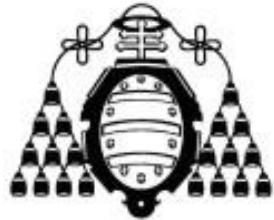


	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 1 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	---

UNIVERSIDAD DE OVIEDO



ESCUELA DE INGENIERÍA INFORMÁTICA

TRABAJO FIN DE GRADO

“SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA EN EL BALONCESTO
PROFESIONAL”

DIRECTOR: Hugo Lebredo Bujan

AUTOR: Alejandro Galán Freire

Enlace GitHub

<https://github.com/AlejandroGalanFreire/TFG>

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 2 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	---

CONTROL DE VERSIONES

Versión	Fecha	Descripción
1.0	04/07/2023	Primera versión del proyecto

Resumen

NbaAnalytics tiene como objetivo desarrollar un sistema de reporting sobre estadística avanzada de baloncesto profesional proporcionando al usuario un servicio en el que de manera unificada pueda acceder a la información de diferentes competiciones. Para ello se ha desarrollado un sistema capaz de consumir información de fuentes de datos independientes entre sí, procesarlas y unificarlas.

En cuanto a la visualización de los datos se ha desarrollado un sitio web que permita analizar la información de cualquier partido de una determinada competición, estadísticas de rendimiento de jugadores y de equipos en diferentes períodos de tiempo, pudiendo incluso hacer comparaciones entre ellos.

La consecución de este sistema ha requerido del análisis y utilización de diferentes tecnologías y armonizarlas en las diferentes etapas del sistema. Desde la recolección y consumo de información con su correspondiente normalización, hasta la realización de gráficos. Todos los frameworks y librerías están integrados creando un sistema robusto que tiene como objetivos principales la escalabilidad y la usabilidad.

Palabras Clave

Análisis estadístico deportivo, Visualización de datos, Normalización de la información, Tecnologías Web, Modelización de datos, Baloncesto, Ligas Profesionales.

Abstract

NbaAnalytics aims to develop a reporting system on advanced professional basketball statistics necessary for the user, a service in which they can access information from different competitions in a unified manner. For this, a system capable of consuming information from data sources independent of each other, processing and unifying them has been developed.

Regarding the visualization of the data, a website has been developed that allows analyzing the information of any match of a certain competition, performance statistics of players and teams in different periods of time, and can even make comparisons between them.

In order to achieve this system, it has required the analysis and use of different technologies and harmonizing them in the different stages of the system. From the collection and consumption of information with its corresponding processing, to the creation of graphics, all the frameworks and libraries are integrated creating a robust system whose main objectives are scalability and usability.

Keywords

Sports statistical analysis, Data visualization, Information normalization, Web Technologies, Data modeling, Basketball, Professional Leagues.

Índice General

CAPÍTULO 1. INTRODUCCIÓN	17
1.1 JUSTIFICACIÓN DEL PROYECTO	17
1.2 OBJETIVOS DEL PROYECTO	17
1.3 ESTUDIO DE LA SITUACIÓN ACTUAL	18
1.3.1 <i>Evaluación de Alternativas</i>	18
CAPÍTULO 2. ASPECTOS TEÓRICOS	27
2.1 HTML.....	27
2.2 CSS.....	27
2.3 TYPESCRIPT.....	28
2.4 BASE DE DATOS RELACIONAL	28
2.5 ANGULAR.....	29
2.6 CHART.JS.....	30
2.7 BOOTSTRAP	31
2.8 SPRINGBOOT	31
2.9 PYTHON	32
2.10 MYSQL	33
2.11 MYBATIS.....	34
2.12 PATRÓN MVC	35
2.13 VISUAL STUDIO CODE	36
2.14 SPRING TOOL SUITE.....	36
2.15 MySQL WORKBENCH	37
2.16 ANACONDA – SPYDER.....	37
CAPÍTULO 3. PLANIFICACIÓN Y GESTIÓN DEL PROYECTO.....	38
3.1 PLANIFICACIÓN INICIAL.....	39
3.2 PLANIFICACIÓN FINAL	39
3.3 ESTRUCTURA DE DESGLOSE DE TRABAJO.....	40
3.3.1 <i>Tutorías</i>	40
3.3.2 <i>Investigación</i>	40
3.3.3 <i>Planificación y gestión del proyecto</i>	41
3.3.4 <i>Análisis</i>	41
3.3.5 <i>Diseño</i>	42
3.3.6 <i>Implementación</i>	42
3.3.7 <i>Desarrollo de pruebas</i>	43
3.3.8 <i>Documentación</i>	43
3.4 RIESGOS.....	43
3.4.1 <i>Identificación de los riesgos</i>	43
3.4.2 <i>Evaluación del impacto</i>	44
3.4.3 <i>Planificación de la gestión de cada riesgo</i>	45
3.5 RESUMEN DEL PRESUPUESTO	46
3.5.1 <i>Resumen presupuesto de costes</i>	46
3.5.2 <i>Resumen presupuesto de cliente</i>	47
CAPÍTULO 4. ANÁLISIS	48

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 8 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	---

4.1	DEFINICIÓN DEL SISTEMA.....	48
4.1.1	<i>Diagrama de 'contexto' del sistema</i>	48
4.1.2	<i>Funcionalidades del sistema</i>	48
4.1.3	<i>Estado del arte</i>	49
4.1.4	<i>Determinación del Alcance del Sistema</i>	52
4.2	REQUISITOS DEL SISTEMA	53
4.2.1	<i>Obtención de los Requisitos del Sistema</i>	53
4.2.2	<i>Identificación de Actores del Sistema</i>	55
4.2.3	<i>Especificación de Casos de Uso</i>	56
4.3	IDENTIFICACIÓN DE LOS SUBSISTEMAS EN LA FASE DE ANÁLISIS	58
4.3.1	<i>Descripción de los Subsistemas</i>	59
4.3.2	<i>Descripción de los Interfaces entre Subsistemas</i>	59
4.4	DIAGRAMA DE CLASES PRELIMINAR DEL ANÁLISIS.....	60
4.4.1	<i>Diagrama de Clases</i>	61
4.4.2	<i>Descripción de las Clases</i>	61
4.5	ANÁLISIS DE CASOS DE USO Y ESCENARIOS.....	73
4.5.1	<i>Visualizar Inicio</i>	73
4.5.2	<i>Ver Clasificación</i>	74
4.5.3	<i>Ver Jugadores</i>	74
4.5.4	<i>Filtrar jugadores</i>	75
4.5.5	<i>Ver detalle de jugador</i>	75
4.5.6	<i>Comparar jugadores</i>	76
4.5.7	<i>Ver equipos</i>	77
4.5.8	<i>Filtrar equipos</i>	77
4.5.9	<i>Ver detalle de equipo</i>	78
4.5.10	<i>Comparar equipos</i>	78
4.5.11	<i>Ver plantilla de equipo</i>	79
4.5.12	<i>Ver partidos</i>	79
4.5.13	<i>Ver detalle de partidos</i>	80
4.6	RELACIÓN ESCENARIOS – CASOS DE USO – REQUISITOS.....	81
4.7	ANÁLISIS DE INTERFACES DE USUARIO	82
4.7.1	<i>Descripción de la Interfaz</i>	82
4.7.2	<i>Descripción del Comportamiento de la Interfaz</i>	89
4.7.3	<i>Diagrama de Navegabilidad</i>	90
4.8	ESPECIFICACIÓN DEL PLAN DE PRUEBAS.....	91
4.8.1	<i>Pruebas unitarias y de calidad</i>	91
4.8.2	<i>Pruebas de integración y de sistema</i>	93
4.8.3	<i>Pruebas de usabilidad</i>	96
CAPÍTULO 5. DISEÑO DEL SISTEMA.....		97
5.1	ARQUITECTURA DEL SISTEMA	97
5.1.1	<i>Diagramas de Paquetes</i>	97
5.1.2	<i>Diagrama de Despliegue</i>	100
5.2	DISEÑO DE CLASES	102
5.2.1	<i>Diagrama de Clases</i>	102
5.3	DIAGRAMAS DE INTERACCIÓN.....	109
5.3.1	<i>Visualizar partido</i>	109
5.3.2	<i>Visualizar jugador específico tras búsqueda</i>	111
5.4	DISEÑO DE LA BASE DE DATOS.....	112

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 9 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	---

5.4.1	<i>Descripción del SGBD Usado</i>	112
5.4.2	<i>Integración del SGBD en Nuestro Sistema</i>	112
5.4.3	<i>Diagrama E-R</i>	113
5.5	DISEÑO DE LA INTERFAZ	114
5.6	ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS.....	122
5.6.1	<i>Pruebas Unitarias y de Calidad</i>	122
5.6.2	<i>Pruebas de Integración y del Sistema</i>	123
5.6.3	<i>Pruebas de Usabilidad</i>	123
CAPÍTULO 6. IMPLEMENTACIÓN DEL SISTEMA		126
6.1	ESTÁNDARES Y NORMAS SEGUIDOS	126
6.2	LENGUAJES DE PROGRAMACIÓN.....	126
6.2.1	<i>Angular</i>	126
6.2.2	<i>SpringBoot</i>	127
6.2.3	<i>MySQL</i>	127
6.2.4	<i>Python</i>	128
6.3	HERRAMIENTAS Y PROGRAMAS USADOS PARA EL DESARROLLO DEL SISTEMA.....	128
6.3.1	<i>Visual Studio Code</i>	128
6.3.2	<i>Eclipse – Spring Tool Suite</i>	128
6.3.3	<i>MySQL Workbench</i>	128
6.3.4	<i>Spyder (Anaconda 3)</i>	129
6.3.5	<i>Microsoft Excel</i>	129
6.3.6	<i>SonarQube</i>	129
6.4	HERRAMIENTAS Y PROGRAMAS USADOS PARA EL DESARROLLO DE LA DOCUMENTACIÓN	129
6.4.1	<i>Microsoft Word</i>	129
6.4.2	<i>Microsoft Project</i>	129
6.4.3	<i>UMLetino</i>	130
6.5	OTRAS HERRAMIENTAS.....	130
6.5.1	<i>GitHub</i>	130
6.6	CREACIÓN DEL SISTEMA	130
6.6.1	<i>Documentación sistemas</i>	130
6.6.2	<i>Problemas Encontrados</i>	131
CAPÍTULO 7. DESARROLLO DE LAS PRUEBAS		133
7.1	PRUEBAS UNITARIAS Y DE CALIDAD.....	133
7.2	PRUEBAS DE INTEGRACIÓN Y DEL SISTEMA	134
7.3	PRUEBAS DE USABILIDAD.....	138
7.3.1	<i>Sesión 1</i>	138
7.3.2	<i>Sesión 2</i>	140
7.3.3	<i>Sesión 3</i>	142
7.3.4	<i>Conclusiones</i>	144
CAPÍTULO 8. 8 MANUALES DE SISTEMA.....		144
8.1	MANUAL DE EJECUCIÓN	144
8.2	MANUAL DE USUARIO	149
8.3	MANUAL DEL PROGRAMADOR.....	160
8.3.1	<i>Backend</i>	160
8.3.2	<i>Frontend</i>	164
8.3.3	<i>Base de Datos</i>	168

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 10 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	--

8.3.4 <i>Extracción de datos</i>	169
CAPÍTULO 9. CONCLUSIONES Y AMPLIACIONES.....	169
9.1 CONCLUSIONES.....	169
9.2 AMPLIACIONES.....	170
CAPÍTULO 10. REFERENCIAS BIBLIOGRÁFICAS.....	171
10.1 REFERENCIAS LIBROS Y PLANTILLAS	171
10.2 REFERENCIAS INTERNET.....	171
CAPÍTULO 11. APÉNDICES	175
11.1 DETALLE DEL PRESUPUESTO.....	175
11.1.1 <i>Definición de la empresa</i>	175
11.1.2 <i>Presupuesto de costes</i>	177
11.1.3 <i>Presupuesto de cliente</i>	181
11.2 GLOSARIO Y DICCIONARIO DE DATOS	182
11.2.1 <i>Glosario</i>	182
11.2.2 <i>Diccionario de datos</i>	184
11.3 CONTENIDO ENTREGADO EN EL ARCHIVO ADJUNTO	186
11.3.1 <i>Contenidos</i>	186
11.4 CÓDIGO FUENTE	188
11.4.1 <i>Procesamiento previo</i>	188
11.4.2 <i>Base de Datos</i>	190
11.4.3 <i>Backend</i>	192
11.4.4 <i>Frontend</i>	197

Índice de Figuras

Ilustración 1-1 Icono Basketball Reference	19
Ilustración 1-2 Captura Basketball Reference.....	19
Ilustración 1-3 Icono de sitio de la NBA.....	20
Ilustración 1-4 Captura sitio de la NBA.....	20
Ilustración 1-5 Icono ESPN Deportes	21
Ilustración 1-6 Captura ESPN Deportes	22
Ilustración 1-7 Icono Marca	23
Ilustración 1-8 Captura Marca.....	23
Ilustración 1-9 Icono As.....	24
Ilustración 1-10 Captura Diario As.....	24
Ilustración 3-1 Planificación Inicial	39
Ilustración 3-2 Planificación Final.....	39
Ilustración 3-3 Desglose tutorías	40
Ilustración 3-4 Desglose Investigación	41
Ilustración 3-5 Desglose Planificación	41
Ilustración 3-6 Desglose Análisis	41
Ilustración 3-7 Desglose Diseño	42
Ilustración 3-8 Desglose Implementación	42
Ilustración 3-9 Desglose Desarrollo de pruebas	43
Ilustración 3-10 Desglose documentación.....	43
Ilustración 4-1 Diagrama de 'contexto' del sistema.....	48
Ilustración 4-2 Diagrama de casos de uso	56
Ilustración 4-3 Diagrama de clases global	61
Ilustración 4-4 Prototipo pantalla de inicio	83
Ilustración 4-5 Prototipo pantalla clasificación	83
Ilustración 4-6 Prototipo pantalla de jugadores	84
Ilustración 4-7 Prototipo pantalla de equipos	85
Ilustración 4-8 Prototipo pantalla partidos	85
Ilustración 4-9 Prototipo pantalla de plantilla de equipo	86
Ilustración 4-10 Prototipo pantalla de detalle de jugador	87
Ilustración 4-11 Prototipo pantalla de detalle de equipo	88
Ilustración 4-12 Prototipo pantalla de detalle de partidos	88
Ilustración 4-13 Prototipo pantalla de comparación de jugadores y equipos	89
Ilustración 4-14 Diagrama de interactividad	90
Ilustración 5-1 Diagrama de paquetes frontend	97
Ilustración 5-2 Diagrama de paquetes backend	98
Ilustración 5-3 Diagrama de Despliegue.....	100
Ilustración 5-4 Diagrama de clases Clasificación.....	102
Ilustración 5-5 Diagrama de clases Player	103
Ilustración 5-6 Diagrama de clases TeamByYear	104
Ilustración 5-7 Diagrama de clases GameElem	105
Ilustración 5-8 Diagrama de clases League	105
Ilustración 5-9 Diagrama de clases front	106
Ilustración 5-10 Diagrama de clases Raíz.....	106

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 12 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	--

Ilustración 5-11 Diagrama de clases módulo Players.....	107
Ilustración 5-12 Diagrama de clases módulo Teams.....	108
Ilustración 5-13 Diagrama de clases módulo Matchs	109
Ilustración 5-14 Diagrama de secuencia Visualizar Partido	110
Ilustración 5-15 Diagrama de secuencia Visualizar Jugador buscado.....	111
Ilustración 5-16 Diagrama Entidad Relación.....	113
Ilustración 5-17 Pantalla Inicio	115
Ilustración 5-18 Ventana Clasificación.....	116
Ilustración 5-19 Pantalla Jugadores.....	116
Ilustración 5-20 Pantalla Equipos	117
Ilustración 5-21 Pantalla Partidos	118
Ilustración 5-22 Pantalla Detalle Jugador	118
Ilustración 5-23 Pantalla Detalle Equipo	119
Ilustración 5-24 Pantalla Detalle Partido	119
Ilustración 5-25 Pantalla Plantilla Equipo.....	120
Ilustración 5-26 Pantalla Comparación Jugadores	121
Ilustración 5-27 Pantalla Comparación Equipos	122
Ilustración 8-1 Lanzamiento Base de Datos.....	145
Ilustración 8-2 Base de Datos en ejecución	146
Ilustración 8-3 Ejecución del backend	147
Ilustración 8-4 Consola del backend tras ejecución	147
Ilustración 8-5 Ejecución del frontend.....	148
Ilustración 8-6 Acceso a la app a través del puerto del frontend	149
Ilustración 8-7 Pantalla de inicio	150
Ilustración 8-8 Menú de navegación	150
Ilustración 8-9 Pantalla Clasificación	151
Ilustración 8-10 Selector de liga	151
Ilustración 8-11 Pantalla listado de jugadores.....	152
Ilustración 8-12 Jugador filtrado por búsqueda.....	152
Ilustración 8-13 Pantalla de detalle de jugador	153
Ilustración 8-14 Botón de comparación de jugadores	153
Ilustración 8-15 Listado tras búsqueda.....	153
Ilustración 8-16 Pantalla comparación jugadores.....	154
Ilustración 8-17 Equipos filtrados tras búsqueda	154
Ilustración 8-18 Pantalla plantilla de equipo	155
Ilustración 8-19 Pantalla de detalle de equipo	155
Ilustración 8-20 Búsqueda de equipo para comparar	156
Ilustración 8-21 Pantalla de comparación de equipos	157
Ilustración 8-22 Pantalla de visualización de partidos	158
Ilustración 8-23 Calendario para seleccionar fecha de partidos	158
Ilustración 8-24 Búsqueda sin coincidencias	159
Ilustración 8-25 Fecha sin partidos.....	159
Ilustración 8-26 Dependencia web	160
Ilustración 8-27 Dependencia Spring.....	160
Ilustración 8-28 Dependencia MySQL	161
Ilustración 8-29 Dependencia Jacoco	161
Ilustración 8-30 Dependencia MyBatis.....	161
Ilustración 8-31 Configuración myBatis.....	162

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 13 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	--

Ilustración 8-32 Archivo XML de un Mapper	162
Ilustración 8-33 Fichero application.properties.....	163
Ilustración 8-34 Configuración Sonarqueba del backend.....	163
Ilustración 8-35 Configuración Sonarqube del backend (2)	164
Ilustración 8-36 Dependencias de proyecto	165
Ilustración 8-37 Dependencias de desarrollo	165
Ilustración 8-38 Comandos ejecutables (scripts)	166
Ilustración 8-39 Configuración Karma	167
Ilustración 8-40 Opción de conexión a base de datos.....	168
Ilustración 8-41 Conexión a base de datos	168
Ilustración 8-42 Ejemplo importaciones fichero extracción de datos	169
Ilustración 11-1 Enpoint estadísticas generales jugadores	188
Ilustración 11-2 Procesamiento datos generales de los jugadores	189
Ilustración 11-3 Fichero CSV con datos generales de los jugadores	190
Ilustración 11-4 Entidad Player	191
Ilustración 11-5 Importación datos a base de datos	192
Ilustración 11-6 Modelo Jugador.....	192
Ilustración 11-7 Controlador jugadores.....	193
Ilustración 11-8 Servicio jugadores (interfaz)	193
Ilustración 11-9 Servicio jugadores (implementación)	194
Ilustración 11-10 Mapeador jugadores (interfaz)	194
Ilustración 11-11 Mapeador jugadores (fichero XML)	195
Ilustración 11-12 Pruebas modelo.....	195
Ilustración 11-13 Pruebas controlador	196
Ilustración 11-14 Pruebas servicio.....	197
Ilustración 11-15 Módulo de jugadores.....	198
Ilustración 11-16 Módulo de rutas de los jugadores.....	198
Ilustración 11-17 Modelo Jugador.....	199
Ilustración 11-18 Componente jugadores	200
Ilustración 11-19 Servicio para peticiones	201
Ilustración 11-20 Fichero HTML Jugadores	201
Ilustración 11-21 Fichero CSS Jugadores	202
Ilustración 11-22 Clase de pruebas Jugadores	203

Índice de Tablas

Tabla 1 Comparativa sitios web de baloncesto estudiados	25
Tabla 2 Identificación de riesgos	44
Tabla 3 Matriz de probabilidad e impacto.....	44
Tabla 4 Probabilidad e impacto de cada riesgo	45
Tabla 5 Resumen presupuesto de costes	46
Tabla 6 Resumen presupuesto cliente	47
Tabla 7 Requisitos funcionales.....	54
Tabla 8 Requisitos no funcionales	55
Tabla 9 Caso de uso Ver Inicio.....	56
Tabla 10 Caso de uso Ver Clasificación	56
Tabla 11 Caso de uso Ver Jugadores.....	57
Tabla 12 Caso de uso Detalle de jugador.....	57
Tabla 13 Caso de uso Comparar Jugadores	57
Tabla 14 Caso de uso Ver Equipos	57
Tabla 15 Caso de uso Plantilla de equipo	57
Tabla 16 Caso de uso Detalle de Equipo	58
Tabla 17 Caso de uso Comparar Equipos.....	58
Tabla 18 Caso de uso Ver Partido	58
Tabla 19 Caso de uso Detalle Partido	58
Tabla 20 Clase Player	62
Tabla 21 Clase Players.....	62
Tabla 22 Clase PlayerCard	62
Tabla 23 Clase PlayerDetail	63
Tabla 24 Clase PlayerComparative	63
Tabla 25 Clase Team	63
Tabla 26 Clase Teams	64
Tabla 27 Clase TeamCard	64
Tabla 28 Clase TeamDetail	64
Tabla 29 Clase TeamTemplate	65
Tabla 30 Clase TeamComparative	65
Tabla 31 Clase Clasification	65
Tabla 32 Clase Match	66
Tabla 33 Clase Matchs	66
Tabla 34 Clase MatchDetail.....	66
Tabla 35 Clase PlayerModel	67
Tabla 36 Clase PlayerController	68
Tabla 37 Clase PlayerService	68
Tabla 38 Clase PlayerMapper	69
Tabla 39 Clase TeamController	69
Tabla 40 Clase ClasificationController	70
Tabla 41 Clase MatchController	70
Tabla 42 Clase LeagueController	70
Tabla 43 Entidad Clasification	71
Tabla 44 Entidad League	71

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 15 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	--

Tabla 45 Entidad Player	72
Tabla 46 Entidad Team.....	72
Tabla 47 Entidad Match	73
Tabla 48 Escenario Visualizar Inicio.....	73
Tabla 49 Escenario Visualizar Clasificacion.....	74
Tabla 50 Escenario Visualizar Jugadores	74
Tabla 51 Escenario Filtrar Jugadores	75
Tabla 52 Escenario Detalle de Jugador	75
Tabla 53 Escenario Comparar Jugadores	76
Tabla 54 Escenario Visualizar Equipos.....	77
Tabla 55 Escenario Filtrar Equipos	77
Tabla 56 Escenario Detalle Equipo	78
Tabla 57 Escenario Comparar Equipos	79
Tabla 58 Escenario Plantilla Equipo.....	79
Tabla 59 Escenario Ver Partidos.....	80
Tabla 60 Escenario Detalle Partido.....	80
Tabla 61 Relación Casos de uso / Requisitos	81
Tabla 62 Relación Casos de uso / Escenarios	82
Tabla 63 Pruebas módulo Clasificación	91
Tabla 64 Pruebas módulo Jugadores	92
Tabla 65 Pruebas módulo Equipos	92
Tabla 66 Pruebas módulo Partidos.....	93
Tabla 67 Pruebas Caso de Uso Ver Clasificación	93
Tabla 68 Pruebas Casos de Uso Jugadores	94
Tabla 69 Pruebas Casos de Uso Equipos.....	95
Tabla 70 Pruebas Casos de Uso Partidos	96
Tabla 71 Resultado Pruebas Clasificación.....	134
Tabla 72 Resultado Pruebas Jugadores	136
Tabla 73 Resultado Pruebas Equipos.....	137
Tabla 74 Resultado Pruebas Partidos	138
Tabla 75 Personal de la empresa.....	175
Tabla 76 Productividad y costes.....	175
Tabla 77 Otros costes.....	176
Tabla 78 Facturación	176
Tabla 79 Resumen modelo de la empresa.....	176
Tabla 80 Partida Tutorías	177
Tabla 81 Partida Investigación	177
Tabla 82 Partida Planificación y gestión del proyecto	177
Tabla 83 Partida Análisis	178
Tabla 84 Partida Diseño	178
Tabla 85 Partida Implementación	179
Tabla 86 Partida Desarrollo de pruebas	180
Tabla 87 Partida Documentación	180
Tabla 88 Partida Otros costes	180
Tabla 89 Hardware y Software.....	181
Tabla 90 Presupuesto de costes	181
Tabla 91 Ponderaciones	182
Tabla 92 Presupuesto cliente	182

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 16 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	--

Capítulo 1. Introducción

1.1 Justificación del Proyecto

Este proyecto es realizado para permitir que los usuarios puedan disfrutar de una aplicación que muestre los datos de interés sobre el baloncesto profesional de una forma más atractiva, visual y cómoda de lo que se puede encontrar en internet.

En la versión actual del proyecto las funcionalidades aportadas son aquellas que se consideran como básicas, pero esenciales, sobre jugadores, equipos y partidos pero que sería fácilmente escalable para versiones futuras y que la aplicación fuera más profesional.

En el apartado siguiente [2.2 Objetivos del Proyecto](#) se describen las necesidades que el sistema cubre.

1.2 Objetivos del Proyecto

Los objetivos que pretende alcanzar este proyecto son:

1. Creación de un sitio web que permita al usuario la utilización de un sistema de reporting sobre estadísticas del baloncesto profesional:
 - a. Albergar información de diferentes competiciones centrada en:
 - i. Análisis de partidos.
 - ii. Análisis actuaciones individuales de los jugadores.
 - iii. Análisis de actuaciones de equipos en los últimos 5 años.
 - iv. Comparación de estadísticas de jugadores y equipos.
 - v. Visualización de la clasificación de una temporada.
 - b. Usabilidad, logrando que el sistema sea fácil de comprender y utilizar, resultando también atractivo visualmente al usuario.
2. Consumir información de diferentes fuentes, unificando y normalizando los datos.
3. Fácilmente escalable, permitiendo añadir nueva información.
4. Fácilmente mantenable.

1.3 Estudio de la Situación Actual

Son muchos los usuarios que consumen aplicaciones de estadísticas en su tiempo libre, ya sea por simple afición o por estudio de dichos datos con diversas intenciones, como puede ser el análisis de un jugador por parte de un profesional para un posterior fichaje, por ejemplo.

