabla CLXVIII: Prueba 138

Prueba 139: Pagar entradas

La prueba se describe a partir de que el usuario haya comprado alguna entrada para un evento en concreto.

Requisito testado	RF-38
Acción	El usuario acciona el botón de pago habilitado en la confirmación de la compra
Checkpoint 1	El sistema redirige al usuario a PayPal, donde se muestra un resumen de la compra y el formulario de acceso
Acción	El usuario introduce sus credenciales
Checkpoint 2	PayPal valida los datos y habilita al usuario el botón de pago
Acción	El usuario autoriza el pago
Checkpoint 3	PayPal redirige al usuario a la aplicación, donde se mostrará la confirmación de pago y los datos de la compra

Tabla CLXIX: Prueba 139

Prueba 140: Pagar entradas sin introducir correctamente las credenciales en PayPal

Requisito testado	RF-38
Acción	El usuario acciona el botón de pago habilitado en la confirmación de la compra
Checkpoint 1	El sistema redirige al usuario a PayPal, donde se muestra un resumen de la compra y el formulario de acceso
Acción	El usuario introduce sus credenciales de manera errónea
Checkpoint 2	PayPal muestra un mensaje de error

Tabla CLXX: Prueba 140

Prueba 141: Imprimir entradas

Requisito testeado	RF-39
Acción	El usuario accede a la opción "Ver Compras Realizadas" del menú
Checkpoint 1	El sistema muestra las compras realizadas por el usuario
Acción	El usuario pulsa sobre el botón habilitado para imprimir las entradas de la compra en cuestión
Checkpoint 2	El sistema redirige al usuario a la confirmación de la compra, donde se muestran los datos de la misma y el botón preparado para imprimir
Acción	El usuario acciona el botón "Imprimir entradas"
Checkpoint 3	El sistema habilita la ventana de impresión
Acción	El usuario pulsa sobre el botón de imprimir
Checkpoint 4	El sistema imprime las entradas

Tabla CLXXI: Prueba 141

Prueba 142: Imprimir las entradas una vez haya pasado el tiempo permitido

Requisito testado	RF-39
Acción	El usuario accede a la opción "Ver Compras Realizadas" del menú
Checkpoint 1	El sistema muestra las compras realizadas por el usuario
Acción	El usuario pulsa sobre el botón habilitado para imprimir las entradas de la compra en cuestión
Checkpoint 2	El sistema debe mostrar un mensaje de error indicando que se ha superado el tiempo máximo habilitado para imprimir las entradas de una compra determinada

Tabla CLXXII: Prueba 142

4.3.2. Resultados obtenidos

Las pruebas realizadas ofrecen resultados satisfactorios, por lo que puede concluirse que el software desarrollado proporciona al usuario la funcionalidad deseada, ya que se ha verificado que cubren todos los requerimientos funcionales del sistema, especificados en el [capítulo 2.5.1](#) de la memoria.

5. Conclusiones

Para finalizar esta memoria, podemos concluir que se han cumplido los objetivos y propósitos propuestos al inicio de este proyecto, además de las expectativas personales puestas sobre su realización.

En primer lugar se ha realizado un estudio de necesidades para la venta de entradas a eventos deportivos a través de Internet. También se ha llevado a cabo un análisis sobre el uso de Paypal como pasarela de pago para comercio electrónico. A partir del estudio desarrollado se ha diseñado e implementado un prototipo de aplicación web que permite gestionar la venta de localidades de eventos deportivos a través de Internet.

Desde el punto de vista personal, el proyecto ha supuesto un gran reto para mí. Inicialmente, sentí un poco de pánico al enfrentarme yo solo a un proyecto de tal envergadura. Poco a poco fui adentrándome en las tecnologías utilizadas, tomándome su aprendizaje como un reto a corto plazo para finalmente poder desarrollar el proyecto. Estos últimos meses han supuesto un gran esfuerzo personal, pero la experiencia ha sido muy enriquecedora.

Gracias a la realización del proyecto, he puesto en práctica muchas de las destrezas y metodologías adquiridas durante los años de formación académica, además de confirmar el uso diario por parte del ser humano de aplicaciones web orientadas a la venta de entradas, en mi caso deportivas, lo que me brinda una gran oportunidad de futuro. Aprovechándonos de las soluciones estudiadas y de las necesidades existentes en la actualidad, unido a los mecanismos escogidos, hemos llevado a cabo la ejecución de la propuesta.

Desde el punto de vista del desarrollo, se han seguido las fases de la ingeniería del software. Cualquier proyecto software contiene un número de actividades, independientemente de su tamaño o complejidad, que garantizan la adquisición de productos de calidad. En primer lugar, se lleva a cabo la etapa de análisis. En ella, se definen de forma precisa los requerimientos que debe cumplir el sistema, además de obtenerse el dominio de información del software. La etapa siguiente consiste en la realización del diseño detallado del sistema. Esta fase se centra en proporcionar la solución técnica más apropiada que cumpla los requerimientos. Finalmente, la última etapa hace referencia a la implementación y las pruebas del software, las

cuales verifican y validan el funcionamiento del sistema desarrollado. Además, respecto a las tecnologías usadas en esta última fase, he profundizado en mi conocimiento sobre el framework Spring MVC y la utilización de un sistema ORM que nos permite establecer conexión con la base de datos y comenzar a realizar operaciones con los mismos.

Como conclusión final, espero que todo lo aprendido me sea de gran utilidad para el desarrollo de trabajos posteriores a lo largo de mi carrera profesional.

5.1. Mejoras y trabajos futuros

Cabe destacar que la funcionalidad de la aplicación web puede no ser completa, ya que se se ha llevado a cabo la implementación de un prototipo. En este apartado se citan varias mejoras y ampliaciones, las cuales quedan fuera del alcance del proyecto por su duración y extensión. Sin embargo, desde mi punto de vista, se trata de un proyecto realizado satisfactoriamente, ya que las funciones definidas en la especificación de requerimientos están implementadas.

A continuación se plantean una serie de mejoras que podrían ser abordadas en el futuro:

- Incorporar información adicional relacionada con las entradas que indique la fila, el número de asiento y el sector al que pertenecen.
- Disponer de nuevos métodos de búsqueda de eventos que mejoren los utilizados actualmente. Por ejemplo, sería interesante agregar filtros que permitan localizar eventos según el número de entradas disponibles, situación en el campo o precio.
- Una vez que el cliente ha seleccionado la opción de visualizar las entradas de un evento en concreto, podría integrarse un mapa que mostrase la distribución de los asientos, facilitándole al usuario la elección de éstos.
- Además de comprar las entradas establecidas por el administrador del sistema, podría ofrecerse al cliente la posibilidad de poner en venta las entradas adquiridas para que sean compradas por otros clientes.

- Habilitar en el sistema la opción de venta de entradas para tours de estadios.

Al disponer de un sistema robusto y escalable, el modelo permite extenderse para añadir estos elementos de manera sencilla.