Teniendo en cuenta estos tipos de usuario, podemos garantizar que a todos ellos les gustará que la aplicación que visiten le muestre los datos de la forma más accesible y comprensible que se pueda.

Existen diversas alternativas que tratan de conseguir este objetivo de forma que el usuario no se frustre y esté satisfecho con los servicios que el sistema aporta, pero vamos a continuación en el apartado 2.3.1 Evaluación de Alternativas a hacer un análisis un poco más profundo sobre ciertos sitios, para orientar nuestro objetivo en el proyecto a realizar.

1.3.1 Evaluación de Alternativas

En esta sección se describirán, una por una, todas las alternativas estudiadas. Conviene estudiar 3 o 4 alternativas importantes, salvo que por algún motivo justificado se deba incluir un número menor o mayor de las mismas. En todo caso, siempre es conveniente cuidar de que en esta sección haya un conjunto de sistemas significativo. En función de lo dicho anteriormente, cada sistema podrá dividirse en tres secciones: “Descripción”, “Ventajas” e “Inconvenientes”, aunque es posible cualquier otra división que contenga los aspectos descritos, dependiendo de qué tipo de sistemas se estudien.

Teniendo en cuenta los potenciales usuarios que utilizarán nuestra aplicación podemos identificar una serie de sistemas ya existentes que pueden ser comparables con lo que queremos conseguir, y que nos permiten tener una base de la que partir con el objetivo de mejorarlo.

Como ya se comentó anteriormente uno de los objetivos principales es lograr la mayor usabilidad posible, y esto nos lleva a analizar las siguientes alternativas:

1.3.1.1 *Basketball References*

1.3.1.1.1 Descripción

Sitio web creado por Sports Reference que aporta información de la NBA y WNBA acerca de los jugadores, equipos, líderes en estadísticas, clasificación y demás datos de interés.

Enlace: <https://www.basketball-reference.com/>



Ilustración 1-1 Icono Basketball Reference

Players	Teams	Seasons	Leaders	Scores 1	WNBA	Draft	Stathead
---------	-------	---------	---------	-----------------	------	-------	----------

Basketball Stats and History Statistics, scores, and history for the NBA, ABA, WNBA, and top European competition.

Every NBA & Every WNBA Player



View any Active Player:
 Choose a team
 ... then a player

Select a Hall of Famer:
 Select a player

Recent Debuts
[Chance Comanche](#) (POR), [Jacob Gilyard](#) (MEM), [Raiquan Gray](#) (BRK), [Justin Minaya](#) (POR), [Donovan Williams](#) (ATL) and [Jeenathan](#)

Every NBA Team

2022-23 NBA Standings
[Summary](#) · [Schedule](#) · [Scores](#) · [Leaders](#) · [Rookies](#)

East	W	L	West	W	L
MIL * (1)	58	24	DEN * (1)	53	29
BOS * (2)	57	25	MEM * (2)	51	31
PHI * (3)	54	28	SAC * (3)	48	34
CLE * (4)	51	31	PHO * (4)	45	37
NYK * (5)	47	35	LAC * (5)	44	38
BRK * (6)	45	37	GSW * (6)	44	38
MIA * (7)	44	38	LAL * (7)	43	39
ATL * (8)	41	41	MIN * (8)	42	40
TOR * (9)	41	41	NOP * (9)	42	40
CHI * (10)	40	42	OKC * (10)	40	42
IND x (11)	35	47	DAL x (11)	38	44
WAS x (12)	35	47	UTA x (12)	37	45
ORL x (13)	34	48	POR x (13)	33	49
CHO x (14)	27	55	HOU x (14)	22	60
DET x (15)	17	65	SAS x (15)	22	60

* - Clinched playoffs; x - Eliminated
 » See [standings as of any historical date](#)
Playoff Probabilities:

Ilustración 1-2 Captura Basketball Reference

1.3.1.1.2 Ventajas

- Glosario con el significado de las estadísticas (tooltips)
- Dispone de una sección de ayuda

1.3.1.1.3 Desventajas

- Demasiada información mezclada que no tiene relación directa
- Muy poco visual
- Mala estructuración del contenido
- Solo información de ligas estadounidenses de baloncesto

1.3.1.2 NBA Web

1.3.1.2.1 Descripción

Sitio oficial de la NBA, aunque también aporta datos de otros deportes como fútbol, tenis, fórmula 1 y fútbol americano. Ofrece toda la información posible de la NBA, incluyendo estadísticas de jugadores, equipos y partidos, todos los resultados, calendario, clasificación y noticias del día a día sobre la NBA.

Enlace: nba.com



Ilustración 1-3 Icono de sitio de la NBA

The Sporting News
EST 1886

NBA.com Equipo Global Establecer edición predeterminada

FÚTBOL NBA NFL F1 TENIS España

Noticias NBA ID Resultados Calendario Clasificación Estadísticas ...

JUN 10 Final **JUN 13** Final
SAB 1 PARTIDO MAR 1 PARTIDO
DEN 108 - MIA 95 MIA 89 - DEN 94
DEN VICTS 4 - 1 DEN VICTS 4 - 1

NOTICIAS DE NBA

Ilustración 1-4 Captura sitio de la NBA

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 21 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	--

1.3.1.2.2 Ventajas

- Toda la información necesaria
- Visual y atractiva
- Información bien estructurada

1.3.1.2.3 Desventajas

- Demasiados anuncios que reducen el área del sitio web como se puede apreciar en Ilustración 1-4. Especialmente molesto.
- No contiene un glosario o leyenda con el significado de cada estadística ni una sección de ayuda por lo que he podido observar
- Solo información de la NBA.

1.3.1.3 ESPN Deportes

1.3.1.3.1 Descripción

Página web realizada por ESPN Enterprises, muy completa, al igual que nba.com ya que aporta toda la información posible, pero tan solo de la NBA.

Enlace: <https://www.espn.com.mx/basquetbol>



Ilustración 1-5 Icono ESPN Deportes

Ilustración 1-6 Captura ESPN Deportes

1.3.1.3.2 Ventajas

- La más visual y atractiva para el usuario
- Toda la información necesaria
- Información bien estructurada
- Sin anuncios molestos
- Contiene glosario explicativo con el significado de las estadísticas

1.3.1.3.3 Desventajas

- En algunos lugares demasiada información y a un tamaño algo pequeño.
- Solo información de la NBA.
- No dispone de una sección de ayuda

1.3.1.4 Marca

1.3.1.4.1 Descripción

Diario español de información deportiva, realizado por la empresa *Unidad Editorial*, no tan solo de baloncesto que es lo analizado en este caso, sino también de fútbol, motor, tenis y demás.

Enlace: <https://www.marca.com/>



Ilustración 1-7 Icono Marca

Es noticia: Tour de Francia / Mercado fichajes / Djokovic / Etapa 3 Tour de Francia TV / Partidos Wimbledon hoy / Wimbledon 2023 cuadro / Djokovic donde ver / Alonso Aston Martin / Mbappé / Iza Almansa / GP Gran Bretaña /

MARCA ES Fútbol Baloncesto Motor Polideportivo Coches Más + Suscríbete Iniciar sesión

Baloncesto / Noticias Resultados Eurobasket femenino Liga Endesa Liga Femenina Endesa NBA Euroliga Eurocup Más +

PUBLICIDAD

VOLVO
Volvo C40 totalmente eléctrico DESCUBRE MÁS

Noticias de Baloncesto

SELECCIÓN ESPAÑA BALONCESTO
El MVP Iza Almansa: otro español carne de NBA... que rompió las reglas
EUGENIO MUÑOZ FERNÁNDEZ 170

Ilustración 1-8 Captura Marca

1.3.1.4.2 Ventajas

- Información de varias ligas, estadounidenses, europeas y de género masculino y femenino
- Dispone de sección de ayuda

1.3.1.4.3 Desventajas

- No dispone de toda la información de la que debería (sección jugadores, detalle de sus estadísticas...)
- Secciones como la de equipos con mala estructuración y poco visuales
- Demasiada publicidad

1.3.1.5 As

1.3.1.5.1 Descripción

Al igual que Marca, As es otro diario deportivo español editado por el Grupo Prisa. Contiene información de deportes como el baloncesto, tenis, ciclismo, fútbol y motor aportando los datos de diferentes ligas y competiciones.

Enlace: <https://as.com/>

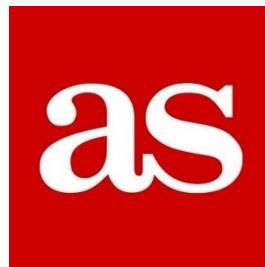


Ilustración 1-9 Icono As

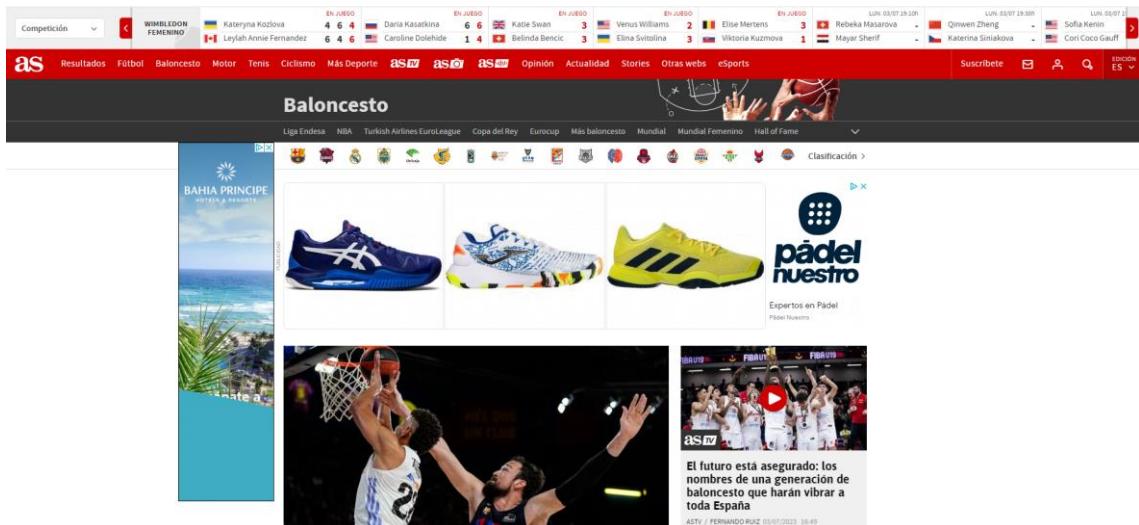


Ilustración 1-10 Captura Diario As

1.3.1.5.2 Ventajas

Aporta información de varias ligas

1.3.1.5.3 Desventajas

- Mucha publicidad
- Fallos en el uso de algún apartado como presionar en un equipo
- No dispone de toda la información relevante.
- Estructuración mejorable
- No dispone de ayuda

1.3.1.6 Conclusiones

En definitiva, tras el estudio de los diferentes sitios a los que podemos acceder relacionados con el dominio del baloncesto, tenemos planteado obtener una aplicación que permita consumir los datos de mayor interés para el usuario, de una forma visual y accesible consiguiendo la mejor experiencia posible. Hay que tener también en cuenta el alcance de dicho proyecto, simplificándose a un Trabajo de Fin de Grado en el que no se podrá entrar en tanto detalle como en las aplicaciones profesionales descritas en dicho apartado.

Por último, vamos a realizar una tabla con las secciones que se tratarán de cubrir en este proyecto, haciendo la comparativa con las alternativas estudiadas. En la tabla, para cada punto se marcarán con una 'X' los sitios que lo cumplan.

	Basketball references	Nba.com	Espn deportes	Marca	As
Sección de ayuda	X			X	
Demasiada publicidad		X	X	X	X
Buena estructuración		X	X		
Toda la información relevante (al menos la más básica)		X	X		
Información de varias ligas de distintas nacionalidades				X	X
Contiene glosario para las estadísticas	X		X		
Visualmente atractiva		X	X		

Tabla 1 Comparativa sitios web de baloncesto estudiados

Hemos de tener en cuenta que es normal la elevada cantidad de anuncios en este tipo de sitios web, con un alto flujo de personas, lo que hace que se convierten en algo comercial.

En cuanto a la sección de ayuda, todos los sitios tienen la obligada sección de política de privacidad, así como una sección de contáctanos, pero solo dos tienen una sección de ayuda como tal o de preguntas y respuestas.

Los diarios españoles Marca y As, se centran más en otros deportes como el fútbol, donde si hay una excelente estructuración de la información además de contener todos los datos de interés, pero en el baloncesto esto es escaso.

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 26 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	--

En este proyecto se va a tratar de cubrir al máximo estos puntos, quizás no disponer de una sección de ayuda pero sí de una introducción al sitio cuando el usuario entre a este, teniendo la información básica de interés, estando lo mejor estructurada posible, siendo visual y atractiva, disponiendo del glosario para explicar los siglas de las estadísticas, y demostrando que es el sistema es escalable para consumir datos de cualquier otra liga (nos centraremos en la NBA ya que es la liga que más información facilita y de manera más sencilla).

Capítulo 2. Aspectos Teóricos

Sección en la que se explicarán todas las herramientas, lenguajes, frameworks y librerías utilizadas en el proyecto.

2.1 HTML

HTML



Su origen se encuentra en 1980 cuando Tim Berners-Lee quería crear una forma de crear y compartir contenido en la web de forma que todos pudieran acceder a él.

Es el lenguaje que utilizan los navegadores web para interpretar el contenido de los sitios web. Por tanto, este lenguaje de marcado se utiliza para crear tanto la estructura de una página web a través de elementos como encabezados, párrafos, imágenes, etc., como su contenido.

Algunas aplicaciones:

- Páginas web
- Aplicaciones móviles
- Aplicaciones de escritorio

Se puede obtener más información en <https://www.w3.org/TR/html52/>.

2.2 CSS

CSS



Su primera versión fue lanzada en 1996 debido al anterior estudio de cómo poder separar la estructura de un sitio del estilo que se le quería aplicar.

Es el lenguaje de hojas de estilo que se va a encargar de aplicarle un estilo visual a los elementos HTML a través de selectores y reglas. Es decir, define el estilo visual de un sitio web aplicando colores, fuentes, tamaños y definiendo la disposición de los elementos.

Algunas aplicaciones:

- Páginas web
- Aplicaciones móviles
- Juegos

Se puede obtener más información en <https://www.w3.org/Style/CSS/>.

2.3 Typescript



Lenguaje de programación tipado estáticamente que se basa en Javascript y se utilizar para el desarrollo de la lógica en aplicaciones web. Al ser compatible con las últimas versiones de JavaScript es posible utilizar características de este último en los archivos TypeScript. Es super popular para los frameworks de JavaScript como Angular, React y Vue.js.

Algunas de las ventajas que presenta sobre Javascript gracias al tipado estático:

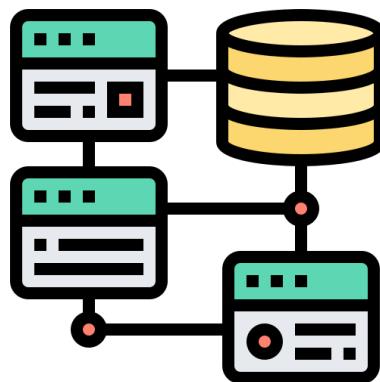
- Prevenir errores
- Código más fácil de mantener
- Código más fácil de depurar

Algunas aplicaciones de TypeScript son:

- Páginas web
- Aplicaciones de escritorio
- Juegos
- Herramientas de línea de comandos

Se puede obtener más información en <https://www.typescriptlang.org/docs/>.

2.4 Base de datos relacional



Son una forma de almacenar y organizar datos en tablas. Dentro de una tabla, cada fila representa un registro individual y cada columna un atributo de dicho registro, además de tener que contener un valor único conocido como clave primaria.

Son importantes ya que permiten el almacenamiento de una gran cantidad de datos de una manera organizada y accesible, de tal forma que facilita a las empresas a procesar y analizar datos de forma más efectiva.

Este tipo de bases de datos trabajan siguiendo, como es lógico, el modelo relacional, donde se generan relaciones entre tablas, las cuales están diseñadas para minimizar la redundancia de datos y lograr la integridad de los datos. Las relaciones entre las tablas se generan a través de las claves primarias y externas. Este tipo de bases de datos utilizan el lenguaje de consulta estructurado (SQL).

En este proyecto se utilizará una base de datos de este tipo.

2.5 Angular



Angular fue desarrollado por Misko Hevery y Adam Abrons en 2009 como herramienta interna para la construcción de aplicaciones web en Google, de hecho, en la actualidad está mantenido por el propio Google. La primera versión lanzada de este fue AngularJS, que ha ido actualizándose con el paso de los años. En este proyecto yo usaré Angular 13.

En concreto es un framework que se usa para desarrollar aplicaciones web de una sola pantalla, lo que se conoce como SPA (Single Page Applications) usando el lenguaje TypeScript. Su objetivo es desarrollar aplicaciones web que sean complejas y escalables, ofreciendo herramientas y utilidades que permiten que las interfaces de usuario sean dinámicas y responsivas.

Por ejemplo, de entre todas las funcionalidades que ofrece este framework las más destacables son las siguientes:

- Componentes: utilizados para crear la interfaz de usuario, encapsulando por cada uno de ellos un archivo HTML, CSS y ts encargado de la lógica de negocio.
- Inyección de dependencias: para administrar las dependencias y componentes de la aplicación (ficheros package.json, node_modules...).
- Enrutamiento: sistema de enrutamiento para crear rutas que nos permiten navegar entre las “páginas” de la aplicación. Realmente se recarga la misma página con el contenido de otro componente.
- Directivas: característica de Angular que nos permite incorporar nuevos atributos o elementos al HTML.
- Pipes: herramienta para transformar los datos en tiempo real y en la vista de la aplicación.

Algunos de los sitios que utilizan Angular son:

- PayPal

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 30 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	--

- Google Ads
- NBA.com

Para obtener más información sobre Angular mi recomendación es que se visite su documentación a través de este enlace: <https://angular.io/docs>. Aporta guías, tutoriales, ejemplos y referencias para la creación de aplicaciones web con Angular.

En este proyecto Angular será utilizado para realizar el desarrollo frontend del mismo. Se comunicará con el backend realizado con java Spring Boot explicado posteriormente, haciendo peticiones al mismo y procesando la información recibida de este. La herramienta que se va a utilizar para trabajar con esta tecnología es Visual Studio Code.

2.6 Chart.js



Chart.js es una biblioteca de JavaScript de código abierto que permite crear gráficos interactivos y visualmente atractivos en páginas web. La originó Nick Downie en 2013 como un proyecto personal, ahora es mantenida por una gran comunidad de desarrolladores.

Se utiliza para representar datos en forma de gráficos en tiempo real o estáticos en páginas web. Proporciona una amplia variedad de tipos de gráficos, como gráficos de líneas, barras, radar, área, pastel y más. Estos gráficos se pueden personalizar fácilmente para adaptarse a las necesidades específicas de un proyecto.

Algunas de las ventajas que presenta esta biblioteca son:

- Interactividad: los usuarios pueden interactuar con los gráficos.
- Personalización: Chart.js ofrece una amplia gama de opciones de personalización para adaptar los gráficos al diseño y estilo del sitio web.
- Responsive: Los gráficos de Chart.js son responsivos y se adaptan automáticamente al tamaño de la pantalla o contenedor en el que se encuentren.

Si se desea obtener más información esta es la documentación oficial:

<https://www.chartjs.org/docs/>

2.7 Bootstrap



Bootstrap es un framework de desarrollo web frontEnd de código abierto que fue creado por Twitter en 2011. Su objetivo principal es proporcionar un conjunto de herramientas y componentes predefinidos que faciliten la creación de interfaces web modernas y responsivas.

Está basado en HTML, CSS y JavaScript ofreciendo estilos prediseñados y componentes reutilizables que ayudan a estructurar el contenido del sitio web.

Algunas de las ventajas que presenta son las siguientes:

- Diseño responsive: está diseñado para adaptarse a diferentes tamaños de pantalla y dispositivos.
- Componentes y estilos predefinidos: como botones, barras de navegación, formularios, y mucho más.
- Cuadrícula flexible: sistema de cuadrícula flexible basado en columnas que permite organizar y alinear fácilmente el contenido de una página web.
- Compatibilidad con una amplia gama de navegadores.

El enlace al sitio oficial es: <https://getbootstrap.com/>.

2.8 SpringBoot



Fue desarrollado por Pivotal Software, una compañía de desarrollo software empresarial. Actualmente la versión que utilizaremos en este proyecto es la v3.0.3.

Estamos hablando de un framework de desarrollo de aplicaciones java de forma rápida y sencilla al facilitarnos la configuración. También nos aporta para ello convenciones predefinidas.

Vamos a hablar ahora de algunas de las herramientas y funcionalidades que tiene este framework:

- Configuración automática: permite detectar y configurar automáticamente las dependencias y componentes de la aplicación.
- Arranque rápido: permite crear aplicaciones web incluso sin bases de datos en cuestión de minutos gracias a esta característica.
- Integración sencilla: permite una conexión sencilla con otros frameworks como puede ser Spring Security.

- Despliegue fácil: permite desplegar con facilidad en distintos entornos como servidores web o en la nube.

Algunas aplicaciones o sitios que utilizan esta herramienta son:

- Netflix
- Dell
- Alibaba

Para obtener información más detallada de esta tecnología recomiendo visitar la documentación oficial <https://docs.spring.io/spring-boot/docs/current/reference/html/>. Se podrá acceder a tutoriales y ejemplos, así como a una lista de sus características y herramientas.

El uso que le daremos en este proyecto es para la realización del backend, conectándose con Angular al que le enviará los datos solicitados, así como con la Base de Datos MySQL a través de MyBatis. El entorno de trabajo será spring tool suite, que es un eclipse configurado específicamente para Spring Boot.

2.9 Python



Fue creado a finales de los años 80 por Guido van Rossum en los Países bajos. La versión que utilizamos en este proyecto es la 3.9.13.

Es un lenguaje de programación interpretado de alto nivel y de propósito general que tiene una gran facilidad de uso gracias a su legibilidad y sintaxis intuitiva, nosotros lo vamos a utilizar para la extracción de datos ya que contiene módulos muy útiles para esta aplicación como el que vamos a usar 'Pandas' pero también otros como Scrapy. El entorno de trabajo que utilizaremos será Spyder, de Anaconda.

Además de para extracción de datos, permite la programación orientada a objetos, funcional y estructurada.

- Algunas de las aplicaciones de Python son:
- Desarrollo web (Django)
- Análisis de datos (ej.: módulo NumPy)
- Inteligencia artificial y aprendizaje automático (ej.: librería PyTorch)
- Automatización de tareas

La documentación oficial se encuentra en este enlace: <https://docs.python.org/3/>. Incluye información sobre la sintaxis del lenguaje, la estructura de programas Python y las herramientas y módulos de los que dispone.

2.10 MySQL



Creado en 1995 por Michael Widenius y David Axmark y desarrollado por MySQL AB y por la comunidad de desarrolladores de software libre. Actualmente propiedad de Oracle.

Es un sistema gestor de bases de datos relacionales de código abierto que se utilizar para almacenar y recuperar datos, que permite manejar una gran cantidad de datos y procesar consultas complejas en tiempo real. Es especialmente popular para aplicaciones web y de comercio electrónico.

Características principales:

- Fácil de usar
- Escalable
- Seguro
- Para aplicaciones críticas y sensibles
- Personalizable
- Buena integración

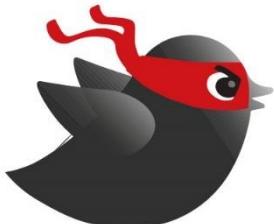
Algunas de las aplicaciones más conocidas que utilizan MySQL son:

- WordPress
- Facebook
- Twitter

La documentación oficial se encuentra en el siguiente enlace: <https://dev.mysql.com/doc/>.

En nuestro proyecto utilizaremos este gestor de base de datos.

2.11 MyBatis



Se originó en 2009 como la evolución de iBatis. Un año después Apache Software Foundation lo llevó a un nivel superior.

Es un framework de persistencia de datos utilizado en aplicaciones Java que interaccionan con bases de datos relacionales. Permite asignar consultas SQL a métodos de Java de tal forma que dentro de un objeto de este lenguaje de programación podamos hacer la solicitud, recuperación y mapeo de los resultados obtenidos de la base de datos.

Algunas de las ventajas que presenta son las siguientes:

- Flexibilidad
- Facilidad de uso
- Control sobre la interacción con la base de datos
- Consultas optimizadas y personalizadas
- Mapeo de resultados optimizado
- Soporte para procedimientos almacenados

Algunos ejemplos de uso son

- Sistema de gestión de contenido
- Sistema de gestión de relaciones con clientes
- Aplicaciones de comercio electrónico
- Aplicaciones financieras

Se puede acceder a más información en el siguiente enlace de la documentación original:
<https://mybatis.org/mybatis-3/es/index.html>.

En nuestro proyecto lo utilizaremos como framework para la interacción de nuestro backend con la base de datos.

2.12 Patrón MVC



MVC (Modelo-Vista-Controlador) es un patrón de diseño de software que se utiliza para estructurar y organizar el código en aplicaciones de software. Fue introducido por primera vez en la década de 1970 por Trygve Reenskaug como parte del desarrollo del lenguaje de programación Smalltalk-76 en Xerox PARC.

El patrón MVC separa la lógica de la aplicación en tres componentes principales:

- **Modelo:** Representa los datos y la lógica de negocio de la aplicación. El modelo no tiene conocimiento de la interfaz de usuario o de cómo se visualizan los datos.
- **Vista:** Es la representación visual de los datos del modelo. La vista es responsable de la presentación de la información al usuario final. La vista no realiza cambios en los datos, sino que solo muestra la información proporcionada por el modelo.
- **Controlador:** Actúa como intermediario entre el modelo y la vista. Responde a las acciones del usuario y actualiza el modelo y la vista según sea necesario.

Algunas de las ventajas que presenta son las siguientes:

- Buena modularidad
- Fácil mantenimiento del código
- Fácil reutilización del código

Algunos de los usos del patrón MVC son:

- Aplicaciones web
- Aplicaciones móviles
- Aplicaciones de escritorio

Frameworks y herramientas de desarrollo como Ruby on Rails o Django lo utilizan.

2.13 Visual Studio Code



IDE lanzado en 2015 por Microsoft para Windows, Linux y macOS. Ofrece características como resaltado de sintaxis, autocompletado, depuración, control de versiones, que facilitan el trabajo a los desarrolladores.

También dispone de un gran listado de extensiones y temas para personalizar el entorno según los gustos y necesidades del programador.

Se utiliza para programar en lenguajes como JavaScript, TypeScript, Python, C++. En nuestro caso lo usaremos para programar en Angular, por lo que editaremos fichero TypeScript, HTML y CSS con él.

Se usa para aplicaciones de cualquier tipo, tanto personales como profesionales y se caracteriza por su rapidez, facilidad de uso y potentes herramientas.

El enlace al sitio oficial de Visual Studio Code es: <https://code.visualstudio.com/>.

2.14 Spring Tool Suite



IDE basado en Eclipse, desarrollado por Pivotal Software, que se utiliza para desarrollar aplicaciones en el framework de Spring, al disponer de herramientas y características específicas para este framework. Su popularidad ha incrementado gracias al desarrollo basado en microservicios.

Algunas de las características que ofrece STS son:

- Crear proyectos de Spring con rapidez
- Autocompletado de código
- Control de versiones
- Generación de código
- Depuración
- Buena integración con otras herramientas (Spring Cloud)

Su sitio web oficial para obtener más información es: <https://spring.io/tools/sts>.

2.15 MySQL Workbench



Herramienta que fue lanzada en 2006, y al igual que el propio MySQL fue desarrollada por MySQL AB y actualmente es propiedad de Oracle.

Esta herramienta se utiliza para el diseño, desarrollo y administración de bases de datos relacionales MySQL. Permite realizar las operaciones comunes sobre una base de datos, como puede ser crear o modelar la propia base de datos, a través de una interfaz gráfica.

Entre todas las funcionalidades que presenta está la creación y modificación de esquemas, edición de tablas y campos de estas, creación de consultas SQL, generación de informes y administración de usuarios y sus permisos.

No solo permite la gestión de bases de datos, sino que también admite la integración con herramientas de control de versiones.

El sitio oficial es: <https://www.mysql.com/products/workbench/>.

2.16 Anaconda – Spyder



Fue creado por la comunidad de desarrolladores de Anaconda, que es una distribución de Python que contiene diversos IDEs y se utiliza para el análisis de datos científicos y computación numérica.

Spyder es un IDE que ha sido diseñado específicamente para Python. Algunas de las características que tiene son:

- Edición de código
- Depuración
- Perfilado
- Exploración de variables
- Visualización de datos
- Herramientas y bibliotecas integradas

Aplicaciones principales:

- Ciencia de datos
- Aprendizaje automático
- Programación general en Python
- Útil para áreas como la física, biología y estadística

El enlace al sitio oficial es: <https://www.spyder-ide.org/>.

Capítulo 3. Planificación y Gestión del proyecto

Se ha realizado al principio del proyecto una planificación dividiendo la misma en las siguientes fases:

Tutorías: reuniones que se van a realizar a lo largo del proyecto con el tutor del Trabajo de Fin de Grado. Es una tarea periódica que se realizará en 1 hora cada viernes.

Planificación y gestión del proyecto: proceso en el que se identifican las tareas del proyecto, asignándoles los recursos necesarios. Identificamos también posibles riesgos y elaboramos el presupuesto.

Investigación: proceso inicial que consiste en estudiar páginas web similares, fuentes de información y tecnologías, frameworks y librerías a utilizar.

Análisis: fase en la que se definirá el sistema, se especificarán los requisitos, se analizarán los casos de uso y finalmente se especificará el plan de pruebas.

Diseño: proceso que se compone de definir la arquitectura del sistema, del diseño de la base de datos, de los prototipos de la interfaz de usuario y de los diagramas de clases y de interacción relacionados con los casos de uso del sistema.

Implementación: fase en la que se realizará la implementación de la extracción de los datos, de la base de datos, así como del almacenamiento en ella de los datos mencionados, de la lógica de negocio y de la realización de la interfaz de usuario final.

Desarrollo de las pruebas: fase que estará compuesta de la implementación de las pruebas unitarias, de sistema, de calidad y de usabilidad.

Todas estas fases incluyen en sí mismas la realización de su documentación.

Las planificaciones que se verán a continuación serán adjuntadas junto con este proyecto, con los nombres 'PlanificacionInicial.mpp' y 'PlanificacionFinal.mpp'.



3.1 Planificación Inicial

Inicialmente se realizó una planificación aproximada de lo que podía ser el coste de tiempo que suponía la realización del proyecto. Esta planificación se obtuvo del conocimiento propio, que como se puede observar en Ilustración 3-1 Planificación Inicial es reducido, ya que resulta bastante optimista, con un total de 285 horas teniendo en cuenta que el máximo establecido para el Trabajo de Fin de Grado son 300 horas, acabando el proyecto a finales de mayo.

Se ha considerado un calendario de 4 horas de lunes a viernes.

TFG	285 hrs	vie 10/02/23	lun 22/05/23
▶ Tutorías	261 hrs	vie 10/02/23	vie 12/05/23
▶ Investigación	20 hrs	vie 10/02/23	jue 16/02/23
▶ Análisis	64 hrs	vie 17/02/23	vie 10/03/23
▶ Diseño	42 hrs	lun 13/03/23	lun 27/03/23
▶ Implementación	106 hrs	lun 27/03/23	mar 02/05/23
▶ Desarrollo de pruebas	38 hrs	mié 03/05/23	mar 16/05/23
Revisión de la documentación	15 hrs	mar 16/05/23	lun 22/05/23

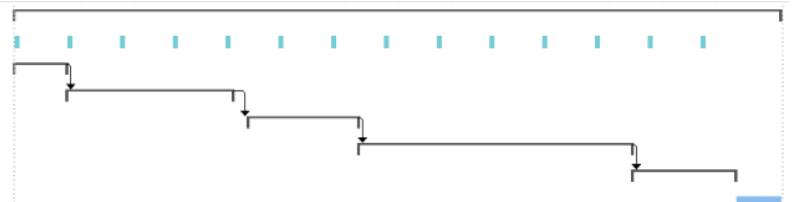


Ilustración 3-1 Planificación Inicial

En el apartado [3.3 Estructura de desglose de trabajo](#) se hace un desglose de trabajo de todas las tareas principales, ya con el tiempo real que nos ha llevado cada tarea.

3.2 Planificación Final

Como digo en el [apartado anterior](#) la planificación inicial fue optimista, acabando a mediados de mayo el proyecto. Finalmente, debido a la carga simultánea de las asignaturas de la universidad, también a la poca experiencia en la planificación, el no haber tenido en cuenta ciertas tareas y ver que pueden surgir más problemas de los esperados, la planificación se ha tenido que ver modificada como se aprecia en Ilustración 3-2 Planificación Final.

En esta planificación vemos que fases como la de implementación nos llevó bastante más de lo previsto, pero todas en general incrementaron el tiempo de realización del proyecto, quedando finalmente en 345 horas aproximadamente, 70 horas más de lo estimado al principio del proyecto.

TFG	345 hrs	vie 10/02/23	lun 12/06/23
▶ Tutorías	341 hrs	vie 10/02/23	vie 09/06/23
▶ Investigación	20 hrs	vie 10/02/23	jue 16/02/23
▶ Planificación y gestión del proyecto	20 hrs	vie 17/02/23	jue 23/02/23
▶ Análisis	64 hrs	vie 24/02/23	vie 17/03/23
▶ Diseño	50 hrs	lun 20/03/23	mié 05/04/23
▶ Implementación	124 hrs	mié 05/04/23	jue 18/05/23
▶ Desarrollo de pruebas	38 hrs	jue 18/05/23	mié 31/05/23
Realizar documentación de Apéndices	14 hrs	jue 01/06/23	mar 06/06/23
Revisión de la documentación	15 hrs	mar 06/06/23	lun 12/06/23

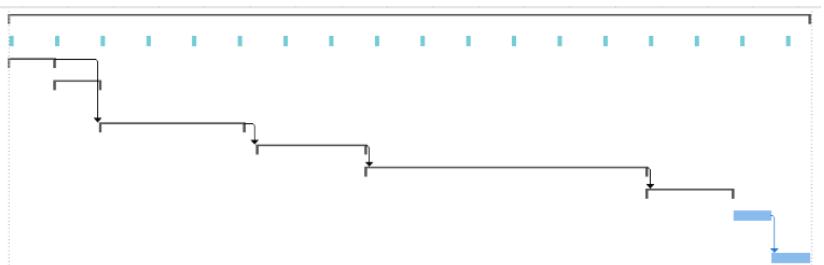


Ilustración 3-2 Planificación Final



3.3 Estructura de desglose de trabajo

Vamos a desglosar a continuación todos los hitos definidos en las planificaciones en las tareas que las componen.

3.3.1 Tutorías

Esta sección de la planificación hace referencia a las reuniones que se realizarán con el tutor del proyecto a lo largo del desarrollo de este. Cada reunión tiene una duración de 1 hora.

▪ Tutorías	341 hrs	vie 10/02/23	vie 09/06/23
Realizar 1º tutoría	1 hr	vie 10/02/23	vie 10/02/23
Realizar 2º tutoría	1 hr	vie 17/02/23	vie 17/02/23
Realizar 3º tutoría	1 hr	vie 24/02/23	vie 24/02/23
Realizar 4º tutoría	1 hr	vie 03/03/23	vie 03/03/23
Realizar 5º tutoría	1 hr	vie 10/03/23	vie 10/03/23
Realizar 6º tutoría	1 hr	vie 17/03/23	vie 17/03/23
Realizar 7º tutoría	1 hr	vie 24/03/23	vie 24/03/23
Realizar 8º tutoría	1 hr	vie 31/03/23	vie 31/03/23
Realizar 9º tutoría	1 hr	vie 07/04/23	vie 07/04/23
Realizar 10º tutoría	1 hr	vie 14/04/23	vie 14/04/23
Realizar 11º tutoría	1 hr	vie 21/04/23	vie 21/04/23
Realizar 12º tutoría	1 hr	vie 28/04/23	vie 28/04/23
Realizar 13º tutoría	1 hr	vie 05/05/23	vie 05/05/23
Realizar 14º tutoría	1 hr	vie 12/05/23	vie 12/05/23
Realizar 15º tutoría	1 hr	vie 19/05/23	vie 19/05/23
Realizar 16º tutoría	1 hr	vie 26/05/23	vie 26/05/23
Realizar 17º tutoría	1 hr	vie 02/06/23	vie 02/06/23
Realizar 18º tutoría	1 hr	vie 09/06/23	vie 09/06/23

Ilustración 3-3 Desglose tutorías

3.3.2 Investigación

Esta fase consiste en analizar sitios web relaciones con el dominio del proyecto con el objetivo de tener una base, qué podemos aprovechar de estos y qué podemos mejorar.

También incluye la correspondiente investigación sobre las tecnologías que mejor se adaptan a nuestro proyecto, así como la búsqueda de APIs que nos aporten la información deseada.

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 41 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	--

▪ Investigación	30 hrs	vie 10/02/23	mar 21/02/23
Investigar páginas similares	8 hrs	vie 10/02/23	lun 13/02/23
Investigar las tecnologías	12 hrs	mar 14/02/23	jue 16/02/23
Investigar APIs	10 hrs	vie 17/02/23	mar 21/02/23

Ilustración 3-4 Desglose Investigación

3.3.3 Planificación y gestión del proyecto

Esta fase contiene las tareas de realizar la planificación y desglose de tareas del proyecto, incluyendo también la realización del plan de gestión de riesgos y el presupuesto del proyecto.

▪ Planificación y gestión del proyecto	20 hrs	vie 17/02/23	jue 23/02/23
Realizar la planificación del proyecto	7 hrs	vie 17/02/23	lun 20/02/23
Identificar y estudiar riesgos	5 hrs	lun 20/02/23	mar 21/02/23
Realizar el presupuesto del proyecto	8 hrs	mié 22/02/23	jue 23/02/23

Ilustración 3-5 Desglose Planificación

3.3.4 Análisis

Este hito se centra en hacer un primer análisis del proyecto, definir alcance y objetivos, requisitos a cumplir, así como casos de uso y escenarios, diagramas esquemáticos, prototipos de interfaces, y especificar el plan de pruebas.

▪ Análisis	64 hrs	mar 21/02/23	mié 15/03/23
Definir el sistema	7 hrs	mar 21/02/23	jue 23/02/23
Obtener los requisitos del sistema	9 hrs	jue 23/02/23	lun 27/02/23
Identificar los subsistemas	5 hrs	lun 27/02/23	mar 28/02/23
Realizar el diagrama de clases preliminar	9 hrs	mar 28/02/23	jue 02/03/23
Analizar casos de uso y escenarios	12 hrs	vie 03/03/23	mar 07/03/23
Analizar las interfaces de usuario	12 hrs	mié 08/03/23	vie 10/03/23
Especificar el plan de pruebas	10 hrs	lun 13/03/23	mié 15/03/23

Ilustración 3-6 Desglose Análisis

3.3.5 Diseño

Fase que se encarga de desarrollar más en profundidad lo definido en el análisis. Incluye definir la arquitectura del sistema, así como el desarrollo de las clases, con sus diagramas, diagramas de secuencia entre los distintos objetos en escenarios de uso concretos, diseño de la base de datos y de la interfaz y, por último, la especificación técnica de las pruebas.

- Diseño	50 hrs	lun 20/03/23	mié 05/04/23
Definir la arquitectura del sistema	5 hrs	lun 20/03/23	mar 21/03/23
Diseñar las clases	12 hrs	mar 21/03/23	vie 24/03/23
Crear diagramas de interacción	2 hrs	vie 24/03/23	vie 24/03/23
Diseñar la base de datos	10 hrs	vie 24/03/23	mié 29/03/23
Diseñar la interfaz	14 hrs	mié 29/03/23	lun 03/04/23
Especificar técnicamente el plan de pruebas	7 hrs	lun 03/04/23	mié 05/04/23

Ilustración 3-7 Desglose Diseño

3.3.6 Implementación

Fase más larga del proyecto que consiste en la implementación de todos los módulos y subsistemas del sitio web.

- Implementación	124 hrs	mié 05/04/23	jue 18/05/23
Procesar los datos de las APIs	12 hrs	mié 05/04/23	lun 10/04/23
Crear la Base de Datos	10 hrs	lun 10/04/23	mié 12/04/23
Crear la interfaz de usuario	12 hrs	jue 13/04/23	lun 17/04/23
- Lógica de negocio	90 hrs	mar 18/04/23	jue 18/05/23
Visualizar la clasificación de distintas ligas	12 hrs	mar 18/04/23	jue 20/04/23
Visualizar todos los jugadores	10 hrs	vie 21/04/23	mar 25/04/23
Visualizar todos los equipos	10 hrs	mar 25/04/23	jue 27/04/23
Visualizar estadísticas avanzadas de un jugador	10 hrs	vie 28/04/23	mar 02/05/23
Visualizar estadísticas avanzadas de un equipo	10 hrs	mar 02/05/23	jue 04/05/23
Visualizar la plantilla de un equipo	8 hrs	vie 05/05/23	lun 08/05/23
Visualizar estadísticas de un partido	10 hrs	mar 09/05/23	jue 11/05/23
Comparar dos jugadores	10 hrs	jue 11/05/23	lun 15/05/23
Comparar dos equipos	10 hrs	mar 16/05/23	jue 18/05/23

Ilustración 3-8 Desglose Implementación

3.3.7 Desarrollo de pruebas

Fase que abarca la realización de los distintos tipos de pruebas que se van a realizar en el proyecto, unitarias para probar cada componente de manera individual, de calidad para asegurarnos un código limpio y mantenible, de sistema para garantizar una correcta integración entre los componentes y de usabilidad para que los usuarios interactúen con nuestra aplicación y ver posibles mejoras.

Desarrollo de pruebas	38 hrs	jue 18/05/23	mié 31/05/23
Pruebas unitarias	10 hrs	jue 18/05/23	lun 22/05/23
Pruebas de calidad	9 hrs	mar 23/05/23	jue 25/05/23
Pruebas del sistema	10 hrs	jue 25/05/23	lun 29/05/23
Pruebas de usabilidad	9 hrs	lun 29/05/23	mié 31/05/23

Ilustración 3-9 Desglose Desarrollo de pruebas

3.3.8 Documentación

Todas las fases comentadas anteriormente llevan incorporado en el tiempo la realización del apartado que le corresponde en la documentación. Aun así existen apartados como los anexos y otros que también han de ser realizados. Por último, se tiene en cuenta una revisión final de la documentación.

Realizar documentación de Apéndices	14 hrs	jue 01/06/23	mar 06/06/23
Revisión de la documentación	15 hrs	mar 06/06/23	lun 12/06/23

Ilustración 3-10 Desglose documentación

3.4 Riesgos

En este apartado vamos a identificar aquellos riesgos que pueden afectar al proyecto durante el desarrollo de este, medir su impacto en el sistema y por último, definir la respuesta a cada uno de ellos en el caso de que ocurran.

3.4.1 Identificación de los riesgos

Se van a identificar los riesgos asignándoles un identificador, el nombre del riesgo, una pequeña descripción y su categoría.

ID	Nombre	Descripción	Categoría
R01	Desconocimiento de las tecnologías utilizadas	Es posible que se generen retrasos en el desarrollo del proyecto debido a que este es el primer contacto con las tecnologías escogidas.	Técnico
R02	Avería en el sistema utilizado	Podría darse el caso de que el sistema que contiene toda la configuración del proyecto se estropeara.	Externo
R03	Comunicación con el tutor	Que la comunicación con el tutor no fuera suficiente, efectiva o tardía puede dar lugar a problemas.	Externo
R04	Inconvenientes configurando el sistema	No solo es importante entender las tecnologías, sino que la configuración de frameworks o librerías también puede afectar en el tiempo necesario.	Técnico
R05	Carga de trabajo de la universidad	Es necesario tener en cuenta que no estamos dedicando todo el tiempo a este proyecto. También existe trabajo de otras asignaturas que puedan ocasionar indisponibilidad para el desarrollo de este sistema.	Gestión de proyectos
R06	Trabajo imprevisto	Debido a la poca experiencia en la planificación de proyectos es posible que nos surja trabajo no esperado.	Gestión de proyectos
R07	Falta de potencia en el equipo de trabajo	Que el sistema utilizado no sea lo suficientemente potente puede generar retrasos.	Externo

Tabla 2 Identificación de riesgos

3.4.2 Evaluación del impacto

Probabilidad	Muy Alta	0,90	0,05	0,14	0,27	0,50	0,81
	Alta	0,70	0,04	0,11	0,21	0,39	0,63
	Media	0,50	0,03	0,08	0,15	0,28	0,45
	Baja	0,30	0,02	0,05	0,09	0,17	0,27
	Muy Baja	0,10	0,01	0,02	0,03	0,06	0,09
Impacto		0,05	0,15	0,30	0,55	0,90	
Inapreciable		Bajo	Medio	Alto	Crítico		
0,05		0,15	0,30	0,55	0,90		
Impacto							

Tabla 3 Matriz de probabilidad e impacto

Con la matriz anterior vamos a obtener el impacto de los riesgos identificados.

Identificador	Probabilidad	Impacto				Impacto
		Coste	Tiempo	Alcance	Calidad	
R01	Alta	Bajo	Alto	Alto	Alto	0,39
R02	Baja	Crítico	Alto	Medio	Medio	0,27
R03	Baja	Bajo	Medio	Bajo	Bajo	0,09
R04	Alta	Inapreciable	Alto	Medio	Medio	0,39
R05	Muy Alta	Inapreciable	Crítico	Medio	Medio	0,81
R06	Muy Alta	Inapreciable	Crítico	Bajo	Bajo	0,81
R07	Media	Inapreciable	Medio	Bajo	Bajo	0,15

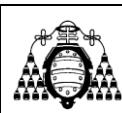
Tabla 4 Probabilidad e impacto de cada riesgo

Aquellos riesgos que se han de priorizar son los que tienen el impacto más alto. Cómo se puede observar en la tabla los riesgos R05 y R06 tendrían un impacto muy alto en el sistema.

3.4.3 Planificación de la gestión de cada riesgo

Vamos ahora a describir la respuesta a cada riesgo en caso de que suceda.

ID	Nombre	Respuesta al riesgo	Estrategia
R01	Desconocimiento de las tecnologías utilizadas	Tratar de acceder a tutoriales de las tecnologías utilizadas lo antes posible para evitar el retraso que puede originar este desconocimiento	Mitigar el riesgo
R02	Avería en el sistema utilizado	Es algo que no depende de nosotros, en caso de que suceda tendríamos que aceptarlo y obtener un nuevo equipo de trabajo lo antes posible.	Asumir el riesgo
R03	Comunicación con el tutor	Mantener al tutor informado de nuestro trabajo y planificar todas las reuniones con él	Mitigar el riesgo



		es lo que se debe hacer para no tener problemas.	
R04	Inconvenientes configurando el sistema	Acceder a tutoriales de configuración de frameworks y librerías lo antes posible	Mitigar el riesgo
R05	Carga de trabajo de la universidad	Es algo que tenemos que saber desde un principio, tenerlo en cuenta nos ayudará a realizar una planificación más exacta y evitar retrasos. De todas formas, es algo inevitable que va a aparecer sí o sí	Asumir el riesgo
R06	Trabajo imprevisto	Debido a la poca experiencia en la planificación de proyectos esto es algo que es inevitable y que nos va a suceder. La respuesta a este riesgo es tratar de replanificar lo antes posible todo el trabajo.	Asumir el riesgo
R07	Falta de potencia en el equipo de trabajo	Sustituir el sistema supondría un aumento en el coste y también repercutiría en el tiempo, es algo que tendríamos que asumir.	Asumir el riesgo

3.5 Resumen del Presupuesto

El presupuesto detallado, tanto el de costes como el del cliente, se pueden encontrar en el apartado [11.1 Detalle del presupuesto](#).

3.5.1 Resumen presupuesto de costes

Presupuesto de costes

Código	Partida	Total
01	Partida 1: Tutorías	251,28 €
02	Partida 2: Investigación	279,20 €
03	Partida 3: Planificación y gestión del proyecto	279,20 €
04	Partida 4: Análisis	893,44 €
05	Partida 5: Diseño	698,00 €
06	Partida 6: Implementación	1.731,04 €
07	Partida 7: Desarrollo de pruebas	530,48 €
08	Partida 8: Documentación	404,84 €
09	Partida 9: Otros costes	64,80 €
10	Hardware y software	543,65 €
Total		5.675,93 €

Tabla 5 Resumen presupuesto de costes

3.5.2 Resumen presupuesto de cliente

Presupuesto cliente

Código	Partida	Total
01	Partida 1: Tutorías	351,81 €
02	Partida 2: Investigación	390,90 €
03	Partida 3: Planificación y gestión del proyecto	390,90 €
04	Partida 4: Análisis	1.250,89 €
05	Partida 5: Diseño	977,26 €
06	Partida 6: Implementación	2.423,61 €
07	Partida 7: Desarrollo de pruebas	742,72 €
08	Partida 8: Documentación	566,81 €
Total		7.094,91 €

Coste sin IVA	7.094,91 €
IVA (21%)	1.489,93 €
Coste con IVA	8.584,84 €

Tabla 6 Resumen presupuesto cliente



Capítulo 4. Análisis

Este apartado contendrá toda la especificación de requisitos y toda la documentación del análisis de la aplicación, a partir de la cual se elaborará posteriormente el diseño.

4.1 Definición del Sistema

4.1.1 Diagrama de 'contexto' del sistema

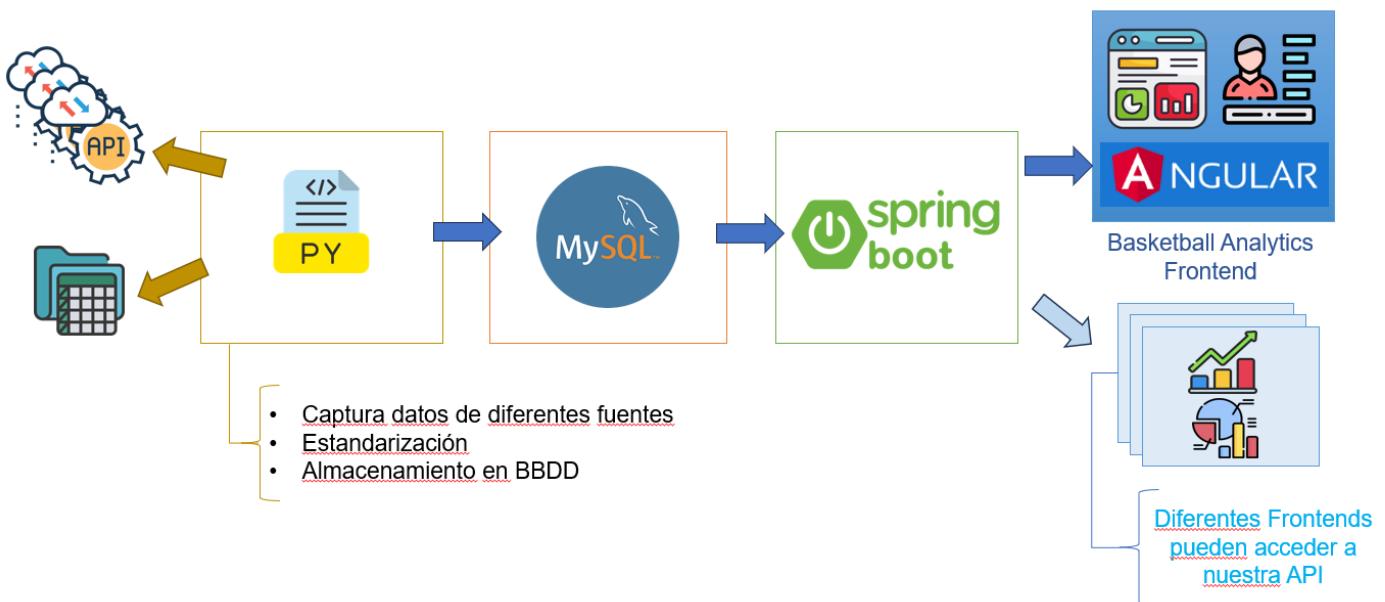


Ilustración 4-1 Diagrama de 'contexto' del sistema

Este diagrama es para darle una idea al lector de lo que es este sistema, mostrando los diferentes subsistemas que lo componen y el objetivo de estos.

4.1.2 Funcionalidades del sistema

El sistema se va a centrar en cumplir con la visualización de los datos más importantes del baloncesto profesional, especialmente de la NBA y en la representación de estos a través de gráficos y tablas para la mejor comprensión de los usuarios y que estos sean capaces de analizarlos de una manera más sencilla.

Toda esta funcionalidad se detalla más profundamente en [5.2.1 Obtención de los Requisitos del Sistema](#)

4.1.3 Estado del arte

En este apartado vamos a ver las diferentes posibilidades que hemos tenido en cuenta antes de decidir las tecnologías y fuentes de información que íbamos a utilizar para el desarrollo del proyecto.