6. Bibliografía

Bibliografía

- [1] «PayPal,» [En línea]. Available: <https://www.paypal.com/es/home>.
- [2] R. S. Pressman, Ingeniería del Software. Un enfoque práctico, Quinta Edición. Mc Graw Hill, 2002.
- [3] «What is Scrum?,» [En línea]. Available: <https://www.scrum.org/resources/what-is-scrum>.
- [4] S. Luján Mora, Programación de aplicaciones web: historia, principios básicos y clientes web., 2002.
- [5] «Ticketmaster,» [En línea]. Available: <http://www.ticketmaster.es/es/entradas-deportes/>.
- [6] «Tengoentradas.com,» [En línea]. Available: <https://www.tengoentradas.com/Deportes.html>.
- [7] «Atrápalo,» [En línea]. Available: <http://www.atrapalo.com/entradas/deportes/>.
- [8] «Viagogo,» [En línea]. Available: <http://www.viagogo.es/Entradas-Deportes>.
- [9] I. PayPal, «Guía de integración de Pasarela integral,» 2014.
- [10] J. R. Balsas Almagro, «Apuntes de la asignatura Desarrollo de Aplicaciones Web,» Jaén.
- [11] «Spring Framework,» [En línea]. Available: <http://projects.spring.io/spring-framework/>.
- [12] A. J. Rueda Ruiz, «Apuntes de la asignatura Desarrollo de Aplicaciones Empresariales,» Jaén.
- [13] «JavaServer Faces Technology,» [En línea]. Available: <http://www.java-serverfaces.org/>.
- [14] R. Lubke, J. Ball y P. Delisle, Unified Expression Language, 2005.
- [15] «Bootstrap,» [En línea]. Available: <http://getbootstrap.com/>.
- [16] «Pure.css,» [En línea]. Available: <http://purecss.io/>.
- [17] M. J. Barranco García, «Apuntes de la asignatura Ingeniería de Requisitos,» Jaén.
- [18] V. M. Rivas Santos, «Apuntes de la asignatura Interacción Persona-Ordenador,» Jaén.
- [19] L. Martínez López, «Apuntes de la asignatura Gestión y Control de Proyectos

Informáticos,» Jaén.

- [20] «Agencia Estatal Boletín Oficial del Estado,» [En línea]. Available:
<http://www.boe.es/boe/dias/2015/01/12/pdfs/BOE-A-2015-255.pdf>.
- [21] C. Larman, UML y Patrones, Segunda Edición, Prentice Hall, 2013.
- [22] L. A. Ureña López y J. I. Gómez Espínola, «Apuntes de la asignatura Fundamentos de Ingeniería del Software,» Jaén.
- [23] A. Molina Aguilar, «Apuntes de la asignatura Bases de Datos I,» Jaén.
- [24] «Best MVC Practices,» [En línea]. Available:
<http://www.yiiframework.com/doc/guide/1.1/en/basics.best-practices> .
- [25] «Stack Overflow,» [En línea]. Available:
<http://stackoverflow.com/questions/10138715/objects-binding-in-spring-mvc-form>.
- [26] «Definición de Layout,» [En línea]. Available: <http://definicion.de/layout/> .
- [27] P. Morville y L. Rosenfeld, Information Architecture for the World Wide Web, 1998.
- [28] «Manual oficial de Bootstrap. Iconos,» [En línea]. Available:
http://librosweb.es/libro/bootstrap_3/capitulo_6/iconos_glyphicons.html .
- [29] «Tutorial JSP,» [En línea]. Available:
<http://docs.oracle.com/javase/5/tutorial/doc/bnagx.html> .
- [30] «HTML Tutorial,» [En línea]. Available: <http://www.w3schools.com/html/>.
- [31] «Introducción a CSS,» [En línea]. Available: <https://librosweb.es/libro/css/>.
- [32] «Tutorial JSTL,» [En línea]. Available:
<http://docs.oracle.com/javase/5/tutorial/doc/bnakc.html> .
- [33] D. Marco, «Tutorial de JPA,» [En línea]. Available:
<http://www.davidmarco.es/archivo/tutorial-jpa20>.
- [34] «Introducción a jQuery,» [En línea]. Available: <https://jquery.com/> .
- [35] «Manual oficial de Bootstrap. Tipos de rejillas,» [En línea]. Available:
http://librosweb.es/libro/bootstrap_3/capitulo_2/tipos_de_rejillas.html .
- [36] «Manual oficial de Bootstrap. Barras de navegación,» [En línea]. Available:
http://librosweb.es/libro/bootstrap_3/capitulo_6/barras_de_navegacion.html .
- [37] «Bootstrap Tutorial,» [En línea]. Available:
http://www.java2s.com/Tutorials/HTML_CSS/Bootstrap_Tutorial/Bootstrap_Dialog.htm.

- [38] «Manual oficial de Bootstrap. Tablas,» [En línea]. Available:
http://librosweb.es/libro/bootstrap_3/capitulo_4/tablas.html .
- [39] «Apache Derby,» [En línea]. Available: <http://db.apache.org/derby/> .
- [40] «Apache Tomcat,» [En línea]. Available: <http://tomcat.apache.org>.
- [41] «Códigos QR,» [En línea]. Available: <http://www.qrcode.com/en/>.
- [42] V. M. Rivas Santos, «Apuntes de la asignatura Programación Orientada a Objetos,»
Jaén.

Apéndice I. Registro de cuenta en PayPal

Paypal contiene un sitio de pruebas, conocido como PayPal Sandbox. Este módulo proporciona un entorno de simulación de PayPal, en el que se ejecuta su integración de pago exprés sin llegar a intercambiar dinero. Para ello, se necesitarán dos cuentas de prueba: la primera representa al comerciante o vendedor y la segunda a un comprador.

Para crear una cuenta de PayPal Sandbox se requiere previamente una cuenta existente de PayPal, asique accedemos a <https://developer.paypal.com/> para crear dicha cuenta y presionamos el botón **Sign Up**. Una vez ahí, rellenamos el formulario que aparece a continuación:

Sign up for a business account

Our REST APIs support a number of [countries and currencies](#). If you're a non-US developer and would like to use them, please sign up on [paypal.com](#).

Create a log in

Enter your business contact information



By clicking the button below, I agree to the [PayPal User Agreement](#), [Privacy Policy](#), [Acceptable Use Policy](#), [The Electronic Communications Delivery Policy](#), [PayPal Developer Agreement](#).

Ilustración 118: Formulario alta cuenta

Tras enviar el formulario, se crea la cuenta del comprador (rb389-buyer@hotmail.com). El servicio envía un correo a la dirección de email indicada en

el registro para confirmar la cuenta. Para empezar a recibir pagos hay que confirmar la misma. Para ello, hacemos click en el enlace proporcionado en el correo y nos aparece:

Enter Password

Please enter your PayPal password to confirm your email address.

PayPal Password:

Login

[Forgot your password?](#)

Ilustración 119: Confirmación cuenta

Nuevamente, recibimos otro correo indicando que el correo electrónico está confirmado y que ya hay acceso al sitio de desarrolladores de PayPal con la posibilidad de probar las herramientas.

Accedemos al sitio de pruebas mediante la dirección proporcionada anteriormente para la creación de la cuenta de PayPal, pero esta vez accionamos el botón **Log In**. Una vez introducidos los credenciales, nos aparecerá la siguiente pantalla:

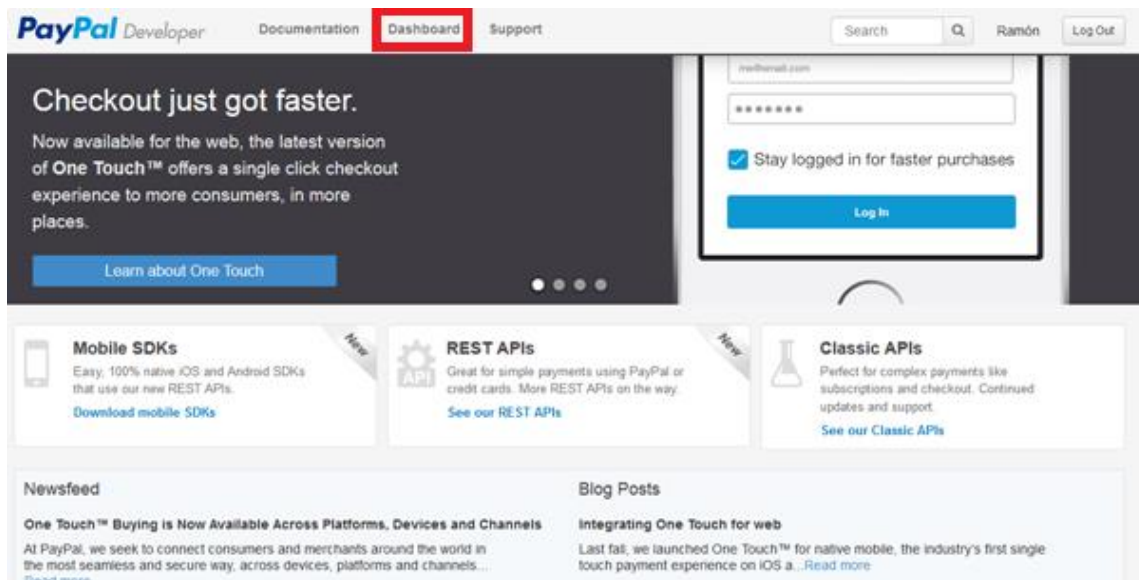


Ilustración 120: Inicio sitio pruebas

A continuación seleccionamos la opción **Dashboard**, marcada en la ilustración anterior y posteriormente pulsamos sobre la primera alternativa del menú Sandbox, denominada **Accounts**, tal y como se muestra en la imagen siguiente:

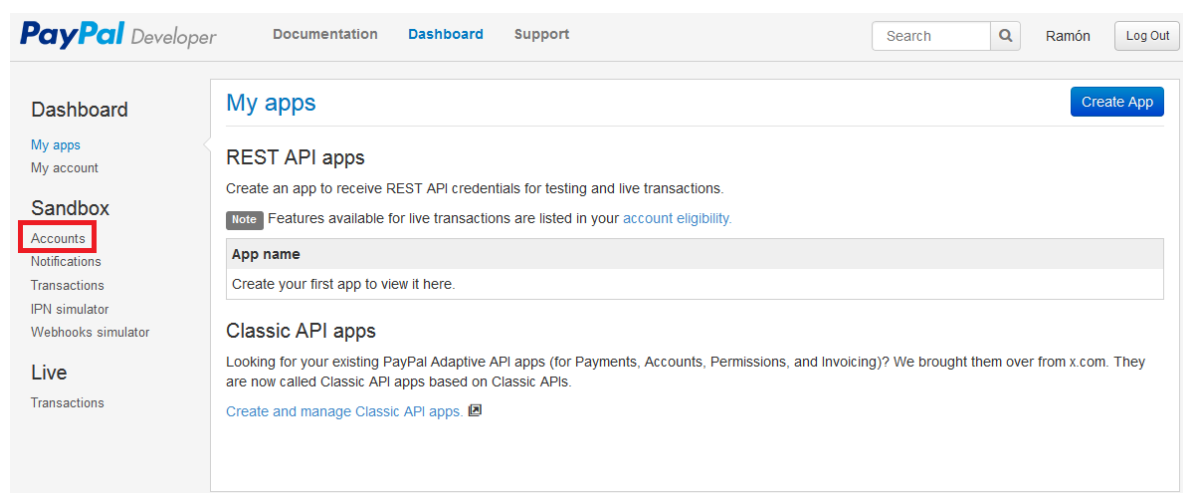


Ilustración 121: Opción Accounts

Una vez ahí, comprobamos la existencia de dos cuentas diferentes (cuadro de color negro insertado en la imagen siguiente), la previamente existente para el comprador (**personal**) y la nueva para el comerciante o vendedor (**business**), marcadas en la ilustración de color amarillo.

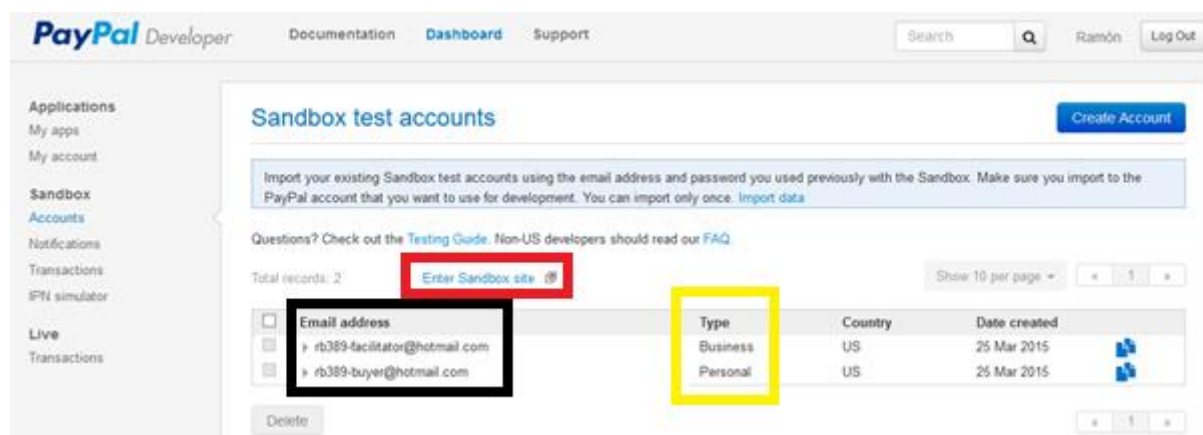


Ilustración 122: Cuentas

Se recomienda cambiar las contraseñas por defecto de las cuentas creadas. Para ello, pinchamos sobre cada una de ellas y seleccionamos la opción **Profile**. El siguiente paso consiste en acceder a la opción **Change password**, tal y como indica la ilustración siguiente, y cambiar la contraseña.

Account details ✕

Profile	Funding	Settings
Email ID:	rb389-buyer@hotmail.com	
Password:	Change password	
Phone number:	4087008391	
Account type:	Personal	
Status:	Verified	
Country:	US	
Notes:		

[Close](#)

Ilustración 123: Account details

Finalmente, para acceder al sitio de prueba pinchamos en **Enter Sandbox site** (cuadro rojo de la ilustración Cuentas) e introducimos la dirección de email del comprador (rb389-buyer@hotmail.com) o bien la del vendedor (rb389-facilitator@hotmail.com), con sus respectivas contraseñas.

Apéndice II. Manual de Usuario

En este apéndice se va a describir de forma detallada el manual de usuario de la aplicación web desarrollada para la gestión de venta de localidades. El principal objetivo es resolver cualquier duda que el usuario pueda tener a la hora de utilizar la aplicación.

A continuación se van a detallar cada una de las opciones disponibles en la aplicación, teniendo en cuenta la existencia de los roles de administrador y cliente.

- **Acceso a la aplicación**

La pantalla principal de la aplicación es la siguiente:

Venta de Localidades Deportivas

Regístrese

Nombre:

Correo:

Ciudad:

Usuario:

Clave:

¿Es mayor de 18 años? ☐

Guardar

Trabajo Fin de Grado
Curso 2014-2015
Grado en Ingeniería Informática
Escuela Politécnica Superior de Jaén

© Ramón Bailén Sánchez

Ilustración 124: Pantalla principal

La pantalla de inicio contiene los formularios de registro y login del sistema. Para que un usuario acceda al contenido de la aplicación tiene que registrarse previamente, rellenando los campos que aparecen en la ilustración.

- **Rol administrador**

Un usuario administrador es aquel que tiene la posibilidad de crear nuevos usuarios con rol administrador o asignar este permiso a usuarios existentes. Para iniciar sesión, introducirá sus credenciales en el formulario de login. La ilustración siguiente muestra la funcionalidad que puede realizar un usuario con este rol:

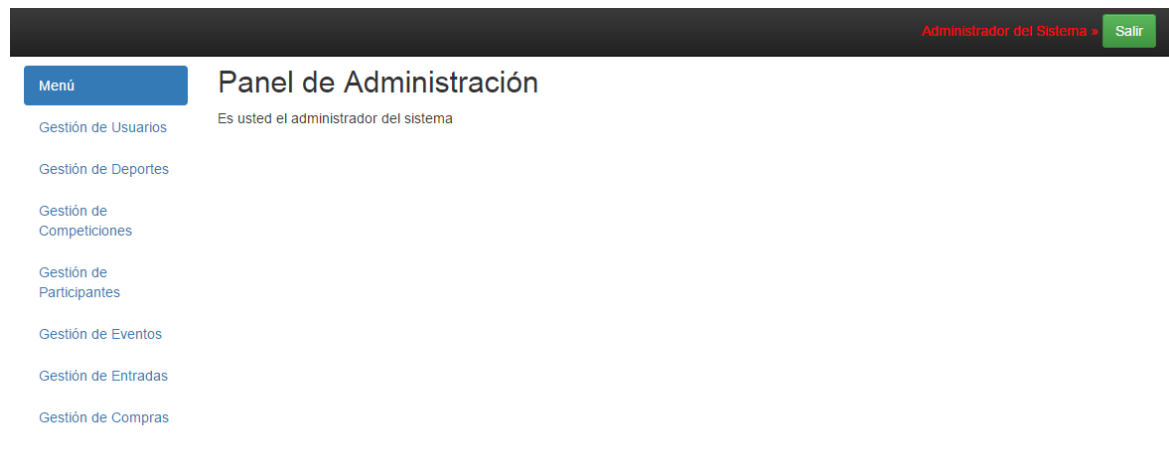


Ilustración 125: Menú Administrador

En cualquier momento, el usuario administrador podrá finalizar su sesión actual pulsando en el botón “Salir” situado en la esquina superior derecha. A continuación se detallan cada una de las opciones disponibles del menú:

- **Gestión de usuarios**

El usuario administrador puede gestionar los usuarios existentes en el sistema, es decir, tiene la capacidad de crear nuevos usuarios, modificar la información de cada uno de ellos o bien eliminarlos.

En la ilustración siguiente aparece un listado con los usuarios almacenados actualmente en el sistema, con su respectiva información:



Ilustración 126: Gestión de usuarios

Como puede observarse en la columna rol, existe un usuario de tipo administrador y el resto son de tipo usuario, es decir, clientes potenciales del sistema.

Cada fila de la tabla se corresponde con un usuario del sistema. En cada una de ellas, se encuentran los botones pertinentes para editar o eliminar el usuario oportuno. Si el administrador pulsa el botón de eliminar, el sistema mostrará un mensaje indicando la confirmación de la operación, con las opciones de cancelar o borrar a dicho usuario de la base de datos y automáticamente se redireccionará al listado de usuarios, donde no aparecerá el usuario eliminado. Si por el contrario, desea editar un usuario, se encontrará con la ilustración inferior:

The screenshot displays a web application interface for editing a user. At the top, a dark navigation bar contains the text 'Administrador del Sistema' and a green 'Salir' button. On the left, a blue sidebar menu lists various management functions: 'Menú', 'Gestión de Usuarios', 'Gestión de Deportes', 'Gestión de Competiciones', 'Gestión de Participantes', 'Gestión de Eventos', 'Gestión de Entradas', and 'Gestión de Compras'. The main content area is titled 'Editar Usuario'. It displays the 'Id Cliente: 2' and provides input fields for 'Nombre' (filled with 'Ramón Bailén Sánchez'), 'Correo' (filled with 'rb389@hotmail.com'), 'Ciudad' (filled with 'Jaén'), 'Usuario' (filled with 'ramon'), and 'Clave' (filled with 'ramon'). At the bottom of the form are three buttons: 'Guardar' (green), 'Restablecer' (blue), and 'Volver' (dark blue).

Ilustración 127: Editar usuario

Una vez editados los campos necesarios, se pulsará sobre el botón de guardar, redireccionándose de nuevo al listado de usuarios con los cambios realizados. Cabe destacar que el identificador de usuario no se puede editar, ya que es generado automáticamente por el sistema. El botón de restablecer permite restaurar los valores por defecto antes guardar los nuevos, mientras que el botón de volver regresa al listado de usuarios.

La última funcionalidad correspondiente a la gestión de usuarios es la creación de nuevos usuarios. Para llevar a cabo esta tarea, el usuario administrador tendrá que seleccionar la opción “Nuevo Usuario”, situada al término del listado de éstos. A continuación se muestra la ilustración perteneciente a dicha alternativa:

Administrador del Sistema » Salir

Menú

- Gestión de Usuarios
- Gestión de Deportes
- Gestión de Competiciones
- Gestión de Participantes
- Gestión de Eventos
- Gestión de Entradas
- Gestión de Compras

Alta de Usuario

Nombre:

Correo:

Ciudad:

Usuario:

Clave:

Rol:

☐ ROL_ADMIN

☒ ROL_USER

Ilustración 128: Alta de usuario

Para dar de alta un nuevo usuario en el sistema, el administrador tendrá que rellenar el formulario adjuntado en la imagen anterior. Todos los campos son obligatorios. Por defecto, el rol seleccionado es “ROL_USER”, refiriéndose a los clientes que van a comprar entradas a través de la aplicación. Una vez creado un nuevo usuario, dependiendo del rol fijado, podrá acceder a una parte de la aplicación u a otra, siempre desde la pantalla de inicio utilizando los credenciales asignados.

En este momento, ningún cliente que acceda a la aplicación utilizando sus credenciales podría adquirir una entrada, ya que el sistema carece de datos sobre otras entidades. Por lo tanto, en el proceso de compra llevado a cabo por el cliente también entran en juego las siguientes funcionalidades de la aplicación:

▪ Gestión de deportes

Análogamente, el usuario administrador puede gestionar los deportes del sistema, es decir, es competencia suya la creación de nuevos deportes, modificar la información de cada uno de ellos o bien eliminarlos.

En la ilustración siguiente aparece un listado con los deportes existentes actualmente en el sistema, con su respectiva información:

Id	Nombre	Denominación Del Participante	Denominación Del Evento		
1	Fútbol	Equipos	partidos	Editar	Eliminar
2	Tenis	Tenistas	partidos	Editar	Eliminar

[+ Nuevo Deporte](#)

Ilustración 129: Listado de deportes

Cada fila de la tabla hace referencia a un deporte diferente del sistema. El procedimiento para editar, eliminar o crear un nuevo deporte es equivalente al realizado en la gestión de usuarios. A continuación se muestran algunas ilustraciones acerca de la funcionalidad mencionada:

Editar Deporte

Id Deporte: 1

Nombre:

Denominación del Participante:

Denominación del Evento:

[Guardar](#) [Restablecer](#) [Volver](#)

Ilustración 130: Editar deporte

Administrador del Sistema » Salir

Menú

Gestión de Usuarios

Gestión de Deportes

Gestión de Competiciones

Gestión de Participantes

Gestión de Eventos

Gestión de Entradas

Gestión de Compras

Alta de Deporte

Nombre:

Denominación del Participante:

Denominación del Evento:

Guardar Reestablecer Volver

Ilustración 131: Alta de deporte

■ Gestión de competiciones

El usuario administrador es el encargado de la gestión de las competiciones del sistema. Para ello, tiene la capacidad de crear nuevas competiciones, editar la información de cada uno de ellas o bien borrarlas.

A continuación se muestra una ilustración en la que aparece un listado con las competiciones existentes en el sistema, con sus respectivos datos:

Administrador del Sistema » Salir

Menú

Gestión de Usuarios

Gestión de Deportes

Gestión de Competiciones

Gestión de Participantes

Gestión de Eventos

Gestión de Entradas

Gestión de Compras

Listado de Competiciones

Id	Nombre	Deporte		
1	Champions League	Fútbol	Editar	Eliminar
2	Liga BBVA	Fútbol	Editar	Eliminar
3	Roland Garros	Tenis	Editar	Eliminar
4	US Open	Tenis	Editar	Eliminar

+ Nueva Competición

Ilustración 132: Listado de competiciones

Como puede observarse, la información relevante acerca de una competición es el nombre y el deporte al que pertenece. Por lo tanto, para crear una nueva competición es indispensable que previamente exista en el sistema el deporte al que pertenece. Además, en la gestión de las competiciones también se incluyen los botones citados anteriormente para editar, eliminar o designar una nueva competición.