4.1.3.1 Páginas investigadas

Visitar páginas web que trataran sobre el baloncesto es importante para así poder extraer funcionalidades, ideas o información para nuestro propio sistema. Algunas de las páginas que se han visitado son las nombradas a continuación:

- <https://www.nba.com/stats>
 - o Página oficial de la NBA, que ayuda a extraer funcionalidades para nuestra app, así como saber que datos se pueden mostrar y analizar.
- <https://spain.id.nba.com/reportajes/medir-rendimiento-jugador-nba>
 - o Para saber cómo medir el rendimiento de un jugador con un solo número
- <https://revistasuma.fespm.es/wp-content/uploads/2021/08/Las-estadísticas-avanzadas.pdf>
 - o Explicación de estadísticas avanzadas y comparación con las tradicionales. Explica por qué motivo son mejores las avanzadas, así como cuáles existen.
- https://www.espn.com.mx/basquetbol/nba/juego/_juegold/401468919
 - o Otro sitio del que extraer funcionalidades, como, por ejemplo, la comparación de equipos en un partido.
- <https://uvadoc.uva.es/bitstream/handle/10324/43847/TFG-G4618.pdf;jsessionid=C1028FDF757B7822C4F5DB44788D96D3?sequence=1>
 - o Otro TFG que consiste en el estudio de estadísticas de la NBA y del que se puede sacar información útil sobre el significado de estas.
- <https://www.sportingnews.com/es/nba/news/estadistica-avanzada-diccionario-terminos-que-es-que-sirve-nba-analytics-sabermetrics/1gc73ududvomy1s3mqiahg238c>
 - o Diccionario de las estadísticas avanzadas, muy completo para entender el dominio del problema.
- <https://www.kaggle.com/datasets/wyattowalsh/basketball>
<https://www.kaggle.com/competitions/march-machine-learning-mania-2023/data>
 - o Kaggle es una comunidad de machine learning y ciencia de datos que nos aporta datasets interesantes por si queremos distintas fuentes de información.
- <https://observablehq.com/search?query=nba&onlyOwner=false>
 - o Comunidad de la librería de gráficos D3 que me pueden ayudar a saber que datos son interesantes extraer de la API de la nba.

4.1.3.2 APIs y Librerías

Luego por otro lado es necesario buscar APIs y librerías para conseguir el objetivo buscado. Por un lado, tenemos que poder extraer los datos de algún sitio, y por otro, realizar el análisis de dichos datos mediante gráficos.

- `nba_api` (https://github.com/swar/nba_api):
 - o Es un cliente de la API de la NBA.com, con el objetivo de que sus datos sean fácilmente accesibles.
- `D3.js` (<https://d3js.org/>)
 - o Es una librería que permite hacer gráficos con datos a través de la manipulación del árbol DOM
- `ng2-charts` (<https://valor-software.com/ng2-charts/>) / `chart.js` (<https://www.chartjs.org/docs/latest/>)
 - o Librerías de angular también para la realización de gráficos, basada en chart.js

Finalmente, hay que decidir qué tecnologías van a ser las utilizadas para el desarrollo. Inicialmente se hizo una consideración de posibles opciones que se comentarán a continuación para así poder comparar y decidir sobre lo que se puede considerar la opción que más se adapte a nuestro objetivo o a nuestro gusto, que no tiene por qué ser la mejor de todas las posibles combinaciones de tecnologías.

4.1.3.3 Alternativas frontend

Quería hacer una página web que fuera SPA (Single page application), entonces estas tecnologías tuvimos en cuenta

1. Angular: es un framework desarrollado en typescript mantenido por Google. Su objetivo es que el desarrollo y las pruebas sean más fáciles y para ello utiliza el patrón MVC. Utiliza un enfoque basado en componentes y aporta herramientas para trabajar con enrutamiento, validación y más funcionalidades.
2. React: es una biblioteca de JavaScript que está mantenida por Facebook y la comunidad de software libre. Permite crear interfaces de usuario reactivas y dinámicas, es decir, que estén cambiando todo el tiempo. Trata de ser eficiente y fácil de combinar con otro software. Usa el patrón MVC
3. Vue.js: es una biblioteca de javascript que se utiliza para el desarrollo de interfaces de usuario y de aplicaciones SPA. La biblioteca en sí se centra en la capa de vista mientras que, si queremos realizar operaciones de enrutamiento, gestión de estados y demás tenemos que trabajar con librerías y paquetes que están mantenidos con dicha biblioteca.

4.1.3.4 *Alternativas Backend*

1. Java: es un lenguaje de programación orientado a objetos y que tiene un gran soporte para la realización de aplicaciones web, tanto librerías como herramientas por lo que es una gran opción para tener en cuenta.
2. Spring: precisamente es un framework de java para el desarrollo de sitios web, muy útil en caso de optar por este lenguaje ya que nos facilita el trabajo.
3. Node.js: entorno en tiempo de ejecución (RTE) para ejecutar javascript en el lado del servidor, es decir, desarrollar el backend con js. Conocido por permitir crear páginas web de alta velocidad y escalables.
4. PHP: lenguaje de programación adaptado al desarrollo web en el lado del servidor. También permite como node.js realizar sitios de alta velocidad y escalables ya que presenta herramientas y librería para ello.

4.1.3.5 *Alternativas bases de datos*

1. MySQL: sistema de gestión de base de datos relacional, basada en tablas y relaciones entre ellas. Es muy integrable y comúnmente utilizada en aplicaciones web, por lo que es un SGBD muy documentado y con gran cantidad de ejemplos.
2. MongoDB: sistema de gestión de base de datos no relacional, basada en documentos, extremadamente flexible y escalable al no tener unos esquemas estáticos una vez definidos, permitiendo además realizar consultas complejas y a gran velocidad.

4.1.3.6 *Elección final*

Las opciones que se escogieron finalmente son “Angular” para el frontend de la aplicación, “java con spring boot” para el backend y por último una base de datos relacional “MySQL”. También descubrimos un framework de persistencia de datos que reduce mucho el trabajo de acceso a bases de datos en aplicaciones Java. Este framework se llama “MyBatis” y reduce el código necesario, mejora la eficiencia y demás ventajas. Se basa en el patrón ORM y mapea objetos java a consultas SQL.

Para la administración de la base de datos utilizamos “Workbench”, ya que es una interfaz de usuario que nos permite trabajar con bases de datos relacionales MySQL.

Por último, para la extracción de datos de las APIs se ha optado por la opción de Python. Principalmente, esta decisión se debe a que la propia API presenta ejemplos de extracción de datos con este lenguaje y eso facilita el trabajo y ahorra tiempo.

4.1.4 Determinación del Alcance del Sistema

Inicialmente el usuario se encontrará en la pantalla principal del sistema en el que se muestra una introducción con una pequeña explicación de este. A partir de ahí, el cliente podrá navegar a través de un menú superior a diversas secciones.

La primera sección es la de clasificación, en la que se visualizará en un inicio el posicionamiento de los equipos de la liga NBA. Se podrá elegir también la liga ACB para visualizar la clasificación asociada a esta.

La siguiente sección es la de jugadores, donde aparecerán todos los jugadores con los datos generales de los mismos (posición, equipo, altura...). Se podrán filtrar por nombre o por equipo a través de un buscador. Si se presiona sobre alguno de ellos entraremos a una pantalla de detalle en la que se estudian las estadísticas del jugador más a fondo, mostrándose estas a través de tablas y gráficos. Por último, existe una sección de comparación, en la que se puede buscar a cualquier otro jugador por nombre y se desplegará una ventana que comparará las estadísticas de ellos.

Después nos encontramos con la sección de equipos, similar a la de jugadores. En este caso el listado de los equipos permite acceder al detalle de los mismos o a visualizar su plantilla, que contendrá los jugadores pertenecientes al mismo.

Por último, la sección de partidos, donde aparecerán los partidos jugados en la fecha seleccionada a través de un calendario. Si presionamos sobre alguno de ellos se abrirá una ventana que comparará la actuación de ambos equipos a través de gráficos y tablas.

4.2 Requisitos del Sistema

4.2.1 Obtención de los Requisitos del Sistema

4.2.1.1 Requisitos funcionales

Vamos a dividir los requisitos del sistema en dos tablas. En la primera nombraremos los requisitos funcionales para especificar los comportamientos del sistema, en la segunda los no funcionales para establecer restricciones o criterios de calidad.

Código	Nombre Requisito	Descripción del Requisito
R1	Visualizar inicio	Se debe obtener una breve explicación sobre el sistema al entrar en él.
R2	Visualizar clasificación	Se debe poder visualizar la clasificación de la liga seleccionada.
R2.1	Seleccionar liga	Se debe poder escoger entre las ligas 'NBA' y 'ACB' para ver su clasificación
R3	Visualizar jugadores	Se debe obtener un listado de todos los jugadores de la NBA
R3.1	Visualización de jugadores en tarjetas	<p>En el listado de jugadores mencionado en R3 cada jugador aparecerá en una tarjeta que presentará los siguientes datos:</p> <ul style="list-style-type: none"> - Imagen del jugador - Nombre - Peso - Altura - Experiencia - Número - Posición - Equipo
R3.1.1	Controlar los datos	Los datos mostrados en las tarjetas mencionadas en R3.1 deberán tener un valor asignado siempre.
R3.1.2	Presionar tarjeta	Se debe poder presionar sobre el nombre del jugador en la tarjeta para acceder al detalle
R3.1.2.1	Ver detalle de jugador	Se han de visualizar las estadísticas globales de un jugador en la temporada actual, a través de gráficos y tablas.
R3.1.2.1.1	Comparar jugadores	Se debe poder seleccionar dentro del detalle de un jugador a otro jugador para que puedan ser comparados.
R3.1.2.1.1.1	Controlar filtrado comparación jugadores	El nombre introducido en el campo de comparación ha de existir, si no, no se mostrará ningún jugador.
R4	Visualizar equipos	Se debe obtener un listado de todos los equipos de la NBA

R4.1	Visualización de equipos en tarjetas	En el listado de equipos mencionado en R4 cada equipo aparecerá en una tarjeta.
R4.1.1	Acceder al detalle de un equipo	La tarjeta tendrá la opción de acceder al detalle del equipo presionando su nombre
R4.1.1.1	Ver detalle equipo	Se han de visualizar las estadísticas globales de un equipo en la temporada actual, a través de gráficos y tablas.
R4.1.1.1.1	Comparar equipos	Se debe poder seleccionar dentro del detalle de un equipo a otro equipo para que puedan ser comparados.
R4.1.1.1.1.1	Controlar filtrado comparación equipos	El nombre introducido en el campo de comparación ha de existir, si no, no se mostrará ningún equipo.
R4.1.2	Acceder a la plantilla de un equipo	La tarjeta de un equipo tendrá la opción de acceder a la plantilla del mismo.
R4.1.2.1	Formato de vista de la plantilla	La plantilla del equipo seleccionado se visualizará en formato tabla.
R4.1.2.2	Datos de la plantilla	En la tabla de la plantilla mencionada en R4.1.2.1 se visualizarán las estadísticas mencionadas en R3.1 de los jugadores.
R4.1.2.3	Acceder a detalle del jugador desde plantilla	Los jugadores de la lista deben poder ser presionados para acceder a su detalle.
R5	Visualizar partidos	Se podrán visualizar todos los partidos de una fecha seleccionada.
R5.1	Controlar fecha partidos	Si la fecha seleccionada no contiene partidos el sistema le mostrará un mensaje de aviso al usuario.
R5.2	Visualización de partidos en tarjetas	Los partidos se visualizarán en formato tarjeta, mostrando las imágenes de los dos equipos enfrentados.
R5.2.1	Acceder al detalle de un partido	La tarjeta mencionada en R5.2 dará la opción de acceder al detalle del partido.
R5.2.1.1	Ver detalle de un partido	El sistema debe comparar las estadísticas de los dos equipos enfrentados a través de tablas y gráficos.

Tabla 7 Requisitos funcionales

4.2.1.2 Requisitos no funcionales

Código	Nombre Requisito	Descripción del Requisito
RNF1	Conexión	Para usar la aplicación es necesaria una conexión a internet.
RNF2	Usabilidad	La aplicación ha de cumplir con los requisitos de usabilidad realizando pruebas con usuarios para garantizarla.
RNF2.1	Responsivo	La aplicación debe ajustarse a la pantalla de cualquier dispositivo.
RNF3	Rendimiento	El sistema ha de responder de manera rápida a las peticiones del usuario.
RNF4	Escalabilidad	El sistema ha de ser fácilmente extensible
RNF5	Disponibilidad	El sistema ha de estar disponible y actualizado siempre que un usuario lo requiera.
RNF6	Mantenibilidad	El sistema debe estar diseñado para su fácil mantenimiento.

Tabla 8 Requisitos no funcionales

4.2.2 Identificación de Actores del Sistema

El único actor que forma parte del sistema es:

Usuario: todos los usuarios tendrán las mismas funcionalidades, no hay posibilidad de registro. Nos referimos con usuario a aquella persona que puede interactuar con nuestro sistema.



4.2.3 Especificación de Casos de Uso

En este apartado se definirán los casos de uso del proyecto. Además, se le asignará a cada caso un identificador único para hacer la relación con los requisitos y con los escenarios. Esto último se puede ver en [5.6 Relación Escenarios – Casos de Uso – Requisitos](#).

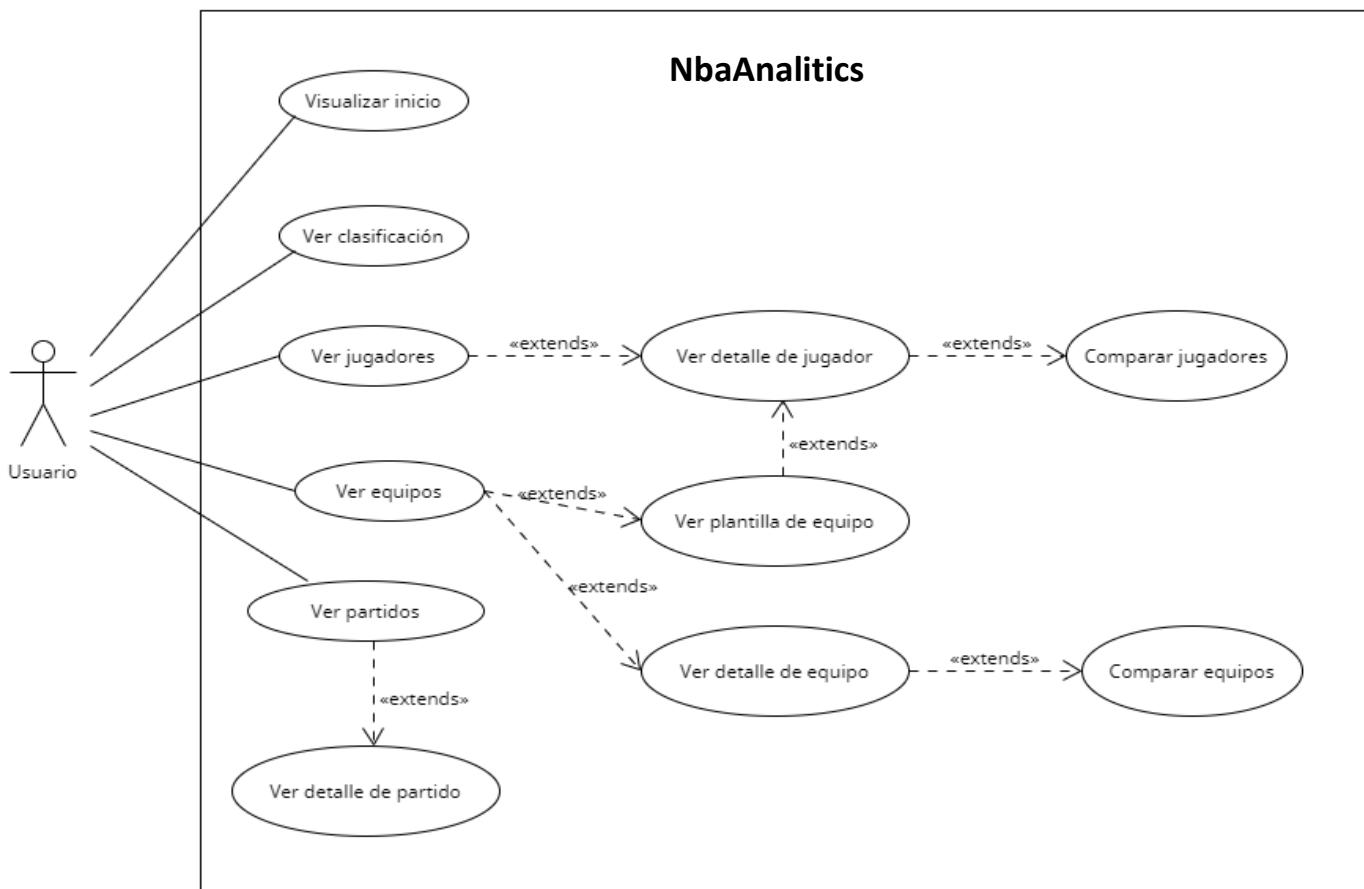


Ilustración 4-2 Diagrama de casos de uso

Nombre del Caso de Uso
Visualizar inicio (CU-VI)
Descripción
El sistema le mostrará una pantalla de inicio al usuario en la que se informará acerca de la aplicación que va a ser utilizada. El usuario podrá volver a esta sección a través del menú siempre que lo desee.

Tabla 9 Caso de uso Ver Inicio

Nombre del Caso de Uso
Ver clasificación (CU-VC)
Descripción
El usuario podrá visualizar la clasificación de la liga que haya seleccionado. Dicha liga la podrá escoger a través de un botón que despliega las distintas opciones existentes

Tabla 10 Caso de uso Ver Clasificación

Nombre del Caso de Uso
Ver jugadores (CU-VJ)
Descripción
El usuario podrá visualizar todos los jugadores de la NBA a través de una opción de menú. El sistema mostrará a todos los jugadores en forma de tarjeta con una serie de características globales del jugador.

Tabla 11 Caso de uso Ver Jugadores

Nombre del Caso de Uso
Ver detalle de jugador (CU-DJ)
Descripción
El usuario podrá seleccionar uno de los jugadores del listado, presionando sobre el nombre de la tarjeta. En ese momento el sistema mostrará una nueva ventana en la que se mostrarán las estadísticas del jugador en la temporada actual, en forma de tablas y gráficas.

Tabla 12 Caso de uso Detalle de jugador

Nombre del Caso de Uso
Comparar jugadores (CU-CJ)
Descripción
El usuario una vez dentro del detalle de un jugador, dispone de la opción de escoger otro jugador para que sean comparados. Esto es posible a través de un botón que permite escribir el nombre del jugador deseado.

Tabla 13 Caso de uso Comparar Jugadores

Nombre del Caso de Uso
Ver equipos (CU-VE)
Descripción
El usuario podrá visualizar todos los equipos de la NBA a través de una opción de menú. El sistema mostrará a todos los equipos en forma de tarjeta permitiendo acceder al detalle del equipo o a la plantilla.

Tabla 14 Caso de uso Ver Equipos

Nombre del Caso de Uso
Ver plantilla de equipo (CU-PE)
Descripción
El usuario podrá visualizar la plantilla de un equipo. Para ello ha de presionar sobre el botón 'plantilla' de la tarjeta del equipo deseado. El sistema mostrará una tabla con todos los jugadores del equipo, pudiendo acceder desde ella al detalle de los jugadores en caso de presionar sobre el nombre de alguno de ellos.

Tabla 15 Caso de uso Plantilla de equipo

Nombre del Caso de Uso
Ver detalle de equipo (CU-DE)
Descripción
El usuario podrá seleccionar uno de los equipos del listado, presionando sobre el nombre de la tarjeta o sobre el botón de estadísticas. En ese momento el sistema mostrará una nueva ventana en la que se mostrarán las estadísticas del equipo en la temporada actual y en los 5 años anteriores, en forma de tablas y gráficas.

Tabla 16 Caso de uso Detalle de Equipo

Nombre del Caso de Uso
Comparar equipos (CU-CE)
Descripción
El usuario una vez dentro del detalle de un equipo, dispone de la opción de escoger otro equipo para que sean comparados. Esto es posible a través de un botón que permite escribir el nombre del equipo deseado.

Tabla 17 Caso de uso Comparar Equipos

Nombre del Caso de Uso
Ver partidos (CU-VP)
Descripción
El usuario podrá visualizar todos los partidos de la NBA disputados en la fecha seleccionada, a través de una opción de menú. El sistema mostrará a todos los partidos en forma de tarjeta permitiendo acceder al detalle.

Tabla 18 Caso de uso Ver Partido

Nombre del Caso de Uso
Ver detalle de partido (CU-DP)
Descripción
El usuario podrá seleccionar uno de los partidos transcurridos en la fecha especificada, presionando sobre un botón de la tarjeta. En ese momento el sistema mostrará una nueva ventana en la que se mostrarán las estadísticas de ambos equipos en el partido, en forma de tablas y gráficas.

Tabla 19 Caso de uso Detalle Partido

4.3 Identificación de los Subsistemas en la Fase de Análisis

Vamos a descomponer la aplicación en subsistemas para desglosar el proyecto en partes más pequeñas y que se pueda comprender de una forma más sencilla. Antes de comenzar con ello, he de mencionar que el proyecto está diseñado con el patrón MVC (Modelo – Vista – Controlador) y que separamos estas funcionalidades en distintos subsistemas que a su vez serán divididos.

4.3.1 Descripción de los Subsistemas

Los principales subsistemas y a su vez los más generales son:

- Frontend: vista de la aplicación, es el subsistema encargado de la interfaz de usuario. Interaccionará directamente con el backend.
- Backend: tiene las clases que funcionan como modelo y controlador, el encargado de la recuperación de la información de la base de datos.
- Base de datos: conjunto de entidades que contienen toda la información de la aplicación. Estos datos son recogidos por el backend.

Otras partes del proyecto, que no se consideran parte del sistema final pero que han intervendrán en su desarrollo son:

- Extracción de datos: documentos que extraen la información haciendo peticiones a las APIs descritas a continuación y que filtra esos datos para que sean almacenados en nuestra base de datos de la manera definida en la misma.
- API NBA: sistema que contiene todos los datos de la liga NBA de baloncesto. Parte de ellos serán extraídos y almacenados en nuestra base de datos.
- RAPIDAPI: sistema que contiene información sobre la liga ACB de baloncesto. Parte de esa información será almacenada en nuestra base de datos.

A continuación, dentro de los sistemas globales frontend, backend y base de datos, se desglosa su estructura en función de los siguientes módulos o subsistemas:

- Módulo clasificación: todas las clases, entidades y ficheros (XML) relacionados con la gestión de la clasificación.
- Módulo jugadores: todas las clases, entidades y ficheros (XML) relacionados con la gestión de la clasificación.
- Módulo equipos: todas las clases, entidades y ficheros (XML) relacionados con la gestión de la clasificación.
- Módulo partidos: todas las clases, entidades y ficheros (XML) relacionados con la gestión de la clasificación.

4.3.2 Descripción de los Interfaces entre Subsistemas

La comunicación entre los subsistemas principales de la aplicación (frontend, backend y base de datos) se hace de forma local, a través de una IP local y conectándose a través del protocolo HTTP al puerto definido en cada subsistema (4200 para el frontend, 8080 para el backend y 3306 para la base de datos).

La comunicación entre los ficheros de extracción de datos con las APIs que contienen los datos es simplemente a través de la realización de peticiones a los endpoints definidos por dichas APIs, a través de la red y del protocolo HTTPS.

La comunicación entre la base de datos y los ficheros de extracción de datos es manual, es decir, será el desarrollador el que se encargue de generar los ficheros CSV con los datos e importarlos en las tablas definidas de la base de datos.

Por último, los módulos más simples de la aplicación (clasificación, jugadores, equipos y partidos) se conectarán en el frontend a través de una clase compartida, en el backend a través de la agregación o composición de los objetos entre clases del mismo módulo (el controlador poseerá el servicio, el servicio el mapper y el modelo de datos) y por último la base de datos contendrá las entidades necesarias para cada módulo con sus correspondientes relaciones.

4.4 Diagrama de Clases Preliminar del Análisis

Vamos a realizar el análisis de las clases que tendrá el sistema y sus relaciones de una forma aproximada, mostrando el diagrama y la especificación de cada una de las clases.



4.4.1 Diagrama de Clases

El diagrama de clases que se va a mostrar en esta sección va a ser una aproximación lo más precisa posible sobre lo que va a ser el sistema final. De todas formas, si se quiere ver la especificación más exacta se puede visualizar el apartado [6.2.1 Diagrama de Clases](#).

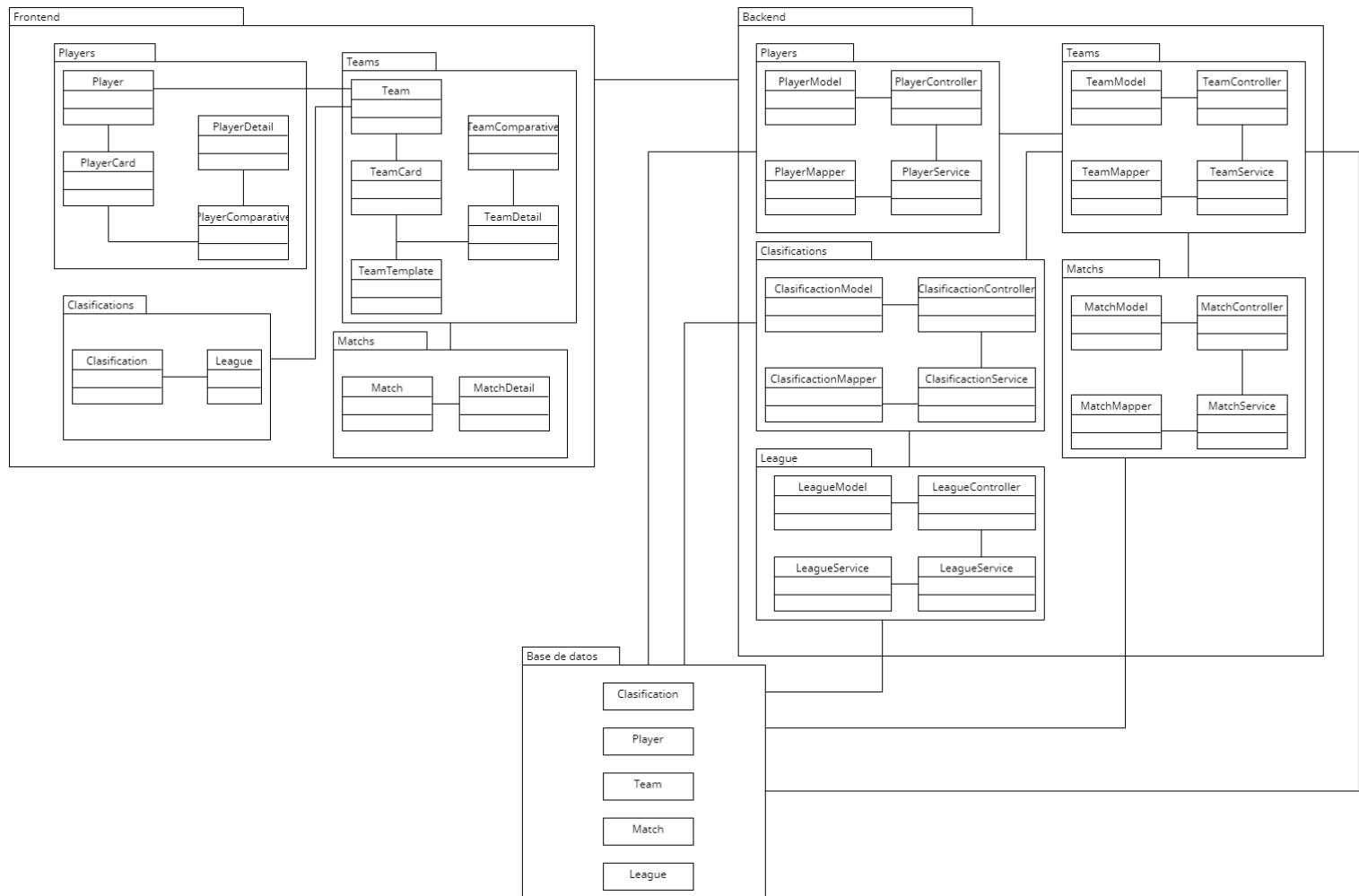


Ilustración 4-3 Diagrama de clases global

4.4.2 Descripción de las Clases

Vamos a hacer un esbozo de lo que serán las clases del sistema, con una aproximación de sus atributos y métodos.