A continuación se exponen las imágenes sobre los procesos de edición y alta de una competición, ya que como hemos dicho anteriormente, tras eliminar una competición se actualiza el listado anterior:

Administrador del Sistema » Salir

Menú

Gestión de Usuarios

Gestión de Deportes

Gestión de Competiciones

Gestión de Participantes

Gestión de Eventos

Gestión de Entradas

Gestión de Compras

Editar Competición

Id Competición: 1

Nombre:

Deporte:

Ilustración 133: Editar competición

Administrador del Sistema » Salir

Menú

Gestión de Usuarios

Gestión de Deportes

Gestión de Competiciones

Gestión de Participantes

Gestión de Eventos

Gestión de Entradas

Gestión de Compras

Alta de Competición

Nombre:

Deporte:

Ilustración 134: Alta de competición

■ Gestión de participantes

El usuario administrador puede gestionar los participantes almacenados en el sistema, es decir, tiene la capacidad de crear nuevos participantes, modificar los existentes o bien eliminarlos.

En la siguiente ilustración se muestra un listado con los participantes actuales del sistema, con su respectiva información:

Administrador del Sistema » Salir

Menú

- Gestión de Usuarios
- Gestión de Deportes
- Gestión de Competiciones
- Gestión de Participantes
- Gestión de Eventos
- Gestión de Entradas
- Gestión de Compras

Listado de Participantes

Id	Nombre	Deporte		
1	Real Madrid	Fútbol	Editar	Eliminar
2	FC Barcelona	Fútbol	Editar	Eliminar
3	Rafael Nadal	Tenis	Editar	Eliminar
4	Roger Federer	Tenis	Editar	Eliminar

[+ Nuevo Participante](#)

Ilustración 135: Listado de participantes

Al igual que las competiciones, cada participante contiene el nombre y el deporte al que pertenece, por lo que es esencial disponer de deportes antes de dar de alta un nuevo participante. Nuevamente, se facilita la posibilidad de modificar, eliminar o crear nuevos participantes, cuyos procesos son homólogos a los definidos para las competiciones.

▪ Gestión de eventos

Una vez determinados los deportes, las competiciones y los participantes necesarios, el siguiente paso consiste en crear los eventos deportivos. Es importante reseñar que para dar de alta un nuevo evento, además de las entidades anteriores, el sistema debe disponer previamente de países.

A continuación se muestra una ilustración en la que se presenta un listado con los eventos existentes en el sistema, con algunos datos:

Administrador del Sistema » Salir

Menú

- Gestión de Usuarios
- Gestión de Deportes
- Gestión de Competiciones
- Gestión de Participantes
- Gestión de Eventos
- Gestión de Entradas
- Gestión de Compras

Listado de Eventos

Id	Nombre	Fecha	Hora	Ciudad	Pais			
1	Real Madrid Vs FC Barcelona	2015-04-18	20:00	Madrid	España	Visualiza	Editar	Eliminar

[+ Nuevo Evento](#)

Ilustración 136: Listado de eventos

Además de la funcionalidad descrita anteriormente para otras entidades, el usuario administrador dispone de un botón para visualizar todos los datos acerca de un evento, ya que el listado muestra solamente información parcial acerca del mismo. Seguidamente se muestra una ilustración referente a la visualización de un evento:

The screenshot shows the 'Visualizar Evento' page. At the top right, there is a dark header with 'Administrador del Sistema »' in red and a green 'Salir' button. On the left, a blue 'Menú' button is above a list of menu items: 'Gestión de Usuarios', 'Gestión de Deportes', 'Gestión de Competiciones', 'Gestión de Participantes', 'Gestión de Eventos', 'Gestión de Entradas', and 'Gestión de Compras'. The main content area is titled 'Visualizar Evento' and displays the following event details: 'Id Evento: 1', 'Nombre: Real Madrid Vs FC Barcelona', 'Fecha: 2015-04-18', 'Hora: 20:00', 'Lugar: Santiago Bernabéu', 'Ciudad: Madrid', 'País: España', 'Número de Entradas: 3', 'Entradas puestas a la Venta: No', 'Deporte: Fútbol', 'Competición: Liga BBVA', and 'Participantes: Real Madrid, FC Barcelona'. A blue 'Volver' button is located at the bottom left of the event details.

Ilustración 137: Visualizar evento

Para crear un evento, el usuario administrador debe rellenar el siguiente formulario:

The screenshot shows the 'Alta de Evento' form. At the top right, there is a dark header with 'Administrador del Sistema »' in red and a green 'Salir' button. On the left, a blue 'Menú' button is above a list of menu items: 'Gestión de Usuarios', 'Gestión de Deportes', 'Gestión de Competiciones', 'Gestión de Participantes', 'Gestión de Eventos', 'Gestión de Entradas', and 'Gestión de Compras'. The main content area is titled 'Alta de Evento' and contains the following form fields: 'Nombre:' (text input), 'Fecha:' (text input with format 'dd/mm/aaaa'), 'Hora:' (text input with format '--:--'), 'Lugar:' (text input), 'Ciudad:' (text input), 'País:' (dropdown menu with '-- Seleccionar país --'), 'Número de Entradas:' (text input with value '0'), 'Deporte:' (dropdown menu with '-- Seleccionar deporte --'), 'Competición:' (dropdown menu with '-- Seleccionar competición --'), and 'Participantes:' (checkboxes for 'Real Madrid', 'FC Barcelona', 'Rafael Nadal', and 'Roger Federer'). At the bottom right, there are three buttons: 'Guardar' (green), 'Reestablecer' (blue), and 'Volver' (blue).

Ilustración 138: Alta de evento

Puede verse que para dar de alta un evento hay que rellenar algunos campos, como por ejemplo el nombre del evento y seleccionar otros, tales como el país o los participantes del mismo, por citar varios.

Un aspecto a destacar sobre el proceso de creación de un evento es el campo referente al número de entradas. Una vez rellenado el formulario, incluido el número de entradas para el evento en cuestión, el siguiente paso consiste en accionar el botón de guardar. En casos anteriores, una vez creada una nueva entidad, el sistema redireccionaba al listado de dicha entidad. Eso ocurre cuando el número de entradas indicado es 0. Sin embargo, si se indica un número de entradas mayor que 0, el sistema no actuará de forma análoga, sino que conducirá al usuario administrador al listado de entradas, donde podrá editar las entradas creadas.

La gestión de eventos también ofrece la posibilidad al administrador de modificar la información correspondiente a cualquiera de ellos, como a continuación se muestra en la siguiente pantalla:

Administrador del Sistema » Salir

Menú

- Gestión de Usuarios
- Gestión de Deportes
- Gestión de Competiciones
- Gestión de Participantes
- Gestión de Eventos
- Gestión de Entradas
- Gestión de Compras

Editar Evento

Id Evento: 1

Nombre:
Real Madrid Vs FC Barcelona

Fecha:
18/04/2015

Hora:
20:00

Lugar:
Santiago Bernabéu

Ciudad:
Madrid

País:
España

Deporte:
Fútbol

Competición:
Liga BBVA

Participantes:

- ☒ Real Madrid
- ☒ FC Barcelona
- ☐ Rafael Nadal
- ☐ Roger Federer

Guardar **Restablecer** **Volver**

Ilustración 139: Editar evento

Cabe destacar que una vez que las entradas del evento estén puestas a la venta, no se podrá llevar a cabo la edición de dicho evento.

▪ Gestión de entradas

La gestión de entradas ofrece al usuario administrador los medios necesarios para dar de alta una entrada, modificar la información correspondiente a cualquiera de éstas, eliminarlas y ponerlas en venta.

En la ilustración siguiente se muestra un ejemplo sobre las entradas pertenecientes a un evento:

Id	Descripción	Precio	Evento	Comprada		
1	Entrada 1 para evento: Real Madrid Vs FC Barcelona	30.0 €	Real Madrid Vs FC Barcelona	No	Editar	Eliminar
2	Entrada 2 para evento: Real Madrid Vs FC Barcelona	45.0 €	Real Madrid Vs FC Barcelona	No	Editar	Eliminar
3	Entrada 3 para evento: Real Madrid Vs FC Barcelona	80.0 €	Real Madrid Vs FC Barcelona	No	Editar	Eliminar

[+ Nueva Entrada](#) [Puesta en Venta](#)

Ilustración 140: Listado de entradas

De igual forma, a continuación se muestran gráficamente los procesos de creación y edición de una entrada:

Alta de Entrada

Descripción:

Precio:

Evento:

[Guardar](#) [Reestablecer](#) [Volver](#)

Ilustración 141: Alta de entrada

Es importante comentar que las entradas se crean al dar de alta los diferentes eventos deportivos. Sin embargo, esta opción resulta interesante para el usuario administrador si desea añadir nuevas entradas a un evento determinado. Es conveniente utilizar en el campo “descripción” la configuración definida para las demás entradas, siguiendo la ordenación numérica establecida por defecto tras la creación del evento.

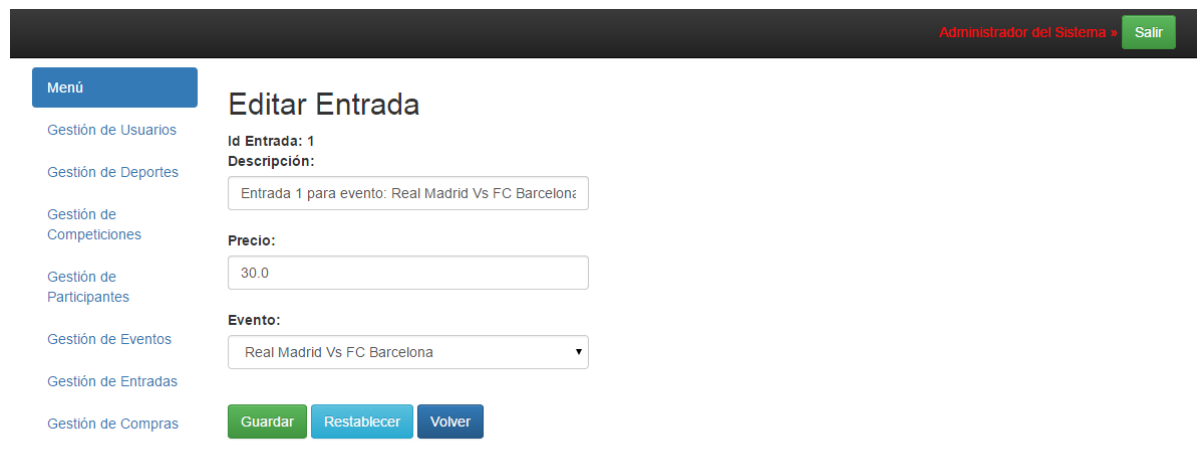


Ilustración 142: Editar entrada

Respecto a la edición, de manera similar al funcionamiento del listado de eventos, una vez que las entradas de un evento estén compradas, no se podrá llevar a cabo la edición de dicha entrada.

Dentro de la gestión de entradas, una alternativa interesante es la puesta en venta de las mismas. Esta funcionalidad permite al administrador disponer del tiempo necesario para poder editar las entradas de un evento, antes de que los clientes puedan comprarlas. Una vez modificados los campos oportunos de cada entrada, generalmente el precio, el usuario administrador accionará el botón de color naranja denominado “Puesta en Venta”, situado en la pantalla del listado de entradas, posteriormente seleccionará el evento cuyas entradas desea poner en venta y guardará los cambios. A partir de este momento, el cliente tendrá la posibilidad de adquirir las entradas que estime conveniente para ese evento, llevándose a cabo la realización del proceso completo de compra y pago de las mismas.

Es importante comentar que el botón de puesta en venta estará desactivado cuando el sistema no disponga de eventos con entradas que no están puestas a la venta.

La siguiente ilustración muestra el formulario existente para poner en venta las entradas de un evento en concreto:

The screenshot shows the 'Puesta en Venta de Entradas' form. At the top right, there is a dark header with 'Administrador del Sistema' and a green 'Salir' button. On the left, a blue sidebar contains a 'Menú' section with links: 'Gestión de Usuarios', 'Gestión de Deportes', 'Gestión de Competiciones', 'Gestión de Participantes', 'Gestión de Eventos', 'Gestión de Entradas', and 'Gestión de Compras'. The main content area has the title 'Puesta en Venta de Entradas' and the instruction 'Seleccione el evento cuyas entradas desea poner en venta'. Below this is a dropdown menu labeled 'Evento:' with the text '-- Seleccione evento --'. At the bottom of the form are three buttons: 'Guardar' (green), 'Reestablecer' (blue), and 'Volver' (dark blue).

Ilustración 143: Puesta en venta

▪ Gestión de compras

La gestión de compras llevada a cabo por el usuario administrador es muy simple, ya que solamente dispone del listado de las mismas y una opción habilitada para visualizar toda la información acerca de las compras realizadas por los clientes de la aplicación.

A continuación se muestran ejemplos de pantallas concernientes a la funcionalidad citada anteriormente:

The screenshot shows the 'Listado de Compras' table. At the top right, there is a dark header with 'Administrador del Sistema' and a green 'Salir' button. On the left, a blue sidebar contains a 'Menú' section with links: 'Gestión de Usuarios', 'Gestión de Deportes', 'Gestión de Competiciones', 'Gestión de Participantes', 'Gestión de Eventos', 'Gestión de Entradas', and 'Gestión de Compras'. The main content area has the title 'Listado de Compras' and a table with the following data:

Id	Usuario	Fecha De Compra	Fecha De Pago
1	Ramón Bailén Sánchez	2015-06-09 12:07:08.596	2015-06-09 12:09:10.291

Below the table is a button labeled 'Visualiza' with a magnifying glass icon.

Ilustración 144: Listado de compras

Administrador del Sistema » Salir

Menú

Visualizar Compra

Id Compra: 1
Usuario: Ramón Bailén Sánchez
Fecha de Compra: 2015-06-09 12:07:08.596
Fecha de Pago: 2015-06-09 12:09:10.291

Entradas	
Descripción	Precio
Entrada 3 para evento: Real Madrid Vs FC Barcelona	45.0 €

Total: 45.0 €

Volver

[Gestión de Usuarios](#)
[Gestión de Deportes](#)
[Gestión de Competiciones](#)
[Gestión de Participantes](#)
[Gestión de Eventos](#)
[Gestión de Entradas](#)
[Gestión de Compras](#)

Ilustración 145: Visualizar compra

- **Rol cliente**

Un usuario cliente es aquel que tiene la posibilidad de buscar los eventos deportivos definidos en el sistema por el usuario administrador, con el objetivo de ejercer la opción de compra de las entradas pertenecientes a dichos eventos y posteriormente realizar el pago a través de PayPal. Para iniciar sesión, introducirá sus credenciales en el formulario de login, al igual que el usuario administrador.