4.4.2.1 Frontend

4.4.2.1.1 Jugadores

<u>Nombre de la Clase</u>
Player
Descripción
Será el modelo del jugador, aquel que contenga exclusivamente los datos que serán utilizados
Responsabilidades
Contener los datos
Atributos Propuestos
PlayerId: Identificador del jugador
TeamId: Identificador del equipo del jugador
Name: Nombre del jugador
Age: Edad del jugador
Position: Posición del jugador
Weight: Peso del jugador
Height: Altura del jugador
AllStats: Nos referimos con esto al conjunto de atributos que se corresponderán con cada estadística del jugador.
Métodos Propuestos
Getters/Setters: Para poder acceder y modificar si es necesario el valor de los atributos

Tabla 20 Clase Player

<u>Nombre de la Clase</u>
Players
Descripción
Será la clase que gestione todos los jugadores existentes
Responsabilidades
Mostrar las tarjetas de todos los jugadores
Atributos Propuestos
Players: Datos de todos los jugadores
Métodos Propuestos
Ninguno, solo hacer petición y maquetar tarjetas

Tabla 21 Clase Players

<u>Nombre de la Clase</u>
PlayerCard
Descripción
Clase que se encarga de maquetar en una tarjeta los datos globales del jugador
Responsabilidades
Maquetar visualmente la información general del jugador y dar acceso al detalle de este
Atributos Propuestos
Player: Objeto que representa al jugador de la tarjeta
Métodos Propuestos
setPlayerSelected: Objeto que establece el jugador de la tarjeta como seleccionado en la clase compartida para que se pueda coger su información desde el detalle

Tabla 22 Clase PlayerCard

<u>Nombre de la Clase</u>	
PlayerDetail	
Descripción	Clase que contendrá todo el tratamiento de las estadísticas del jugador
Responsabilidades	Calcular estadísticas y mostrarlas en tablas y gráficos
Atributos Propuestos	
Player: Objeto que representa al jugador del detalle	
AllPlayers: Resto de jugadores con los que puede compararse	
Chart: un objeto de este tipo por cada gráfico, referencia al gráfico	
Métodos Propuestos	
CalculatePlayerPerformance: Método que calcula el rendimiento del jugador	
InitializeGraphs: Método que crea los gráficos	

Tabla 23 Clase PlayerDetail

<u>Nombre de la Clase</u>	
PlayerComparative	
Descripción	Clase que se encarga de la comparación de dos jugadores
Responsabilidades	Mostrar estadísticas comparativas de ambos jugadores en tablas y gráficos
Atributos Propuestos	
PlayerDetail: Objeto que representa al jugador del detalle	
PlayerToCompare: Objeto que representa al jugador escogido para comparar	
Chart: un objeto de este tipo por cada gráfico, referencia al gráfico	
Métodos Propuestos	
InitializeGraphs: Método que crea los gráficos	

Tabla 24 Clase PlayerComparative

4.4.2.1.2 Equipos

<u>Nombre de la Clase</u>	
Team	
Descripción	Será el modelo del equipo, aquel que contenga exclusivamente los datos que serán utilizados
Responsabilidades	Contener los datos
Atributos Propuestos	
TeamId: Identificador del equipo	
Name: Nombre del equipo	
Year: Año concreto de las estadísticas almacenadas	
Conference: Conferencia del equipo	
Division: División del equipo	
NBAFinalAppearance: Aparición del equipo en la final	
AllStats: Estadísticas del equipo	
Métodos Propuestos	
Getters/Setters: Para poder acceder y modificar si es necesario el valor de los atributos	

Tabla 25 Clase Team

<u>Nombre de la Clase</u>
Teams
Descripción
Será la clase que gestione todos los equipos existentes
Responsabilidades
Mostrar las tarjetas de todos los equipos
Atributos Propuestos
Teams: Datos de todos los equipos
Métodos Propuestos
Ninguno, solo hacer petición y maquetar tarjetas

Tabla 26 Clase Teams

<u>Nombre de la Clase</u>
TeamCard
Descripción
Clase que se encarga de maquetar en una tarjeta los datos globales del equipo
Responsabilidades
Maquetar visualmente la información general del equipo y dar acceso al detalle de este
Atributos Propuestos
Team: Objeto que representa al equipo de la tarjeta
Métodos Propuestos
setTeamSelected: Objeto que establece el equipo de la tarjeta como seleccionado en la clase compartida para que se pueda coger su información desde el detalle

Tabla 27 Clase TeamCard

<u>Nombre de la Clase</u>
TeamDetail
Descripción
Clase que contendrá todo el tratamiento de las estadísticas del equipo
Responsabilidades
Obtener estadísticas y mostrarlas en tablas y gráficos
Atributos Propuestos
TeamByYear: Objeto que representa las estadísticas del equipo en diferentes años
AllTeams: Resto de equipos con los que puede compararse
Chart: un objeto de este tipo por cada gráfico, referencia al gráfico
Métodos Propuestos
InitializeGraphs: Método que crea los gráficos

Tabla 28 Clase TeamDetail

Nombre de la Clase	
TeamTemplate	
Descripción	
Clase que mostrará la plantilla del equipo seleccionado	
Responsabilidades	
Mostrar los jugadores pertenecientes a un equipo	
Atributos Propuestos	
Team: Objeto que representa el equipo seleccionado	
Players: Jugadores pertenecientes al equipo seleccionado	
Métodos Propuestos	
SetPlayerSelectedDetail: Método que establece el jugador seleccionado desde la plantilla en la clase compartida para poder acceder a su detalle	

Tabla 29 Clase TeamTemplate

Nombre de la Clase	
TeamComparative	
Descripción	
Clase que se encarga de la comparación de dos equipos	
Responsabilidades	
Mostrar estadísticas comparativas de ambos equipos en tablas y gráficos	
Atributos Propuestos	
TeamDetail: Objeto que representa al equipo del detalle	
TeamToCompare: Objeto que representa al equipo escogido para comparar	
Chart: un objeto de este tipo por cada gráfico, referencia al gráfico	
Métodos Propuestos	
InitializeGraphs: Método que crea los gráficos	

Tabla 30 Clase TeamComparative

4.4.2.1.3 Clasificación

Nombre de la Clase	
Clasification	
Descripción	
Clase que gestiona la clasificación de una liga	
Responsabilidades	
Mostrar la clasificación de la liga seleccionada	
Atributos Propuestos	
Clasification: Objeto que contiene el conjunto de filas de la tabla de la clasificación	
LeagueSelected: Objeto que representa la liga seleccionada	
Métodos Propuestos	
ChangeLeague: Método que cambia la liga seleccionada	

Tabla 31 Clase Clasification

4.4.2.1.4 Partidos

Nombre de la Clase	
Match	
Descripción	
Será el modelo del partido, aquel que contenga exclusivamente los datos que serán utilizados	
Responsabilidades	
Contener los datos	
Atributos Propuestos	
GameId: Identificador del equipo	
TeamStats: Estadísticas del equipo en el partido	
Métodos Propuestos	
Getters/Setters: Para poder acceder y modificar si es necesario el valor de los atributos	

Tabla 32 Clase Match

Nombre de la Clase	
Matchs	
Descripción	
Será la clase que gestione todos los partidos existentes en una fecha	
Responsabilidades	
Mostrar las tarjetas de todos los partidos	
Atributos Propuestos	
GameStats: Datos de todos los equipos en el partido	
Métodos Propuestos	
SelectDate: Mandar la fecha seleccionada a la clase compartida para poder hacer la petición	
SetMatchSelectedDetail: Establecer el partido seleccionado en la clase compartida para poder acceder a él desde el detalle	

Tabla 33 Clase Matchs

Nombre de la Clase	
MatchDetail	
Descripción	
Clase que contendrá todo el tratamiento de las estadísticas del partido	
Responsabilidades	
Obtener estadísticas y mostrarlas en tablas y gráficos	
Atributos Propuestos	
Chart: un objeto de este tipo por cada gráfico, referencia al gráfico	
Métodos Propuestos	
InitializeGraphs: Método que crea los gráficos	

Tabla 34 Clase MatchDetail

4.4.2.2 Backend

El backend es dividido en paquetes según los módulos existentes en el proyecto, es decir, se forma un paquete por:

- Jugadores
- Equipos

- Clasificación
- Partidos
- Liga

Dentro de cada paquete vamos a tener estos subpaquetes:

- Model
- Controller
- Service
- Mapper

Además de tener asociado un recurso XML que es el encargado de realizar las peticiones a la base de datos.

Por tanto, a continuación, vamos a describir el módulo de jugadores y del resto tan solo el controlador para ver que métodos contiene cada módulo, pero el resto de los paquetes es aplicable para todos de la misma forma.

4.4.2.2.1 Jugadores

Nombre de la Clase
PlayerModel
Descripción
Clase que contiene los atributos del jugador
Responsabilidades
Guardar la información recuperada de la base de datos
Atributos Propuestos
Mismos atributos que los que contiene la entidad Player en la base de datos
Métodos Propuestos
Getters/Setter: Métodos para poder acceder y modificar si es necesario sus atributos

Tabla 35 Clase PlayerModel

Nombre de la Clase	
PlayerController	
Descripción	
Clase que contiene los endpoint de los jugadores	
Responsabilidades	
Definir los endpoint del backend y los métodos que van a realizar las peticiones	
Atributos Propuestos	
PlayerService: Referencia al servicio del jugador	
Métodos Propuestos	
getPlayersStats: Método que hace la llamada al servicio para obtener las estadísticas de todos los jugadores	
getPlayerStatsById: Método que hace la llamada al servicio para obtener las estadísticas del jugador especificado	
getPlayersInfo: Método que hace la llamada al servicio para obtener las estadísticas globales del jugador	
getPlayersInfoByTeamId: Método que hace la llamada al servicio para obtener las estadísticas globales de los jugadores del equipo especificado	

Tabla 36 Clase PlayerController

Nombre de la Clase	
PlayerService	
Descripción	
Clase que contiene los métodos para recuperar la información solicitada desde el controller	
Responsabilidades	
Obtener la información del mapper y devolversela al controller	
Atributos Propuestos	
PlayerMapper: Referencia al mapeador del jugador	
Métodos Propuestos	
findAllPlayersStats: Método que hace la llamada al mapper para obtener las estadísticas de todos los jugadores	
findAllPlayersInfo: Método que hace la llamada al mapper para obtener las estadísticas del jugador especificado	
findPlayerStatsById: Método que hace la llamada al mapper para obtener las estadísticas globales del jugador	
findPlayersInfoByTeamId: Método que hace la llamada al mapper para obtener las estadísticas globales de los jugadores del equipo especificado	

Tabla 37 Clase PlayerService

Nombre de la Clase	PlayerMapper
Descripción	Interfaz con los mismos métodos que el servicio
Responsabilidades	Desde el fichero XML se referencian los métodos de la interfaz y se hacen las consultas SQL a la base de datos. Esta información es devuelta al service para que se la envíe al controller
Métodos Propuestos	<p>findAllPlayersStats: Método que se implementa en el XML a través de la consulta necesaria para obtener información concreta de la base de datos</p> <p>findAllPlayersInfo: Método que se implementa en el XML a través de la consulta necesaria para obtener información concreta de la base de datos</p> <p>findPlayerStatsById: Método que se implementa en el XML a través de la consulta necesaria para obtener información concreta de la base de datos</p> <p>findPlayersInfoByTeamId: Método que se implementa en el XML a través de la consulta necesaria para obtener información concreta de la base de datos</p>

Tabla 38 Clase PlayerMapper

4.4.2.2.2 Equipos

Nombre de la Clase	TeamController
Descripción	Clase que contiene los endpoint de los equipos
Responsabilidades	Definir los endpoint del backend y los métodos que van a realizar las peticiones
Atributos Propuestos	
TeamService:	Referencia al servicio del equipo
Métodos Propuestos	<p>getTeamsStats: Método que hace la llamada al servicio para obtener las estadísticas globales de todos los equipos</p> <p>getTeamsStatsLastFiveYears: Método que hace la llamada al servicio para obtener las estadísticas en los últimos 5 años de todos los equipos</p>

Tabla 39 Clase TeamController

4.4.2.2.3 Clasificación

<u>Nombre de la Clase</u>
ClasificationController
Descripción
Clase que contiene los endpoint de la clasificación
Responsabilidades
Definir los endpoint del backend y los métodos que van a realizar las peticiones
Atributos Propuestos
ClasificationService: Referencia al servicio de la clasificación
Métodos Propuestos
getClasification: Método que hace la llamada al servicio para obtener la clasificación de la liga seleccionada

Tabla 40 Clase ClasificationController

4.4.2.2.4 Partidos

<u>Nombre de la Clase</u>
MatchController
Descripción
Clase que contiene los endpoint de los partidos
Responsabilidades
Definir los endpoint del backend y los métodos que van a realizar las peticiones
Atributos Propuestos
MatchService: Referencia al servicio del partido
Métodos Propuestos
getGamesStats: Método que hace la llamada al servicio para obtener las estadísticas de todos los partidos
getGamesStatsByDate: Método que hace la llamada al servicio para obtener las estadísticas de todos los partidos en una fecha concreta

Tabla 41 Clase MatchController

4.4.2.2.5 Ligas

<u>Nombre de la Clase</u>
LeagueController
Descripción
Clase que contiene los endpoint de las ligas
Responsabilidades
Definir los endpoint del backend y los métodos que van a realizar las peticiones
Atributos Propuestos
LeagueService: Referencia al servicio de la liga
Métodos Propuestos
getLeague: Método que hace la llamada al servicio para obtener la información de la liga especificada

Tabla 42 Clase LeagueController

4.4.2.3 Base de Datos

Recordamos que nos encontramos en la fase de análisis y por tanto esto es tan solo una aproximación. La base de datos se encuentra detallada en [6.5 Diseño de la Base de Datos](#).

Nombre de la Entidad
Clasification
Descripción
Entidad de la base de datos que contendrá toda la información relacionada con las clasificaciones
Atributos Propuestos
<ul style="list-style-type: none"> • Id • TeamId • LeaguelId • Conference • Played • Wons • Lost
Relaciones
Se relacionará con las entidades 'Team' y 'League'

Tabla 43 Entidad Clasification

Nombre de la Entidad
League
Descripción
Entidad de la base de datos que contendrá toda la información relacionada con las ligas
Atributos Propuestos
<ul style="list-style-type: none"> • LeagueName • Conferences
Relaciones
Se relacionará con la entidad 'Clasification'

Tabla 44 Entidad League

Nombre de la Entidad	
Player	
Descripción	Entidad de la base de datos que contendrá toda la información relacionada con los jugadores
Atributos Propuestos	<ul style="list-style-type: none"> • PlayerId • Name • Age • Position • Height • Weight • AllStats • TeamId
Relaciones	Se relacionará con la entidad 'Team'

Tabla 45 Entidad Player

Nombre de la Entidad	
Team	
Descripción	Entidad de la base de datos que contendrá toda la información relacionada con los equipos
Atributos Propuestos	<ul style="list-style-type: none"> • TeamId • Name • Year • Conference • Division • NBAFinalAppearance • AllStats
Relaciones	Se relacionará con las entidades 'Player', 'Clasification' y 'Match'

Tabla 46 Entidad Team

Nombre de la Entidad	
Match	
Descripción	Entidad de la base de datos que contendrá toda la información relacionada con los jugadores
Atributos Propuestos	<ul style="list-style-type: none"> • Gameld • Matchup • Date • AllStats
Relaciones	Se relacionará con la entidad de 'Team'

Tabla 47 Entidad Match

4.5 Análisis de Casos de Uso y Escenarios

En esta sección vamos a describir los escenarios que pueden surgir de los casos de uso especificados en [5.2.3 Especificación de Casos de Uso](#). Le asignaremos a cada escenario un identificador único para hacer la relación con los casos de uso correspondientes en [5.6 Relación Escenarios – Casos de Uso – Requisitos](#).

4.5.1 Visualizar Inicio

Visualizar Inicio (E-VI)	
Precondiciones	No
Poscondiciones	El sistema resaltará en el menú de navegación la sección 'Inicio' de otro color. Se mostrará el contenido de dicha ventana.
Actores	Usuario
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Accederá a la aplicación
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Escenario Alternativo 1: El usuario se encuentra en alguna otra sección de la aplicación. El usuario: <ol style="list-style-type: none"> 1. Presionará el botón del menú con texto 'Inicio'

Tabla 48 Escenario Visualizar Inicio

4.5.2 Ver Clasificación

Visualizar Clasificación (E-VC)	
Precondiciones	No
Poscondiciones	El sistema resaltará en el menú de navegación la sección 'clasificación' de otro color. Se mostrará el contenido de dicha ventana.
Actores	Usuario
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Accederá a la aplicación 2. Pulsará sobre la opción 'clasificación' del menú de navegación
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Escenario Alternativo 1: La base de datos no contiene datos para la clasificación de la liga seleccionada. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no se contienen datos para la liga seleccionada. • Escenario Alternativo 2: El backend que responde la petición no tiene disponibilidad. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no existen datos. • Escenario Alternativo 3: Cambiar la liga de la que ver la clasificación. <ul style="list-style-type: none"> ○ Se mostrará la clasificación de la liga seleccionada

Tabla 49 Escenario Visualizar Clasificación

4.5.3 Ver Jugadores

Visualizar Jugadores (E-VJ)	
Precondiciones	No
Poscondiciones	El sistema resaltará en el menú de navegación la sección 'jugadores' de otro color. Se mostrarán todos los jugadores de la NBA.
Actores	Usuario
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Accederá a la aplicación 2. Pulsará sobre la opción 'jugadores' del menú de navegación
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Escenario Alternativo 1: La base de datos no contiene datos sobre los jugadores. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no se contienen datos. • Escenario Alternativo 2: El backend que responde la petición no tiene disponibilidad. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no existen datos.

Tabla 50 Escenario Visualizar Jugadores

4.5.4 Filtrar jugadores

Filtrar Jugadores (E-FJ)	
Precondiciones	No
Poscondiciones	El sistema mostrará tan solo los jugadores que cumplen con el filtro establecido.
Actores	Usuario
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Accederá a la aplicación 2. Pulsará sobre la opción 'jugadores' del menú de navegación 3. Escribirá en el buscador de la parte superior izquierda el nombre del jugador o por abreviatura de equipo
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Escenario Alternativo 1: Se filtran los jugadores por nombre no existente. <ul style="list-style-type: none"> ○ El sistema muestra un mensaje diciendo que no se encuentran coincidencias • Escenario Alternativo 2: Se filtran los jugadores por abreviación de equipo no existente. <ul style="list-style-type: none"> ○ El sistema muestra un mensaje diciendo que no se encuentran coincidencias

Tabla 51 Escenario Filtrar Jugadores

4.5.5 Ver detalle de jugador

Ver detalle de jugador (E-DJ)	
Precondiciones	No
Poscondiciones	El sistema mostrará el detalle del jugador seleccionado en una pantalla nueva.
Actores	Usuario
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Accederá a la aplicación 2. Pulsará sobre la opción 'jugadores' del menú de navegación 3. Presionará sobre el nombre de un jugador en su tarjeta
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Escenario Alternativo 1: La base de datos no contiene datos sobre el jugador. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no se contienen datos. • Escenario Alternativo 2: El backend que responde la petición no tiene disponibilidad. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no existen datos.

Tabla 52 Escenario Detalle de Jugador

4.5.6 Comparar jugadores

Comparar jugadores (E-CJ)	
Precondiciones	No
Poscondiciones	El sistema mostrará un diálogo con la comparación entre los dos jugadores escogidos.
Actores	Usuario
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Accederá a la aplicación 2. Pulsará sobre la opción 'jugadores' del menú de navegación 3. Presionará sobre el nombre de un jugador en su tarjeta 4. Escribirá en el botón 'comparar' el nombre del jugador deseado 5. Presionara sobre el botón del jugador buscado
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Escenario Alternativo 1: La base de datos no contiene datos sobre alguno de los jugadores. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no se contienen datos. • Escenario Alternativo 2: El backend que responde la petición no tiene disponibilidad. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no existen datos. • Escenario Alternativo 3: la búsqueda del jugador por el nombre del mismo no presenta coincidencias. <ul style="list-style-type: none"> ○ El sistema muestra un mensaje diciendo que no se encuentran coincidencias

Tabla 53 Escenario Comparar Jugadores

4.5.7 Ver equipos

Visualizar equipos (E-VE)	
Precondiciones	No
Poscondiciones	El sistema resaltará en el menú de navegación la sección 'equipos' de otro color. Se mostrarán todos los equipos de la NBA.
Actores	Usuario
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Accederá a la aplicación 2. Pulsará sobre la opción 'equipos' del menú de navegación
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Escenario Alternativo 1: La base de datos no contiene datos sobre los equipos. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no se contienen datos. • Escenario Alternativo 2: El backend que responde la petición no tiene disponibilidad. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no existen datos.

Tabla 54 Escenario Visualizar Equipos

4.5.8 Filtrar equipos

Filtrar Equipos (E-FE)	
Precondiciones	No
Poscondiciones	El sistema mostrará tan solo los equipos que cumplen con el filtro establecido.
Actores	Usuario
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Accederá a la aplicación 2. Pulsará sobre la opción 'equipos' del menú de navegación 3. Escribirá en el buscador de la parte superior izquierda el nombre del equipo.
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Escenario Alternativo 1: Se filtran los equipos por nombre no existente. <ul style="list-style-type: none"> ○ El sistema muestra un mensaje diciendo que no se encuentran coincidencias

Tabla 55 Escenario Filtrar Equipos

4.5.9 Ver detalle de equipo

Ver detalle de equipo (E-DE)	
Precondiciones	No
Poscondiciones	El sistema mostrará el detalle del equipo seleccionado en una pantalla nueva.
Actores	Usuario
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Accederá a la aplicación 2. Pulsará sobre la opción 'equipos' del menú de navegación 3. Presionará sobre el nombre de un equipo en su tarjeta o sobre el botón 'estadísticas'
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Escenario Alternativo 1: La base de datos no contiene datos sobre el equipo. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no se contienen datos. • Escenario Alternativo 2: El backend que responde la petición no tiene disponibilidad. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no existen datos.

Tabla 56 Escenario Detalle Equipo

4.5.10 Comparar equipos

Comparar equipos (E-CE)	
Precondiciones	No
Poscondiciones	El sistema mostrará un diálogo con la comparación entre los dos equipos escogidos.
Actores	Usuario
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Accederá a la aplicación 2. Pulsará sobre la opción 'equipos' del menú de navegación 3. Presionará sobre el nombre de un equipo en su tarjeta 4. Escribirá en el botón 'comparar' el nombre del equipo deseado 5. Presionara sobre el botón del equipo buscado
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Escenario Alternativo 1: La base de datos no contiene datos sobre alguno de los equipos. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no se contienen datos. • Escenario Alternativo 2: El backend que responde la petición no tiene disponibilidad. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no existen datos. • Escenario Alternativo 3: la búsqueda del equipo por el nombre de este no presenta coincidencias. <ul style="list-style-type: none"> ○ El sistema muestra un mensaje diciendo que no se encuentran coincidencias

--	--

Tabla 57 Escenario Comparar Equipos

4.5.11 Ver plantilla de equipo

Ver plantilla de equipo (E-PE)	
Precondiciones	No
Poscondiciones	El sistema mostrará un listado en formato tabla con los jugadores del equipo seleccionado.
Actores	Usuario
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Accederá a la aplicación 2. Pulsará sobre la opción 'equipos' del menú de navegación 3. Presionará sobre el botón 'plantilla' de la tarjeta de un equipo
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Escenario Alternativo 1: La base de datos no contiene datos sobre los jugadores de un equipo. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no se contienen datos. • Escenario Alternativo 2: El backend que responde la petición no tiene disponibilidad. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no existen datos. • Escenario Alternativo 3: no existen partidos en la fecha seleccionada <ul style="list-style-type: none"> ○ Se muestra un mensaje de que no existen partidos en la fecha seleccionada

Tabla 58 Escenario Plantilla Equipo

4.5.12 Ver partidos

Visualizar partidos (E-VP)	
Precondiciones	No
Poscondiciones	El sistema resaltará en el menú de navegación la sección 'partidos' de otro color. Se mostrarán todos los partidos de la fecha seleccionada.
Actores	Usuario
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 4. Accederá a la aplicación 5. Pulsará sobre la opción 'partidos' del menú de navegación 6. Seleccionará la fecha deseada en el calendario
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Escenario Alternativo 1: La base de datos no contiene datos sobre los partidos. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no se contienen datos. • Escenario Alternativo 2: El backend que responde la petición no tiene disponibilidad.