La ilustración siguiente muestra la pantalla principal de un usuario con este rol:

Ramón Bailén Sánchez » Salir

Menú

Bienvenido/a a la aplicación de venta de localidades deportivas



Compre entradas para alguno de nuestros deportes

[Datos Personales](#)
[Ver Eventos](#)
[Ver Compras Realizadas](#)

Ilustración 146: Menú cliente

El usuario cliente podrá finalizar la sesión iniciada en cualquier momento pulsando en el botón “Salir” situado en la esquina superior derecha. A continuación se detallan cada una de las opciones disponibles en la pantalla anterior:

▪ Datos personales

El usuario cliente puede administrar sus datos personales, es decir, tiene la posibilidad de modificar la información referente a su perfil o bien eliminarlo, en cuyo caso será conducido a la pantalla de inicio de la aplicación.

A continuación se muestra una imagen en la que se visualizan los datos personales de un cliente:

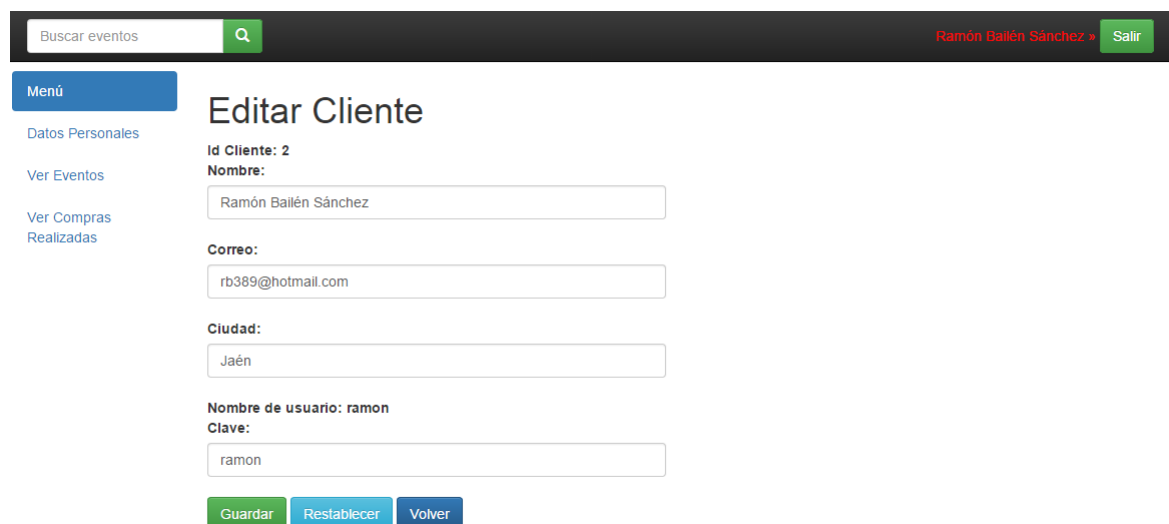


Id Cliente	Ciudad
2	Jaén
Nombre	Usuario
Ramón Bailén Sánchez	ramon
Correo	Contraseña
rb389@hotmail.com	ramon

[Editar](#) [Eliminar](#) [Volver](#)

Ilustración 147: Datos personales

Seguidamente se adjunta una ilustración con el formulario de edición para dicho cliente:



Editar Cliente

Id Cliente: 2

Nombre:

Correo:

Ciudad:

Nombre de usuario: ramon

Clave:

[Guardar](#) [Restablecer](#) [Volver](#)

Ilustración 148: Editar cliente

Como puede apreciarse, todos los campos son modificables, excepto el identificador del cliente (generado automáticamente) y el nombre de usuario, que sólo podrá ser modificado por el usuario administrador.

▪ Ver eventos

Esta opción permite al cliente localizar los eventos a los que desea asistir. Una vez que el cliente ha seleccionado esta opción, el sistema muestra el listado de deportes que en ese momento disponen de entradas para comprar, tal y como muestra el ejemplo de la siguiente imagen:



Ilustración 149: Ver eventos

Después de haber seleccionado el deporte, el sistema presenta la siguiente pantalla:



Ilustración 150: Categorías deporte

Como puede observarse, cada deporte tiene tres categorías: la primera hace referencia a la denominación del deporte, es decir, al término utilizado para la denominación del participante de ese deporte (en este caso equipos), la segunda

son las competiciones de dicho deporte y la tercera alberga el catálogo de todos los eventos disponibles en ese momento para el deporte en cuestión.

El cliente puede acceder a la categoría que desee, según los conocimientos que tenga sobre el evento que está buscando. Cuando localice el evento, se mostrará una pantalla similar al ejemplo siguiente:

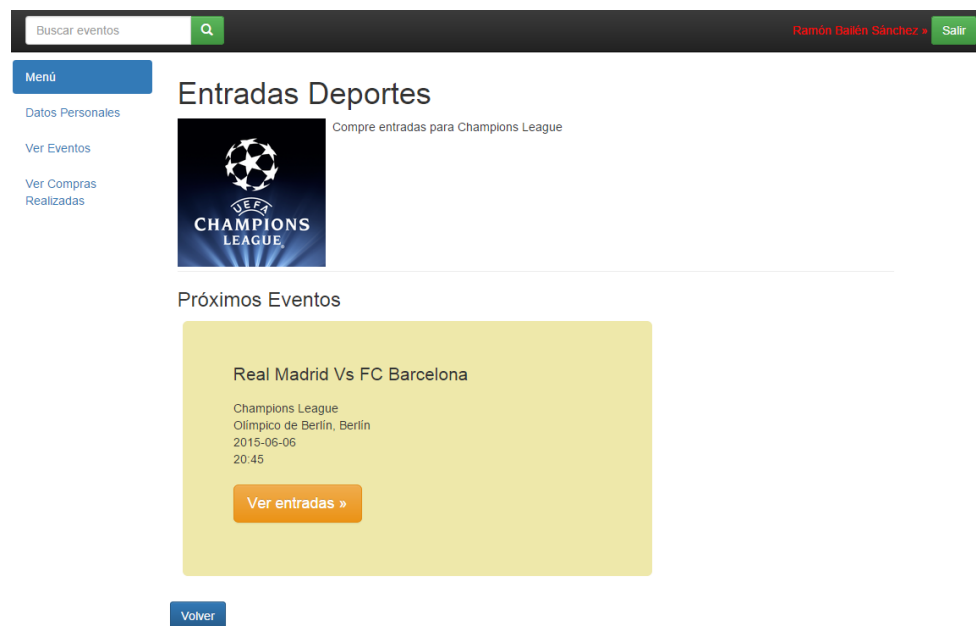


Ilustración 151: Evento localizado

El siguiente paso consiste en pinchar sobre el botón habilitado para ver las entradas y aparece la siguiente pantalla:



Ilustración 152: Entradas deportes

A continuación, el cliente deberá seleccionar las entradas que prefiera y pinchar sobre el botón de compra. Así, el sistema lo redirigirá a la pantalla de confirmación de compra, donde podrá cancelar la compra o pagarla.

Ramón Bailén Sánchez » Salir

Entradas Deportes

Confirmación de Compra

Compra nº
36

Cliente
Ramón Bailén Sánchez

Fecha de Compra
2015-06-06 19:02:29.225

Total
60.0 €

Tipo de Entrada

Descripción
Entrada 1 para evento: Real Madrid Vs FC Barcelona

Deporte
Fútbol

Precio
25.0 €

Descripción
Entrada 3 para evento: Real Madrid Vs FC Barcelona

Deporte
Fútbol

Precio
35.0 €

✕ Cancelar Compra PayPal PAGAR

Ilustración 153: Confirmación de compra

Si el cliente decide cancelar la compra, el sistema lo conducirá al listado de deportes. Por el contrario, si desea pagarla, el cliente será redirigido a la pasarela de pago, donde le aparecerá lo siguiente:

Resumen de su pedido

Descripciones	Importe
Entrada 1 para evento: Real Madrid...	€25,00
Precio del artículo: €25,00	
Cantidad: 1	
Entrada 3 para evento: Real Madrid...	€35,00
Precio del artículo: €35,00	
Cantidad: 1	
Importe total a pagar	€60,00
Total €60,00 EUR	

Seleccione una forma de pago

Pagar con mi cuenta PayPal

PayPal

Inicie sesión en su cuenta para pagar.

Correo electrónico

Contraseña de PayPal

☐ Este ordenador es privado. [¿Qué es esto?](#)

Entrar

[¿Ha olvidado su contraseña o correo electrónico?](#)

Crear una cuenta PayPal

Y pague con su tarjeta de débito o crédito.

Ilustración 154: Resumen PayPal

Para realizar el pago, el cliente tendrá que identificarse. Si ya dispone de una cuenta, introducirá sus credenciales. Si no, deberá de crear una cuenta PayPal.

Una vez que el cliente ha accedido a la plataforma, podrá pagar la compra realizada accionando el botón denominado “Pagar ahora”. La siguiente imagen presenta el botón pertinente:

Resumen de su pedido

Descripciones	Importe
Entrada 1 para evento: Real Madrid...	€25.00
Precio del artículo: €25.00	
Cantidad: 1	
Entrada 3 para evento: Real Madrid...	€35.00
Precio del artículo: €35.00	
Cantidad: 1	
Importe total a pagar	€60.00
Total €60.00 EUR	

Revise su información

Pagar ahora

Dirección de envío [Cambiar](#)

test buyer
1 Main St
San Jose, CA 95131
Estados Unidos

¿Desea hacerle un comentario al vendedor?: [Haga el comentario aquí](#)

Formas de pago [Cambiar](#)

Saldo PayPal \$81.76 USD

Tasa de conversión de PayPal a partir del 6 jun 2015: 1 Dólar estadounidense = 0.733914 Euros

■ Tarjeta de regalo PayPal, certificado de regalo, recompensa u otro descuento [Canjear](#)
Consulte las [Políticas de PayPal](#) y sus derechos sobre orígenes de pago.

Información de contacto
rb389-buyer@hotmail.com

Pagar ahora

Ilustración 155: Pagar PayPal

Tras abonar el importe de las entradas, PayPal redirigirá al cliente a nuestra aplicación, donde se confirmará el pago y se habilitará la posibilidad de imprimir las entradas adquiridas. Además, el cliente dispondrá de un listado de sugerencias deportivas, ofreciéndole de esta manera nuevos eventos para asistir.

Confirmación de Pago

Su compra ha sido pagada

Dispones de 4 días desde la fecha de pago para imprimir las entradas

Id Compra: 36

Usuario: Ramón Bailén Sánchez

Fecha de Compra: 2015-06-06 19:02:29.225

Fecha de Pago: 2015-06-06 19:26:30.59

Precio Total: 60.0 €

ENTRADAS

Descripción

Entrada 1 para evento: Real
Madrid Vs FC Barcelona

Deporte

Fútbol

Precio

25.0 €



Descripción

Entrada 3 para evento: Real
Madrid Vs FC Barcelona

Deporte

Fútbol

Precio

35.0 €



Imprimir entradas

Volver al inicio

Sugerencias



Compre entradas para Fútbol

Ilustración 156: Confirmación de pago

▪ Ver compras realizadas

Esta opción permite al cliente poder revisar su histórico de compras efectuadas a través de la aplicación.

A continuación se muestra el histórico de compras de un cliente determinado:

Id	Usuario	Fecha De Compra	Fecha De Pago	Precio Total		
33	Ramón Bailén Sánchez	2015-06-04 17:29:34.553	2015-06-04 17:41:53.91	7.0 €	Visualiza	Imprimir entradas
34	Ramón Bailén Sánchez	2015-06-04 19:08:33.886	2015-06-04 19:09:03.402	1.0 €	Visualiza	Imprimir entradas
35	Ramón Bailén Sánchez	2015-06-05 13:37:23.535	2015-06-05 13:37:39.241	2.0 €	Visualiza	Imprimir entradas

Ilustración 157: Histórico compras

Como puede observarse, el cliente también tiene la posibilidad de contemplar más información acerca de cada una de las compras seleccionando el botón de visualizar situado al lado de cada una de éstas.

Seguidamente se muestra un ejemplo:

Id Compra: 33
Usuario: Ramón Bailén Sánchez
Fecha de Compra: 2015-06-04 17:29:34.553
Fecha de Pago: 2015-06-04 17:41:53.91

Entradas	
Descripción	Precio
Entrada 3 para evento: Prueba	3.0 €
Entrada 4 para evento: Prueba	4.0 €

Total: 7.0 €

Volver

Ilustración 158: Visualizar compra

Además, si el cliente no ha imprimido las entradas después de pagar la totalidad de la compra, puede hacerlo también a través de la opción del menú “Ver compras realizadas”, y a continuación “Imprimir entradas” de la compra en cuestión. Una vez que el cliente haya abonado el precio de la compra, dispone de cuatro días para imprimir las entradas.

En la ilustración siguiente se presenta la funcionalidad de impresión para las entradas de una compra determinada:

Ramón Bailén Sánchez » Salir

Confirmación de Pago

Su compra ha sido pagada

Dispone de 4 días desde la fecha de pago para imprimir las entradas

Id Compra: 35
Usuario: Ramón Bailén Sánchez
Fecha de Compra: 2015-06-05 13:37:23.535
Fecha de Pago: 2015-06-05 13:37:39.241
Precio Total: 2.0 €

ENTRADAS

Descripción
Entrada 2 para evento: Prueba

Deporte
Fútbol

Precio
2.0 €



 Imprimir entradas  Volver

Ilustración 159: Imprimir

▪ Buscar eventos

Cualquier usuario cliente que este navegando por la aplicación web podrá realizar una búsqueda de eventos. Para realizar esta acción deberá utilizar el buscador situado en la parte superior izquierda de la aplicación e introducir un criterio de búsqueda para localizar los eventos. Los criterios de búsqueda admitidos por el sistema son: ciudad, deporte, competición y participante.

A continuación se muestra un ejemplo de búsqueda de un evento, a través del nombre de la ciudad:



Ilustración 160: Búsqueda

Una vez encontrado el evento deseado, el usuario cliente deberá pulsar sobre el botón habilitado para ver las entradas de dicho acontecimiento y seguirá el mismo proceso descrito para la opción de ver eventos en el menú.

Apéndice III. Descripción de contenidos suministrados

En este apéndice se muestra la organización del material proporcionado en el CD que se adjunta en la documentación de este proyecto:

- /Memoria: esta carpeta contiene este documento en formato PDF: **TFG-Ramón_Bailén_Sánchez.pdf**. Abarca la siguiente subcarpeta:
 - /Material adicional: contiene el fichero **Proyecto.vpp**, donde se incluyen los diagramas de caso de uso, el modelo de dominio, los diagramas de secuencia, los diagramas de clases divididos por paquetes, el modelo Entidad-Relación y la estructura del plan de trabajo. Además, también engloba el fichero correspondiente a la guía de integración de Pasarela integral de PayPal, cuyo nombre es **pasarelaintegral_es.pdf** y la especificación de los casos de uso en un nuevo fichero denominado **Casos de Uso.xlsx**.
- /Aplicación web: esta carpeta comprende todo lo necesario para el funcionamiento de la aplicación web. En la raíz contiene el archivo **Localidades.war**. Además, engloba las subcarpetas siguientes:
 - /Código fuente: incluye la carpeta /Localidades que contiene todos los archivos, librerías y código fuente necesarios para el correcto funcionamiento de la aplicación web.
 - /Herramientas: contiene el fichero **países.sql**, que permite cargar el listado de países en la base de datos.
 - /Base de datos: comprende la carpeta /tfg que contiene la base de datos utilizada en la aplicación.