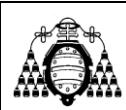
	<ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no existen datos. ● Escenario Alternativo 3: no existen partidos en la fecha seleccionada <ul style="list-style-type: none"> ○ Se muestra un mensaje de que no existen partidos en la fecha seleccionada
--	--

Tabla 59 Escenario Ver Partidos

4.5.13 Ver detalle de partidos

Ver detalle de partido (E-DP)	
Precondiciones	No
Poscondiciones	El sistema mostrará el detalle del partido seleccionado en una pantalla nueva.
Actores	Usuario
Descripción	<p>El usuario:</p> <ol style="list-style-type: none"> 1. Accederá a la aplicación 2. Pulsará sobre la opción 'partidos' del menú de navegación 3. Seleccionará la fecha deseada en el calendario 4. Presionará sobre el botón 'estadísticas' de la tarjeta de un partido.
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> ● Escenario Alternativo 1: La base de datos no contiene datos sobre el equipo. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no se contienen datos. ● Escenario Alternativo 2: El backend que responde la petición no tiene disponibilidad. <ul style="list-style-type: none"> ○ Se mostrará un mensaje de que no existen datos.

Tabla 60 Escenario Detalle Partido

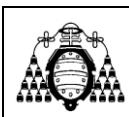


4.6 Relación Escenarios – Casos de Uso – Requisitos

Esta sección consiste en relacionar todos los casos de uso especificados en [5.2.3 Especificación de Casos de Uso](#) con los requisitos establecidos, como se puede observar en *Tabla 61 Relación Casos de uso / Requisitos*, en [5.2.1.1 Requisitos funcionales](#) y con los escenarios generados en [5.5 Análisis de Casos de Uso y Escenarios](#) como muestra en *Tabla 62 Relación Casos de uso / Escenarios*.

Casos de Uso:	CU-VI	CU-VC	CU-VJ	CU-DJ	CU-CJ	CU-VE	CU-DE	CU-CE	CU-PE	CU-VP	CU-DP
R1	X										
R2		X									
R2.1		X									
R3			X								
R3.1			X								
R3.1.1			X								
R3.1.2			X								
R3.1.2.1				X							
R3.1.2.1.1					X						
R3.1.2.1.1.1						X					
1											
R4						X					
R4.1						X					
R4.1.1						X					
R4.1.1.1							X				
R4.1.1.1.1								X			
1								X			
R4.1.2									X		
R4.1.2.1									X		
R4.1.2.2									X		
R4.1.2.3									X		
R5										X	
R5.1										X	
R5.2										X	
R5.2.1										X	
R5.2.1.1											X

Tabla 61 Relación Casos de uso / Requisitos



Casos de Uso:	CU-VI	CU-VC	CU-VJ	CU-DJ	CU-CJ	CU-VE	CU-DE	CU-CE	CU-PE	CU-VP	CU-DP
Escenario E-VI	X										
Escenario E-VC		X									
Escenario E-VJ			X								
Escenario E-FJ				X							
Escenario E-DJ					X						
Escenario E-CJ						X					
Escenario E-VE							X				
Escenario E-FE							X				
Escenario E-DE								X			
Escenario E-CE									X		
Escenario E-PE										X	
Escenario E-VP											X
Escenario E-DP											X

Tabla 62 Relación Casos de uso / Escenarios

4.7 Análisis de Interfaces de Usuario

4.7.1 Descripción de la Interfaz

En este apartado se van a definir los prototipos de la aplicación para así tener una idea inicial de lo que va a resultar finalmente el proyecto en cuanto a interfaz de usuario, teniendo en cuenta que estos prototipos son bocetos iniciales y que pueden sufrir cambios. De todas formas, se intentará que estos cambios sean mínimos.

Para comenzar vamos a definir el prototipo de la pantalla de inicio, en la que se visualizará el menú de navegación (como en todas las pantallas, así como el header y el footer), un cuadro informativo de la aplicación y una serie de imágenes de baloncesto a su alrededor.

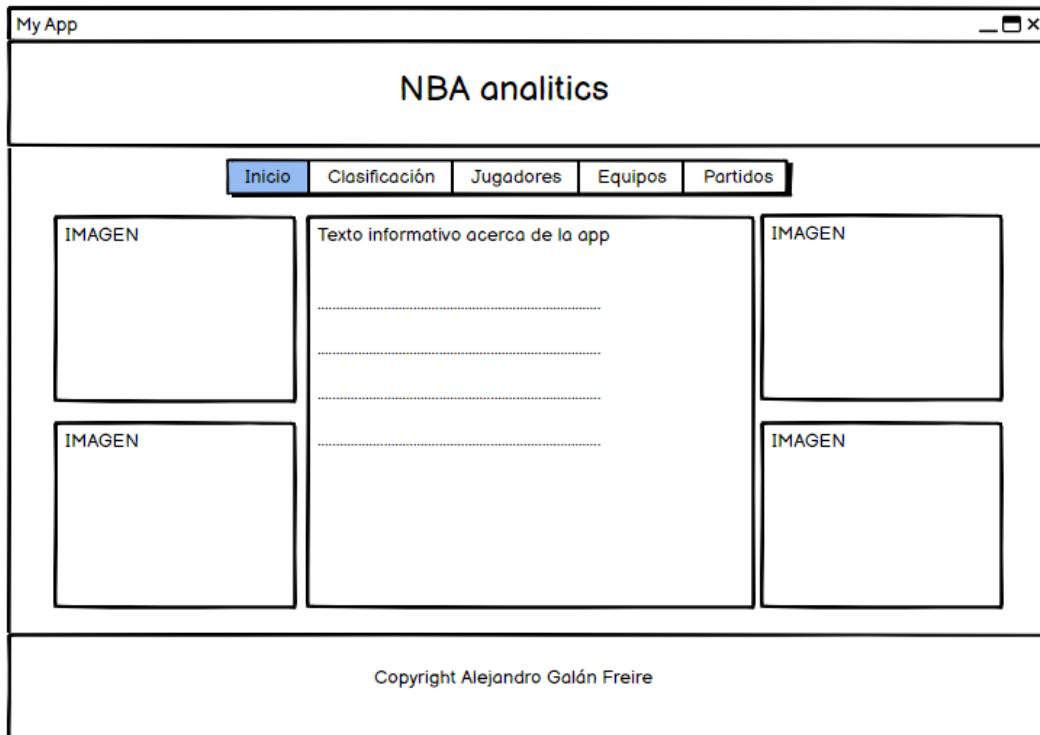


Ilustración 4-4 Prototipo pantalla de inicio

A continuación, vamos a definir el boceto de la pantalla de clasificación, que contendrá un botón para seleccionar la liga de la que se quiere ver la clasificación y una tabla con los equipos en su correspondiente posición.

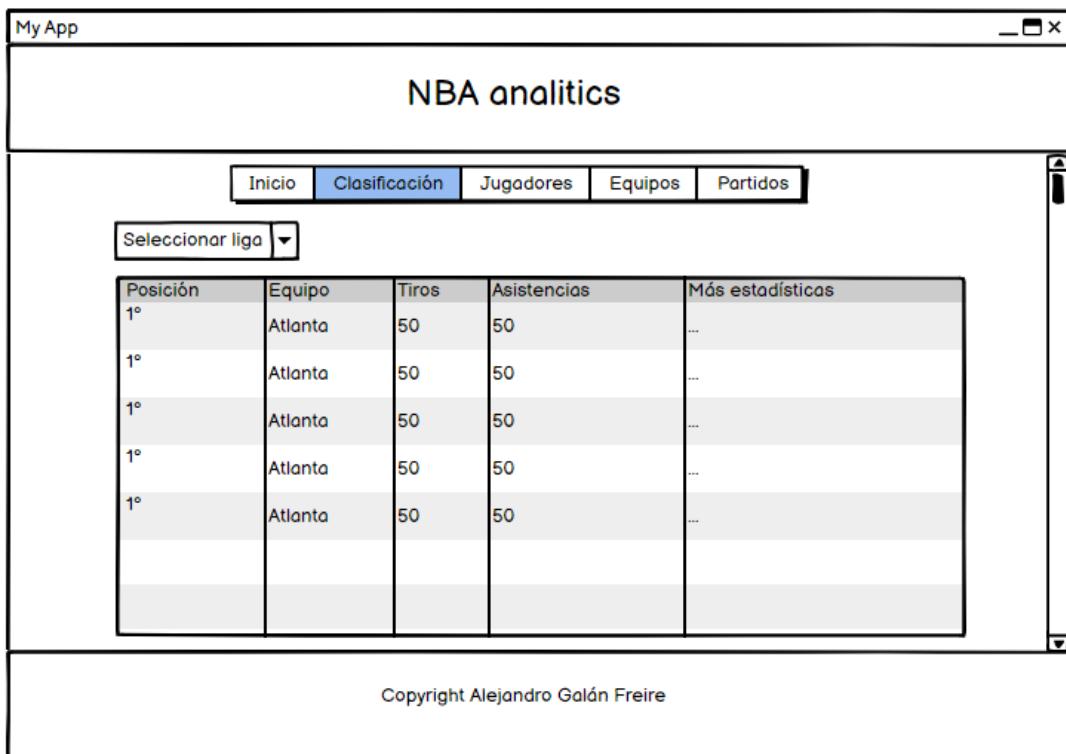
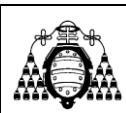


Ilustración 4-5 Prototipo pantalla clasificación



Pasamos ahora a definir la pantalla de visualización de jugadores. En ella aparecerán todos los jugadores en tarjetas, con sus datos más generales, y dando la posibilidad de seleccionarlas para entrar en el detalle. Dispondrá además de un buscador para poder encontrar más fácilmente a un jugador o conjunto de jugadores a través de su nombre o del equipo al que pertenecen.

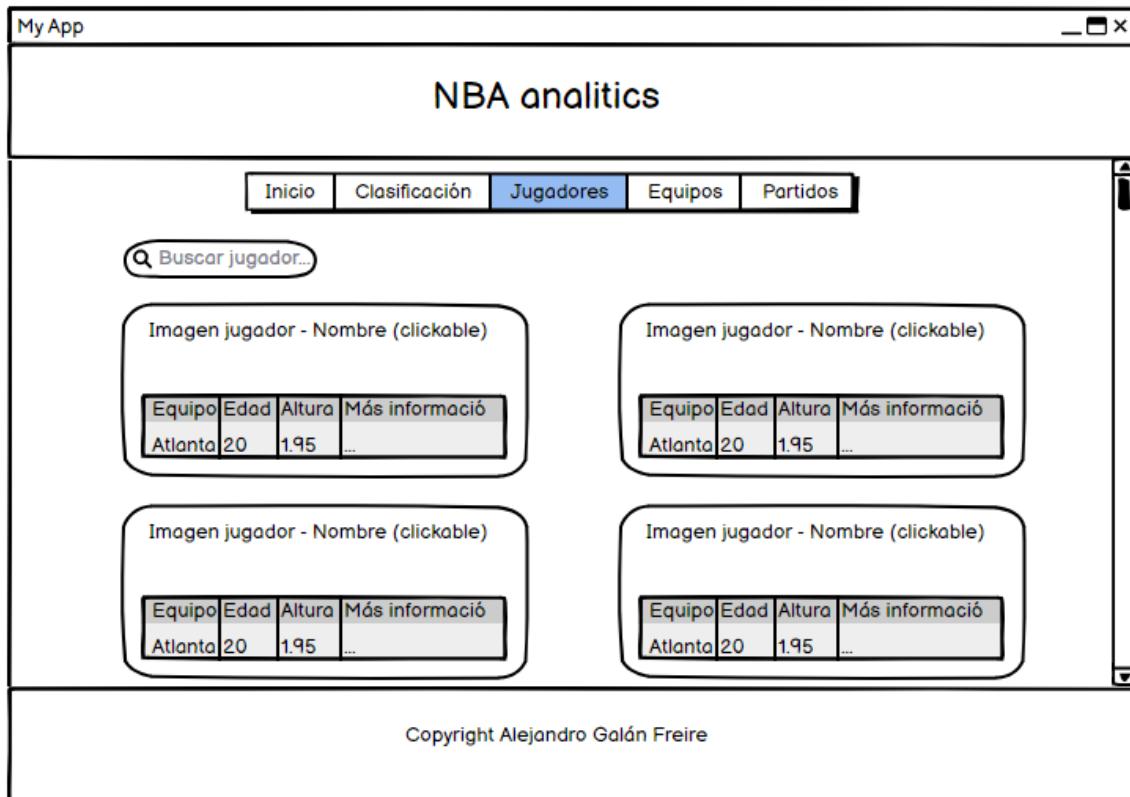


Ilustración 4-6 Prototipo pantalla de jugadores

A continuación, vamos a definir las pantallas de visualización de equipos y partidos, que siguen la misma estructura que la de jugadores. Las únicas diferencias es la información mostrada en las tarjetas y que en la pantalla de partidos habrá un calendario para escoger la fecha de la que se quieren ver los partidos en vez de un buscador.

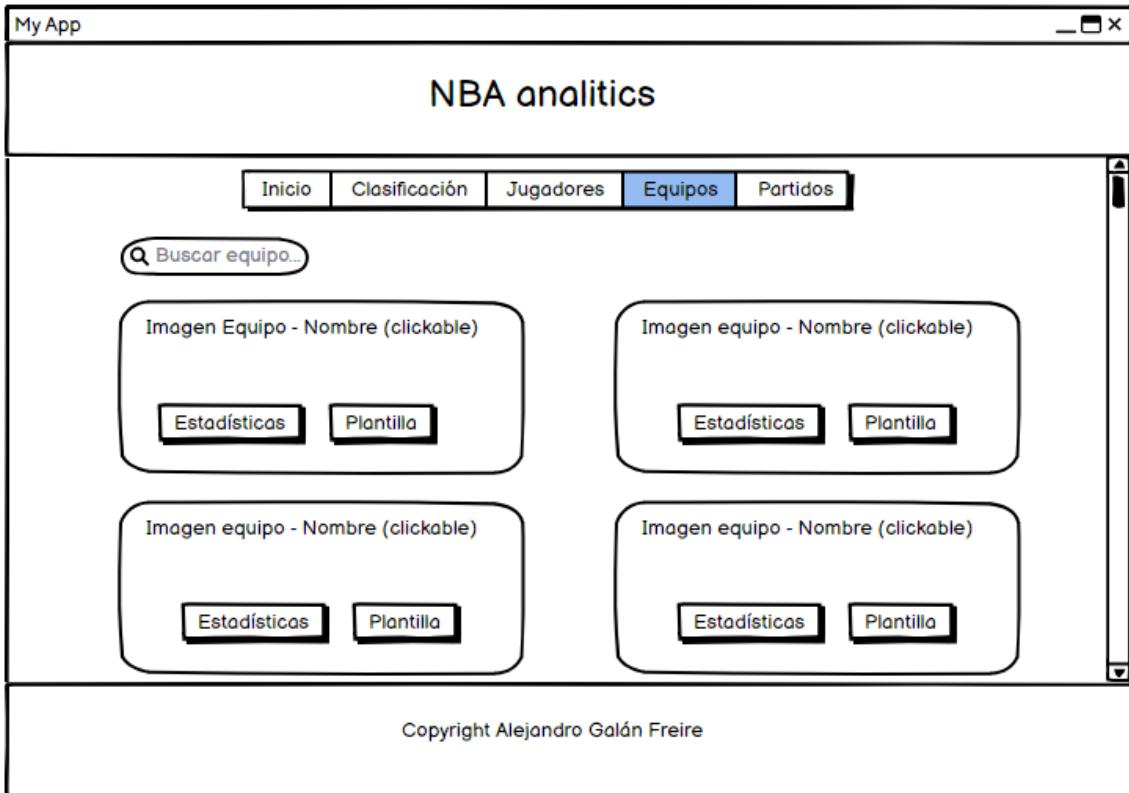


Ilustración 4-7 Prototipo pantalla de equipos

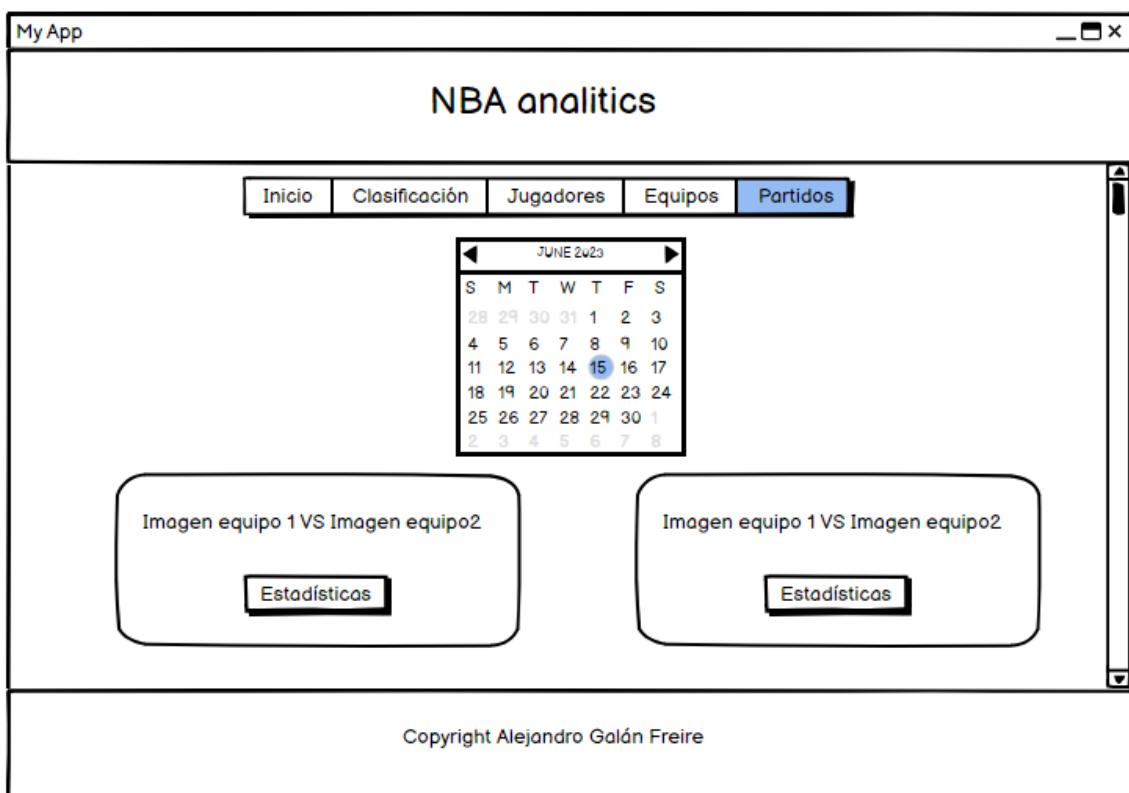
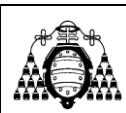


Ilustración 4-8 Prototipo pantalla partidos



Además, para los equipos tienen otra ventana relacionada que es la de la plantilla de este.

The screenshot shows a window titled 'My App' with the title bar 'NBA analytics'. Below the title bar is a navigation menu with tabs: 'Inicio' (Home), 'Clasificación' (Classification), 'Jugadores' (Players), 'Equipos' (Teams), and 'Partidos' (Matches). The 'Equipos' tab is currently selected. The main content area is titled 'NOMBRE EQUIPO - PLANTILLA' (Team Name - Roster). A table displays the following data:

Jugador	Edad	Posición	Más información
Lebron James	30	E	...
Lebron James	30	E	...
Lebron James	30	E	...
Lebron James	30	E	...
Lebron James	30	E	...
Lebron James	30	E	...
Lebron James	30	E	...

At the bottom of the window, the text 'Copyright Alejandro Galán Freire' is visible.

Ilustración 4-9 Prototipo pantalla de plantilla de equipo

Pasamos ahora a describir las pantallas de detalle de los jugadores, equipos y partidos. Todas ellas estructurarán las estadísticas correspondientes de la misma forma. Poseerán una cabecera con la imagen del elemento seleccionado y su nombre. Justo debajo una tabla con todas las estadísticas generales y después un conjunto de gráficos con más estadísticas. Las pantallas de detalle poseerán también un botón para comparar con otros jugadores o equipos.

Se ven a continuación todos sus prototipos.

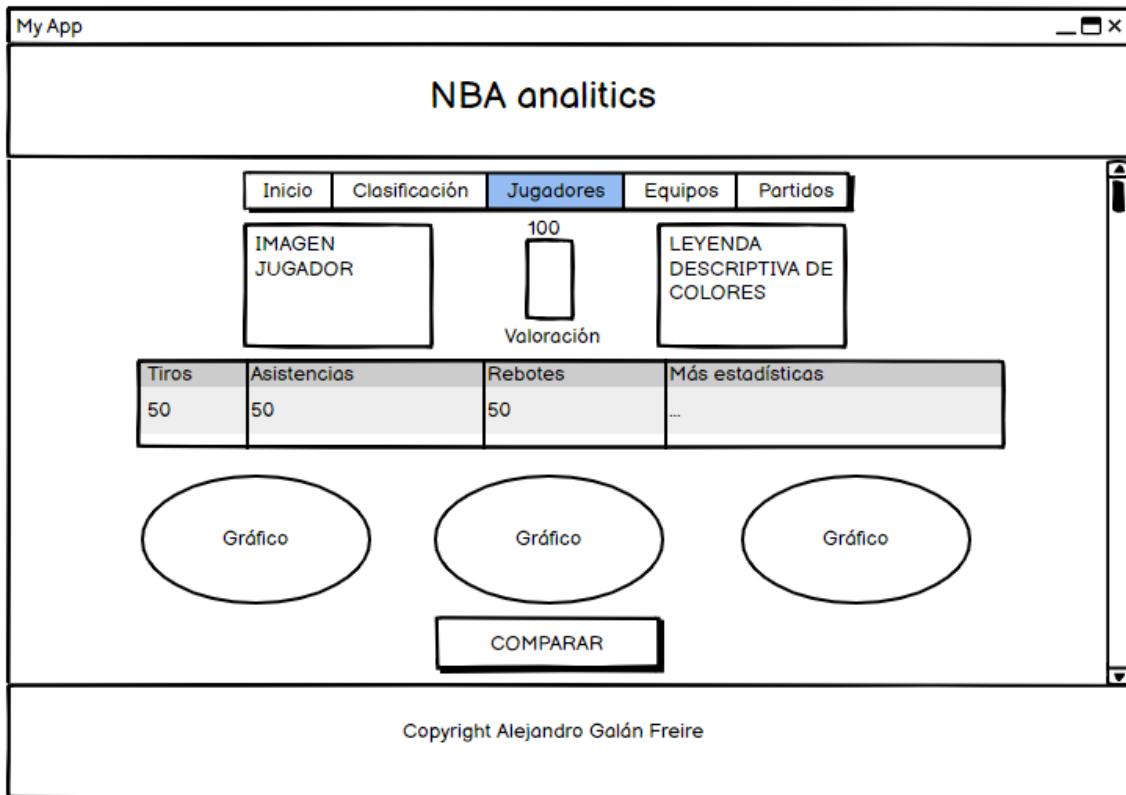


Ilustración 4-10 Prototipo pantalla de detalle de jugador

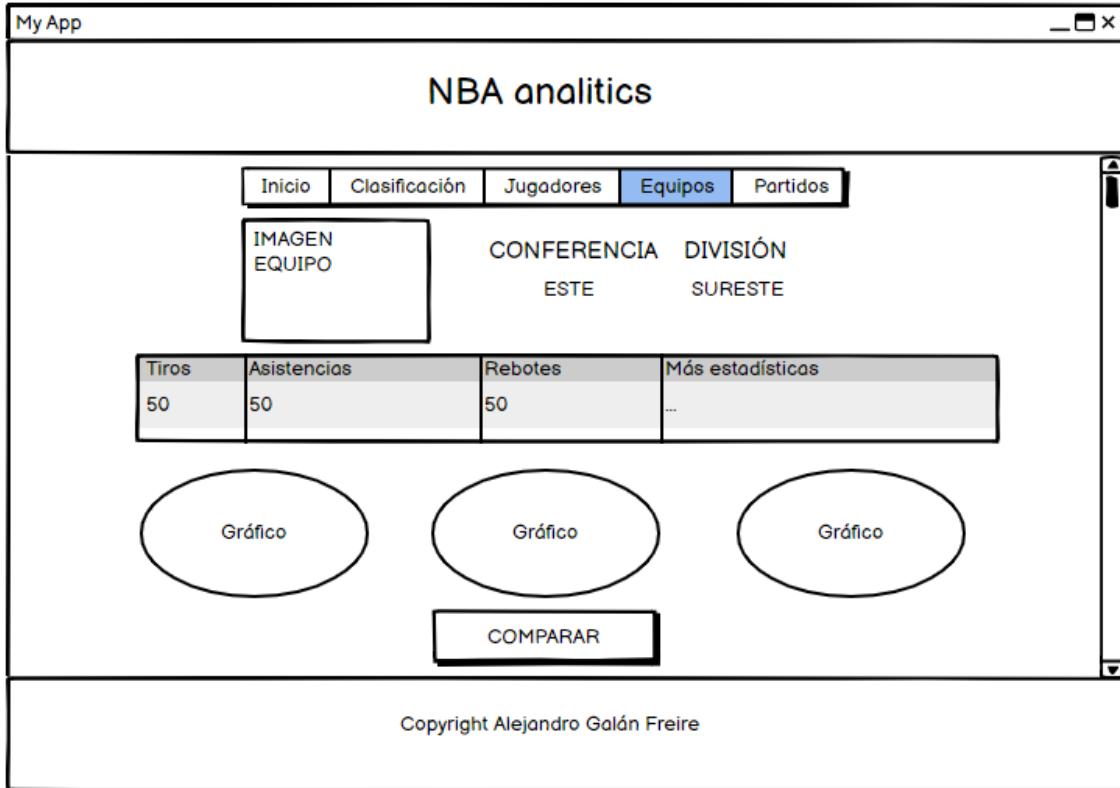


Ilustración 4-11 Prototipo pantalla de detalle de equipo

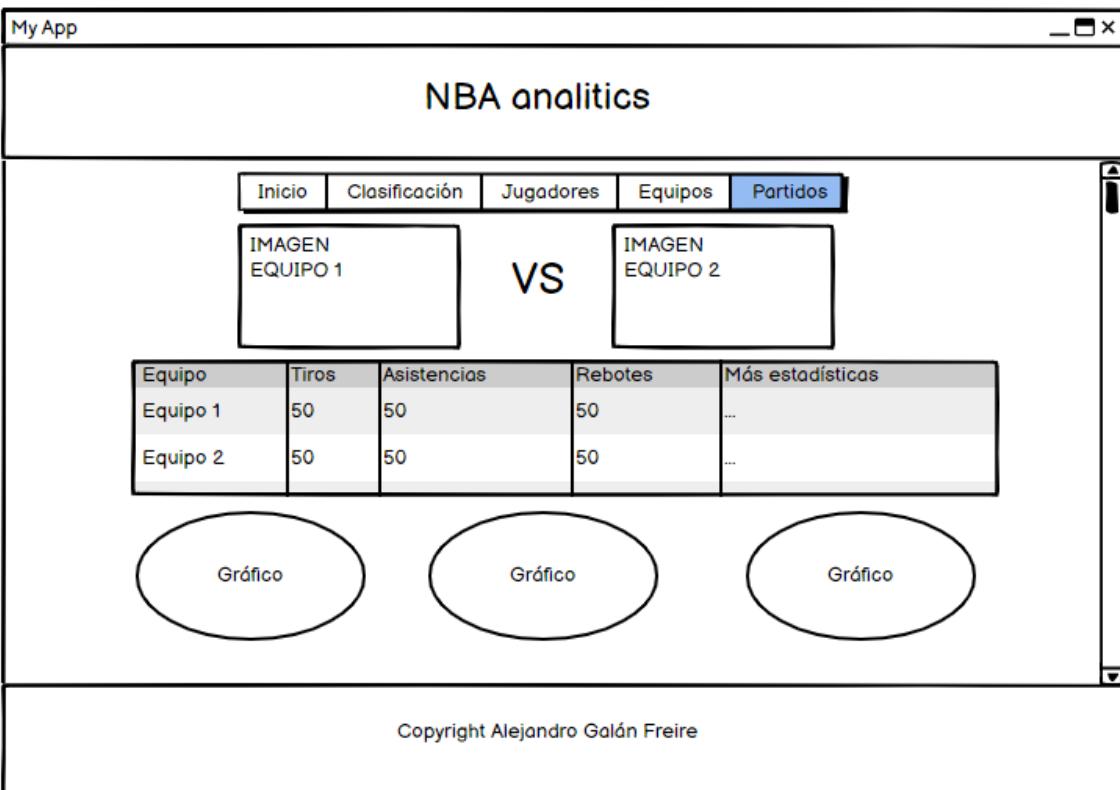


Ilustración 4-12 Prototipo pantalla de detalle de partidos

Y por último los dos últimos prototipos son para los diálogos de comparación de jugadores y equipos, que se despliegan cuando en el botón de comparar se selecciona el otro jugador o equipo. Dicho diálogo mostrará comparaciones de las estadísticas de los dos elementos a través de gráficos. Será igual tanto para jugadores como para equipos.

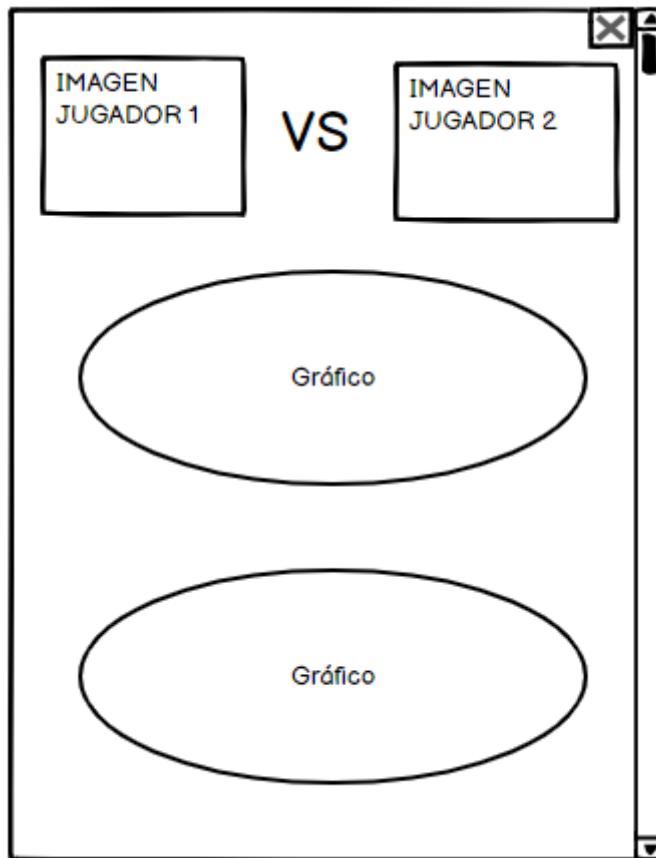


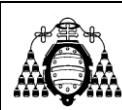
Ilustración 4-13 Prototipo pantalla de comparación de jugadores y equipos

4.7.2 Descripción del Comportamiento de la Interfaz

El usuario tendrá como ayuda la pequeña introducción que se le muestra nada más entra en la aplicación en la pantalla de inicio. Es una aplicación muy sencilla, que no tiene funcionalidad complicada y es muy intuitiva, por lo que se considera suficiente con esta breve introducción.

Los mensajes de error serán mensajes de texto informativos sobre el error que se muestran sobre la misma pantalla donde deberían aparecer los datos en caso de que todo fuera correctamente, o en otros casos, como que no se hayan podido recuperar los datos de la base de datos o se haya perdido la conexión, se mostrará una alerta indicando la acción que debería hacer el usuario para recuperarse de dicho error.

Finalmente, en cuanto a las entradas de los usuarios en los buscadores para encontrar jugadores y equipos ya sea para un filtrado de los mismos en los listados de visualización o para hacer una



comparación, se avisará al usuario si no se encuentran coincidencias con el texto introducido a través de un mensaje que aparecerá sobre la propia aplicación.

4.7.3 Diagrama de Navegabilidad

En el Diagrama de interactividad no se muestran todas las posibles relaciones para que no sea excesivamente complejo de comprender, así que se va a explicar un poco para evitar dudas.

El primer nivel muestra todos los niveles del menú (inicio, clasificación, jugadores, equipos, Partidos). Estos son accesibles desde cualquier otra ventana de la aplicación debido a que el menú estará visible en todas las pantallas. Es decir, una ventana de menú puede ser accedida desde otra pantalla del menú o cualquier otra no accesible directamente.

Aquellas que están en niveles por debajo son las que no pueden ser accedidas directamente, sino que se tiene que pasar previamente por la pantalla que está en su nivel superior, por ejemplo, es imposible llegar a la ventana de 'Detalle Jugador' sin habernos ubicado primero en 'Jugadores', o por otra pantalla que tenga una relación directa con esta, como es el caso de la plantilla hacia el detalle del jugador.

En el diagrama, las líneas indican bidireccionalidad, y las flechas unidireccionalidad.

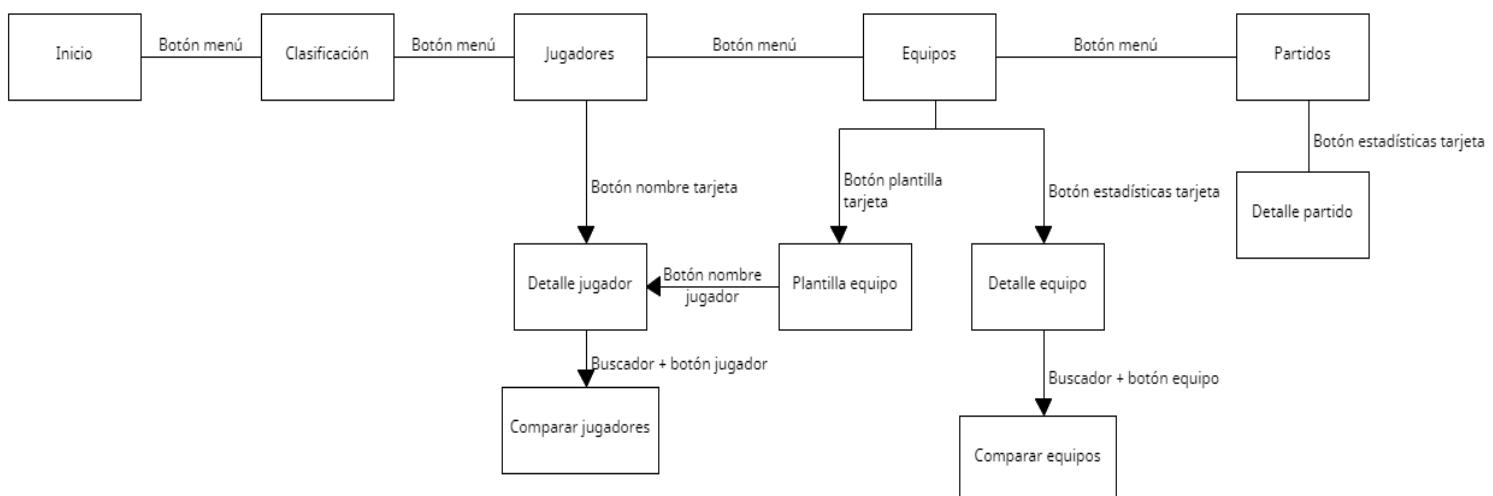


Ilustración 4-14 Diagrama de interactividad

Vamos ahora a explicar el significado de los mensajes en las líneas y flechas de transición:

Botón menú: es necesario presionar sobre el botón de menú correspondiente para acceder.

Botón nombre tarjeta: es necesario presionar sobre el título de la tarjeta de jugador o equipo para acceder.

Botón estadísticas tarjeta: es necesario presionar sobre el botón 'estadísticas' de la tarjeta del equipo o partido para acceder.

Botón plantilla tarjeta: es necesario presionar sobre el botón ‘plantilla’ de la tarjeta del equipo para acceder.

Buscador + Botón jugador/equipo: es necesario presionar primero sobre el botón ‘comparar’, para posteriormente introducir el nombre del jugador y partido, y posteriormente presionar sobre el botón de unas de las coincidencias para acceder.

4.8 Especificación del Plan de Pruebas

En esta sección vamos a crear y diseñar el plan de pruebas de la aplicación y los mecanismos que utilizaremos para detectar errores y corregirlos ya en la fase de implementación.

4.8.1 Pruebas unitarias y de calidad

Tanto en la parte frontend desarrollada con Angular como en el backend desarrollado con Spring Boot, se llevará cabo un proceso de estudio de la calidad del código a través de la herramienta SonarQube. Con esta herramienta podremos analizar la cantidad de código repetido entre clases, las malas prácticas llevadas a cabo en el código, BUGs que puedan surgir y, finalmente, la cobertura del código, es decir cuanta cantidad del mismo ha sido probado. Aquí es donde entran en juego las pruebas unitarias realizadas con Jasmine y Karma en Angular y con JUnit 4 en Spring Boot.

Las pruebas unitarias tratan de garantizar el funcionamiento correcto de cada clase, por lo que se han realizado para cada una de ellas. En el apartado [6.7.1 Pruebas Unitarias](#) se detalla la descripción de la aplicación de las pruebas que a continuación vamos a definir. Esta definición las vamos a agrupar en función de los distintos módulos que contiene el sitio web.

4.8.1.1 Pruebas Clasificación

Prueba	Resultado Esperado
Mostrar clasificación de una liga con varias conferencias	Se genera una tabla por cada conferencia

Tabla 63 Pruebas módulo Clasificación

4.8.1.2 Pruebas Jugadores

Prueba	Resultado Esperado
Calcular actuación de jugador	Se obtiene el resultado correcto con los datos introducidos
Prueba	Resultado Esperado
Obtener el color de la gráfica de valoración	Un color u otro en función del valor de la valoración
Prueba	Resultado Esperado
Obtener color de la gráfica de puntos creados	Un color u otro en función del valor de los puntos creados

Prueba	Resultado Esperado
Obtener color de la gráfica de porcentaje de tiro	Un color u otro en función del valor del porcentaje de tiro
Prueba	Resultado Esperado
Abrir diálogo de comparación	El diálogo se abre
Prueba	Resultado Esperado
Obtener altura de la gráfica de valoración	Un valor u otro en función del valor de la valoración
Prueba	Resultado Esperado
Obtener el color de la gráfica de eficiencia	Un color u otro en función del valor de la eficiencia
Prueba	Resultado Esperado
Inicializar gráfico de líneas	El gráfico se genera
Prueba	Resultado Esperado
Inicializar gráfico de tipo pie	El gráfico se genera
Prueba	Resultado Esperado
Inicializar el gráfico de barras	El gráfico se genera
Prueba	Resultado Esperado
Inicializar el gráfico de tipo radar	El gráfico se genera

Tabla 64 Pruebas módulo Jugadores

4.8.1.3 Pruebas Equipos

Prueba	Resultado Esperado
Abrir diálogo de comparación	El diálogo se abre
Prueba	Resultado Esperado
Obtener el color del gráfico de aparición en la final por año	Si texto 'nbaFinalsAppearance' = a un valor x el gráfico tendrá un color específico
Prueba	Resultado Esperado
Abrir diálogo de comparación	El diálogo se abre
Prueba	Resultado Esperado
Inicializar gráfico de líneas	El gráfico se genera
Prueba	Resultado Esperado
Inicializar gráfico de barras par el playoff	El gráfico se genera
Prueba	Resultado Esperado
Inicializar el gráfico de barras	El gráfico se genera
Prueba	Resultado Esperado
Inicializar el gráfico de tipo radar	El gráfico se genera

Tabla 65 Pruebas módulo Equipos

4.8.1.4 Pruebas partidos

Prueba	Resultado Esperado
Inicializar gráfico de líneas	El gráfico se genera
Prueba	Resultado Esperado
Inicializar el gráfico de barras	El gráfico se genera
Prueba	Resultado Esperado
Inicializar el gráfico de tipo radar	El gráfico se genera

Tabla 66 Pruebas módulo Partidos

4.8.2 Pruebas de integración y de sistema

Las pruebas de integración y sistema se realizarán manualmente llevando a cabo todos los escenarios identificados en [5.5 Análisis de Casos de Uso y Escenarios](#) para comprobar que el funcionamiento de la aplicación con todos sus módulos conectados es correcto.

Ahora vamos a realizar una especificación de las pruebas que se van a realizar y del resultado esperado de las mismas como se hizo para las pruebas unitarias. Se agruparán por casos de uso teniendo en cuenta todos los escenarios que suponen.

4.8.2.1 Pruebas CU-VI

Este escenario no tiene ningún tipo de comunicación con otros módulos ni implementación complicada, tan solo se trata de un mensaje de texto con su correspondiente maquetación por lo que no se realizarán pruebas de él.

4.8.2.2 Pruebas CU-VC

<u>Caso de Uso VC: Ver Clasificación</u>	
Prueba	Resultado Esperado
Acceder a la pantalla de clasificación	El menú de navegación resalta la opción correspondiente y muestra por defecto la clasificación de la NBA
Prueba	Resultado Esperado
Cambiamos la liga seleccionada	El sistema actualiza la tabla con la clasificación de la liga seleccionada (ACB será la única alternativa)
Prueba	Resultado Esperado
Liga sin datos por cualquier error	Se muestra un mensaje al usuario avisando de que la liga seleccionada no tiene datos

Tabla 67 Pruebas Caso de Uso Ver Clasificación

4.8.2.3 Pruebas CU-VJ, CU-DJ, CU-CJ

Caso de Uso VI, DJ, CJ: Jugadores	
Prueba	Resultado Esperado
Acceder a la pantalla de jugadores	El menú de navegación resalta la opción correspondiente y muestra por defecto las tarjetas de todos los jugadores
Prueba	Resultado Esperado
Filtrado por nombre	El sistema muestra tan solo aquellos jugadores que coincidan con el nombre o texto introducido ya que la búsqueda comienza desde la primera letra escrita
Prueba	Resultado Esperado
Filtrado por abreviatura de equipo	El sistema muestra tan solo aquellos jugadores que coincidan con la abreviatura del equipo o texto introducido ya que la búsqueda comienza desde la primera letra escrita
Prueba	Resultado Esperado
Filtrado por nombre no existente	Se avisa al usuario a través de un mensaje de que no existen coincidencias con esa búsqueda
Prueba	Resultado Esperado
Acceso al detalle del jugador	Se presiona sobre el título de la tarjeta de un jugador y se visualiza la pantalla de detalle del mismo
Prueba	Resultado Esperado
Interacción con gráficas en el detalle del jugador	Las gráficas muestran datos y son interactivas
Prueba	Resultado Esperado
Buscar jugador para comparar	Se muestran todos los jugadores cuyo nombre coincide con el texto introducido en el buscador de comparación de jugadores
Prueba	Resultado Esperado
Jugador buscado para comparar no existe	Se avisa al usuario a través de un mensaje de que no existen coincidencias con esa búsqueda
Prueba	Resultado Esperado
Comparar jugador	Se muestra un diálogo con gráficas comparativas entre el jugador del detalle y el seleccionado en el buscador de comparación
Prueba	Resultado Esperado
Interacción con gráficas en el diálogo de comparación de jugadores	Las gráficas muestran datos y son interactivas

Tabla 68 Pruebas Casos de Uso Jugadores

4.8.2.4 Pruebas CU-VE, CU-PE, CU-DE, CU-CE

Caso de Uso VE, PE, DE, CE: Equipos	
Prueba	Resultado Esperado
Acceder a la pantalla de equipos	El menú de navegación resalta la opción correspondiente y muestra por defecto las tarjetas de todos los equipos

Prueba	Resultado Esperado
Filtrado por nombre	El sistema muestra tan solo aquellos equipos que coincidan con el nombre o texto introducido ya que la búsqueda comienza desde la primera letra escrita
Prueba	Resultado Esperado
Filtrado por nombre no existente	Se avisa al usuario a través de un mensaje de que no existen coincidencias con esa búsqueda
Prueba	Resultado Esperado
Acceso a la plantilla de equipo	Se muestra la plantilla del equipo al presionar la opción correspondiente en la tarjeta del mismo
Prueba	Resultado Esperado
Acceso al detalle del jugador a través de plantilla	Se muestra la pantalla de detalle del jugador seleccionado desde la plantilla
Prueba	Resultado Esperado
Acceso al detalle del equipo a través del título de la tarjeta	Se muestra la pantalla de detalle del equipo
Prueba	Resultado Esperado
Acceso al detalle del equipo a través del botón estadísticas de la tarjeta	Se muestra la pantalla de detalle del equipo
Prueba	Resultado Esperado
Interacción con gráficas en el detalle del equipo	Las gráficas muestran datos y son interactivas
Prueba	Resultado Esperado
Buscar equipo para comparar	Se muestran todos los equipos cuyo nombre coincide con el texto introducido en el buscador de comparación de equipos
Prueba	Resultado Esperado
Equipo buscado para comparar no existe	Se avisa al usuario a través de un mensaje de que no existen coincidencias con esa búsqueda
Prueba	Resultado Esperado
Comparar equipo	Se muestra un diálogo con gráficas comparativas entre el equipo del detalle y el seleccionado en el buscador de comparación
Prueba	Resultado Esperado
Interacción con gráficas en el diálogo de comparación de equipos	Las gráficas muestran datos y son interactivas

Tabla 69 Pruebas Casos de Uso Equipos

4.8.2.5 Pruebas CU-VP, CU-DP

Caso de Uso VP, DP: Partidos	
Prueba	Resultado Esperado
Acceder a la pantalla de partidos	El menú de navegación resalta la opción correspondiente y muestra por defecto las tarjetas de todos los partidos con la fecha por defecto del último partido
Prueba	Resultado Esperado

Seleccionar fecha con partidos	El sistema muestra tan solo aquellos partidos que se hayan jugado en la fecha seleccionada en el calendario
Prueba	Resultado Esperado
Seleccionar fecha sin partidos	El sistema muestra un mensaje al usuario avisando de que no existen partidos en la fecha seleccionada
Prueba	Resultado Esperado
Acceso al detalle del partido	Se muestra la pantalla de detalle del partido seleccionado
Prueba	Resultado Esperado
Interacción con gráficas en el detalle del partido	Las gráficas muestran datos y son interactivas

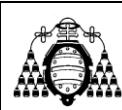
Tabla 70 Pruebas Casos de Uso Partidos

4.8.3 Pruebas de usabilidad

Se realizarán pruebas de usabilidad con usuarios reales para comprobar si la aplicación cumple con los objetivos planteados al inicio del proyecto, siendo atractiva y fácil de usar para el usuario.

Las pruebas consistirán en una serie de ejercicios que tendrá que hacer cada usuario interaccionando con la aplicación mientras nosotros los observamos, y posteriormente una serie de preguntas que les realizaremos sobre su experiencia con el sitio web.

Este proceso aparece detallado en [6.7.3 Pruebas de Usabilidad y Accesibilidad](#).



Capítulo 5. Diseño del Sistema

5.1 Arquitectura del Sistema

5.1.1 Diagramas de Paquetes

Se ha realizado un diagrama de paquetes por cada subsistema de la aplicación, uno para el frontend y otro para el backend.

En ambos se han considerado paquetes aquellas partes modularizables, es decir, que son independientes del resto y que no afectaría al funcionamiento de la aplicación en función de desaparecer o actualizarse.

5.1.1.1 *Diagrama de paquetes frontend*

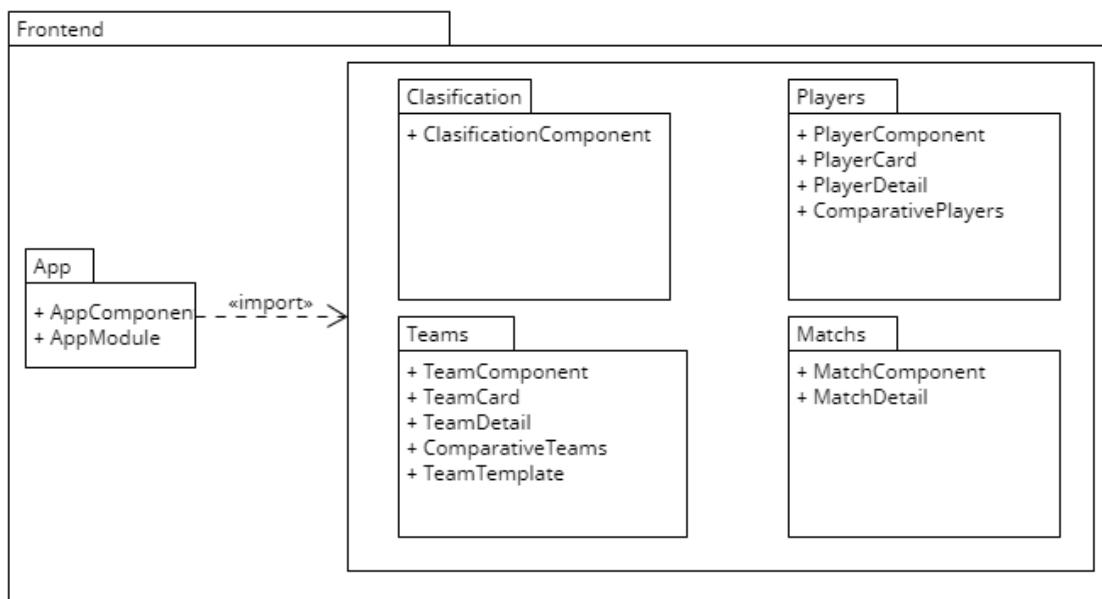


Ilustración 5-1 Diagrama de paquetes frontend

5.1.1.1.1 Paquete App

Paquete principal de la parte frontend que importa el resto de los paquetes generando las rutas de la aplicación. Dispone de dos clases, `AppComponent` que es la raíz de todo el sistema y `AppModule` que es la encargada de generar las rutas y todas las importaciones globales de la aplicación.

5.1.1.1.2 Paquete Clasification

Paquete correspondiente al módulo de clasificación.

5.1.1.1.3 Paquete Players

Paquete correspondiente al módulo de jugadores. Contiene todo lo relacionado con las posibles operaciones que se pueden realizar con estos. PlayersComponent para el listado, PlayerCard para generar las tarjetas de cada jugador, PlayerDetail para obtener el detalle de un jugador y por último, PlayerComparative para poder comparar varios jugadores.

5.1.1.1.4 Paquete Teams

Aplicable lo dicho para Players.

5.1.1.1.5 Paquete Matchs

Este paquete se limita a contener todo lo relacionado con los partidos, que en este caso sería el listado de los mismo a través de MatchComponent así como el detalle de uno en concreto con MatchDetail.

5.1.1.2 Paquete Backend

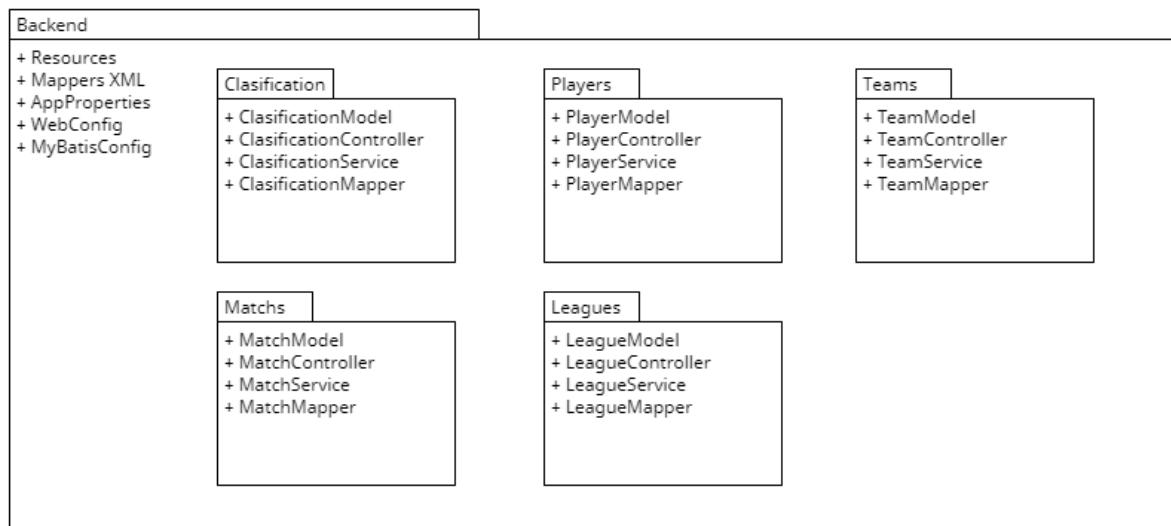


Ilustración 5-2 Diagrama de paquetes backend

5.1.1.2.1 Elementos globales

El backend en sí, además de contener todos los paquetes que describiremos a continuación, presenta una serie de elementos necesarios para el correcto funcionamiento del subsistema.

Estos elementos consisten en la configuración de la aplicación (AppProperties, WebConfig), del framework utilizado (MyBatis) y finalmente la carpeta relacionada con los ficheros XML que contienen las consultas a la Base de Datos (Mappers XML).

5.1.1.2.2 Resto de paquetes

Vamos a hacer una explicación general de los paquetes ya que para todos ellos es aplicable lo mismo.

En primer lugar, cada paquete contiene su modelo, que se encarga de generar el objeto que contendrá los datos recogidos de la Base de Datos.

Posteriormente se crea el controlador que define todos los endpoints del subsistema. Cada uno de estos realizarán la lógica de negocio con los datos obtenidos de la llamada al Servicio. Es decir, el Servicio es el encargado de devolverle los datos al controlador, y para ello tiene que hacer una llamada al Mapper, interfaz que define sus métodos y que son redefinidos a través de los Mapper XML para hacer la consulta a la Base de Datos y obtener los datos solicitados.



5.1.2 Diagrama de Despliegue

En este apartado se va a definir la estructura de despliegue de la aplicación. Hay que tener en cuenta que esta aplicación no se ha desplegado en internet, sino que funciona localmente.

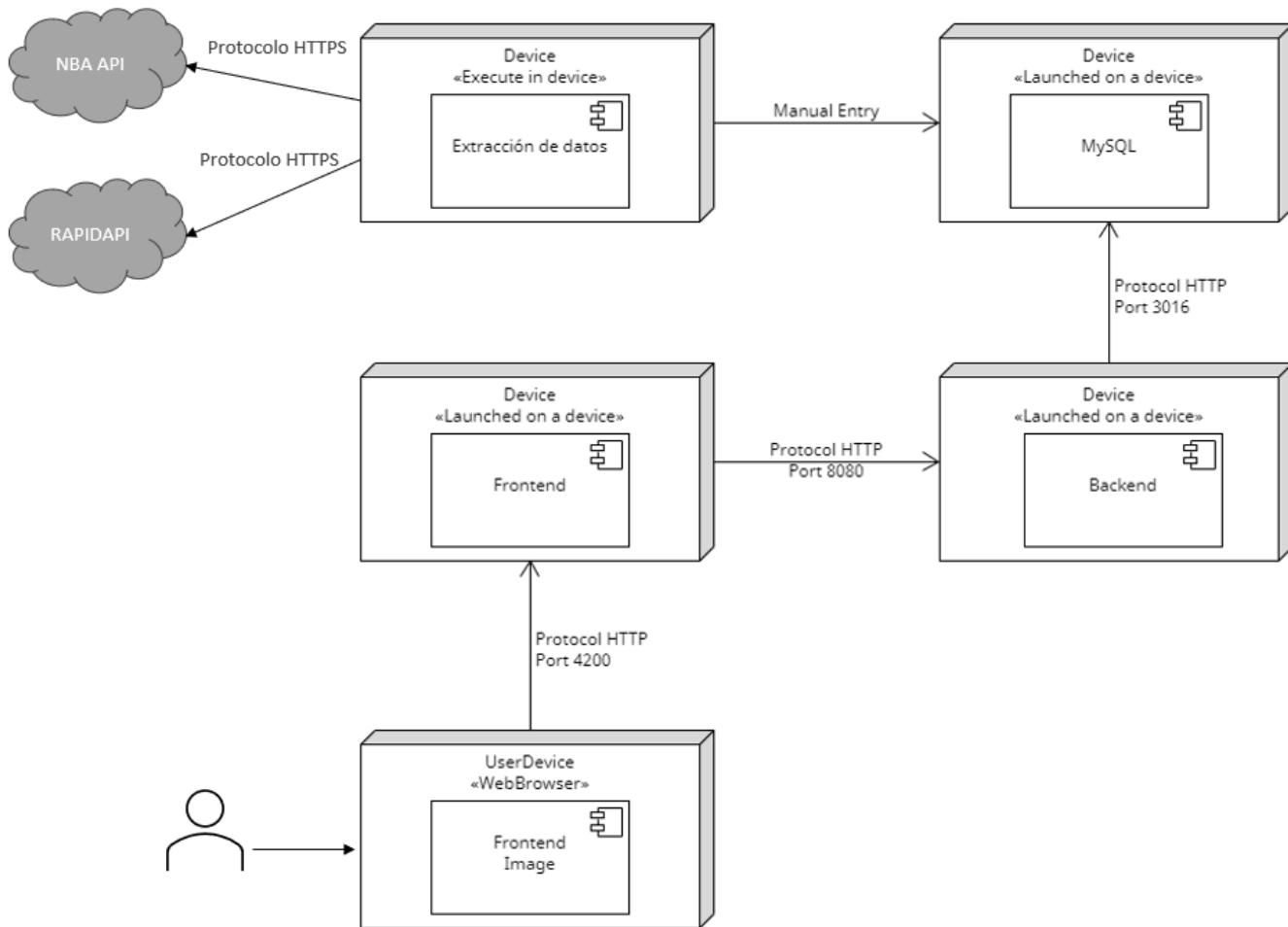


Ilustración 5-3 Diagrama de Despliegue

Toda esta estructuración nos permite que la aplicación sea escalable y modularizable, sin depender directamente de las APIs de datos al extraer esa información y almacenarla en la base de datos propia del sistema. Esto hace que podamos tratar los datos a nuestro gusto y que no dependamos de la disponibilidad de la API ni de sus actualizaciones, además de no tener que limitarnos al consumo de datos en una sola API, sino de las que deseemos.

5.1.2.1 Frontend Image

Es la aplicación que el usuario visualiza desde su dispositivo, a través de un navegador web. Se conecta al frontend a través del protocolo HTTP al puerto 4200.

5.1.2.2 *Frontend*

Parte frontend de la aplicación, es la que está lanzada en el puerto 4200 y la encargada de mostrar al usuario la información solicitada. No solo la muestra, sino que también realiza las peticiones correspondientes al backend para obtener los datos necesarios, conectándose con el a través del puerto 8080 con el protocolo HTTP.

5.1.2.3 *Backend*

Encargado de devolver la información solicitada desde el frontend al mismo, obteniendo dichos datos de la Base de Datos a la que se conecta también a través del protocolo HTTP al puerto 3306.

5.1.2.4 *MySQL*

Base de Datos del sistema, donde se almacena toda la información y de donde el backend la captura. Dicha información se almacena manualmente tras la extracción de los datos de las APIs.

5.1.2.5 *Extracción de Datos*

Conjunto de ficheros Python que se conectan a las APIs utilizadas a través de diversos endpoints que estas definen, con el protocolo HTTPS. Estos documentos se encargan de obtener la información tal y como la devuelven las APIs y formatearla con la estructura que presenta la Base de Datos del sistema.

Una vez esta información esta correctamente estructurada se envía a ficheros CSV y estos se importan en la Base de Datos a través de una opción que nos aporta MySQLWorkbench, entorno de tratamiento de la Base de Datos utilizado en este proyecto.



5.2 Diseño de Clases

5.2.1 Diagrama de Clases

Como se hizo en [6.1.1 Diagramas de Paquetes](#) se va a hacer un diagrama por cada subsistema (frontend y backend), ya que cada uno es independiente del otro.

5.2.1.1 Diagrama de clases Backend

En este subsistema, a su vez cada módulo es independiente, así que no tendría sentido hacer un diagrama muy grande en el que salieran grupos de clases que no se relacionaran entre sí. Para que quede más claro se mostrará el diagrama de clases que conforma cada módulo.

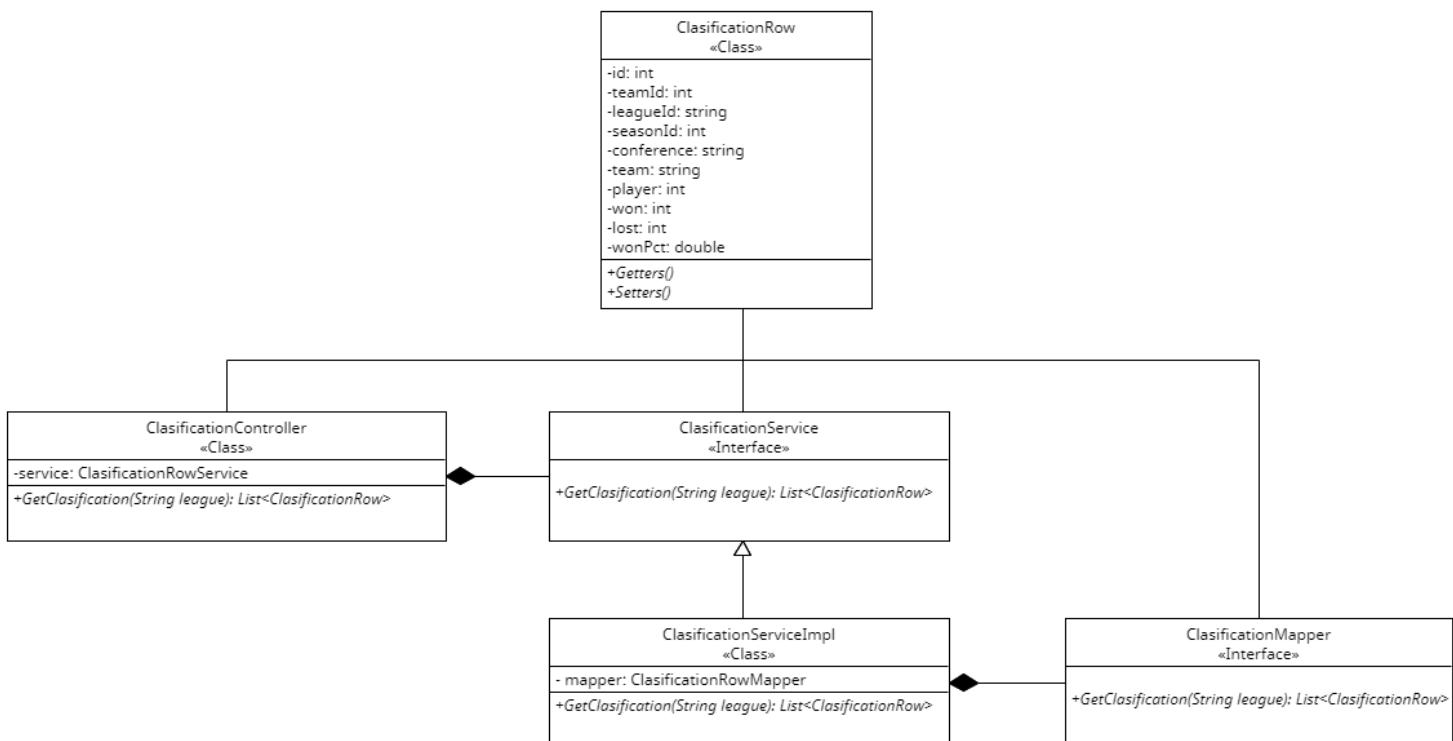
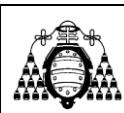


Ilustración 5-4 Diagrama de clases Clasificación

En el diagrama se puede ver la clase **ClasificationRow**, esta es la que representa el modelo de datos y contiene todos los atributos de la clase que será enviada al frontend para operar con los datos y que el usuario pueda visualizarlos.

Vemos también que esta clase tiene tres relaciones del tipo 'Asociación' con el resto de las clases, debido a que todas ellas contienen dicha clase como objeto de transmisión de la información. Dicho objeto, para que se pueda realizar esto, ha de tener los mismos atributos con los mismos nombres que los definidos en la entidad correspondiente de la base de datos.



Después pasamos con la clase ClasificationController. Es la encargada de definir todos los endpoints que responderán las peticiones recibidas desde el frontend. Contiene un ClasificationService como atributo, por lo que estamos antes una relación de tipo 'Composición' ya que el controlador sin el servicio no serviría de nada y desaparecería.

La clase ClasificationServiceImpl es la encargada de implementar los métodos definidos en la interfaz ClasificationService. La razón de crear esta interfaz intermedia es por seguridad y separación de capas, de esta forma el controlador no conoce la implementación del servicio, y si este sufriera cualquier cambio, tanto el controlador como cualquiera de sus clientes (clases que lo utilizan) no se verían afectados. Tiene también una relación de 'Composición' con el mapper.

El Mapper es el elemento final, el que se va a comunicar con la Base de Datos realizando la operación necesaria, en nuestro caso obteniendo los datos. Para ello se define la interfaz ClasificationMapper, que especifica los métodos que tendrán que ser referenciados en unos ficheros XML que contendrán las consultas, ya que esta es la forma de funcionar con el framework MyBatis, utilizado en este sistema.

A continuación, mostraremos el resto de los diagramas del backend, en los que es aplicable la explicación anterior.

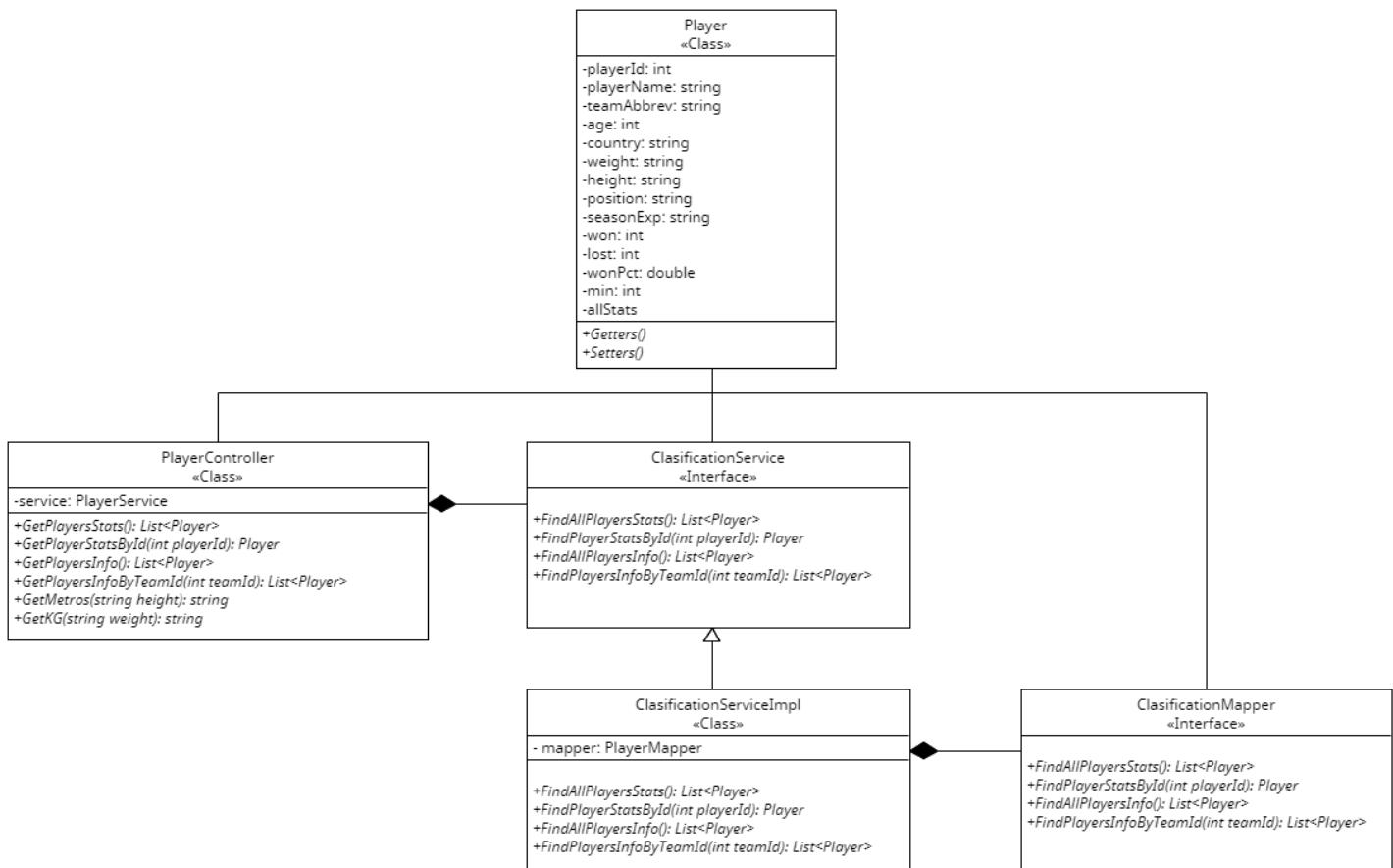


Ilustración 5-5 Diagrama de clases Player

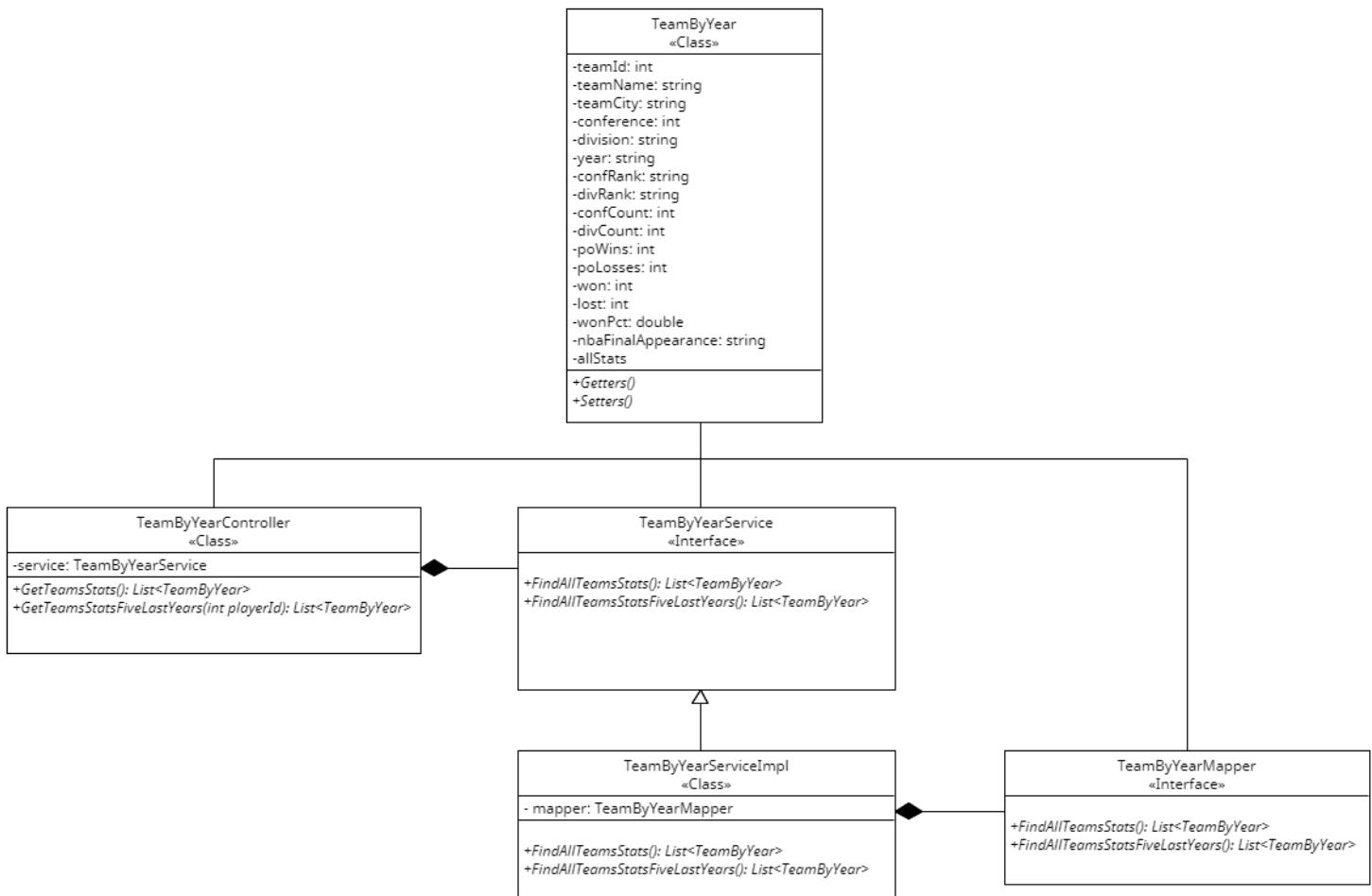


Ilustración 5-6 Diagrama de clases TeamByYear

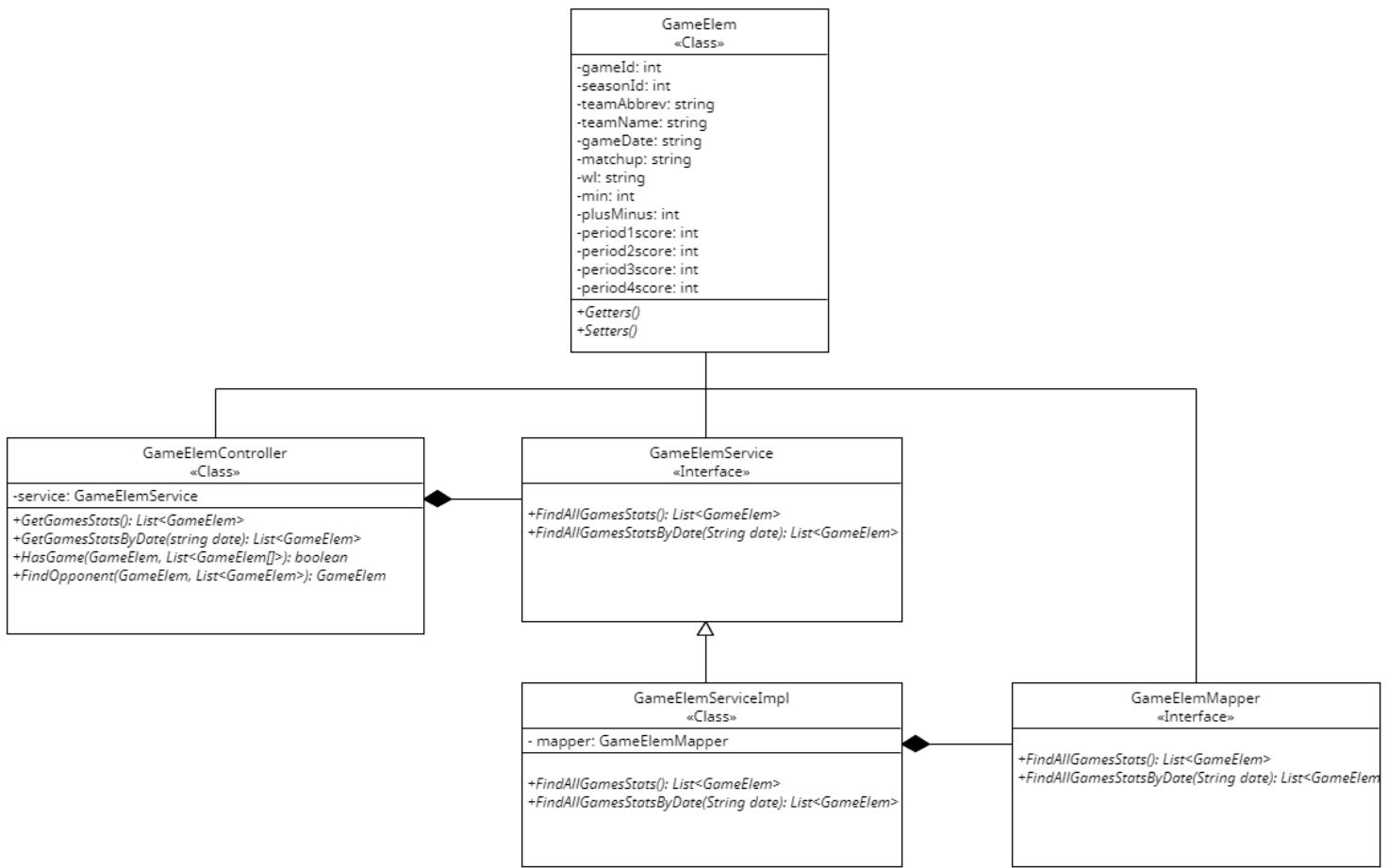


Ilustración 5-7 Diagrama de clases GameElem

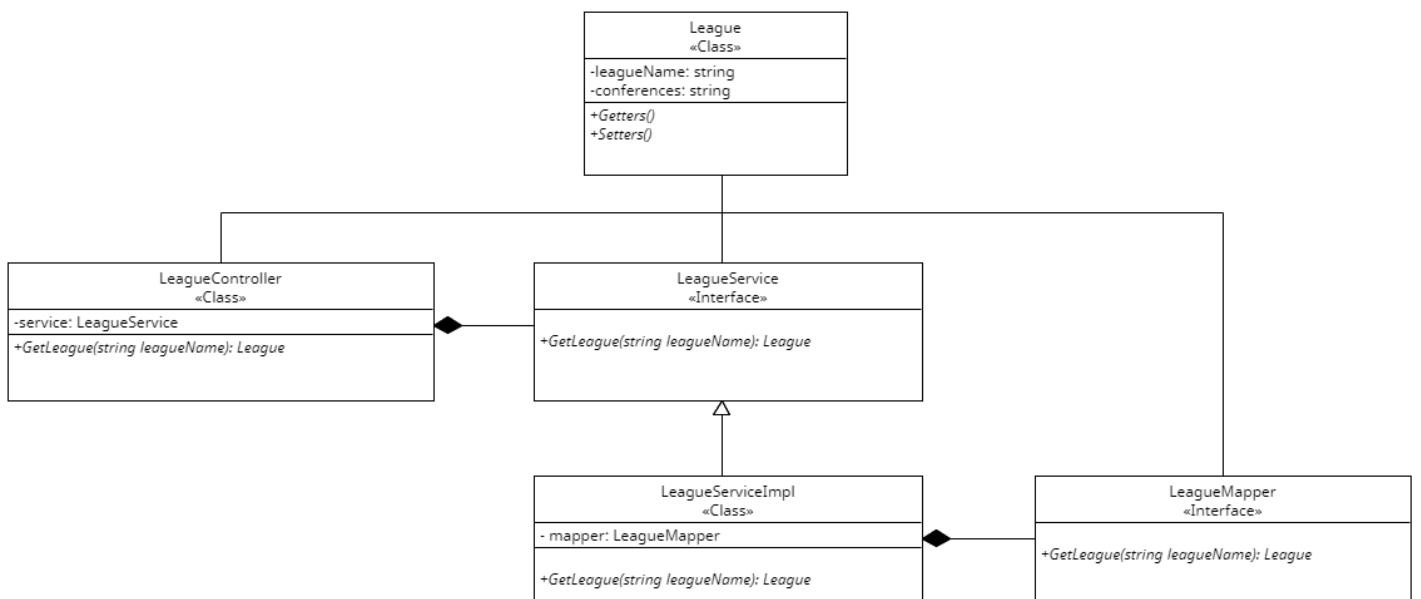


Ilustración 5-8 Diagrama de clases League

5.2.1.2 Diagrama de clases Frontend

El diagrama de esta parte es mucho más grande que la parte del backend, por lo que en los modelos (ClasificationRow, League, Player, TeamByYear y GameElem) se han omitido los atributos ya que son exactamente que los especificados en los diagramas del backend.

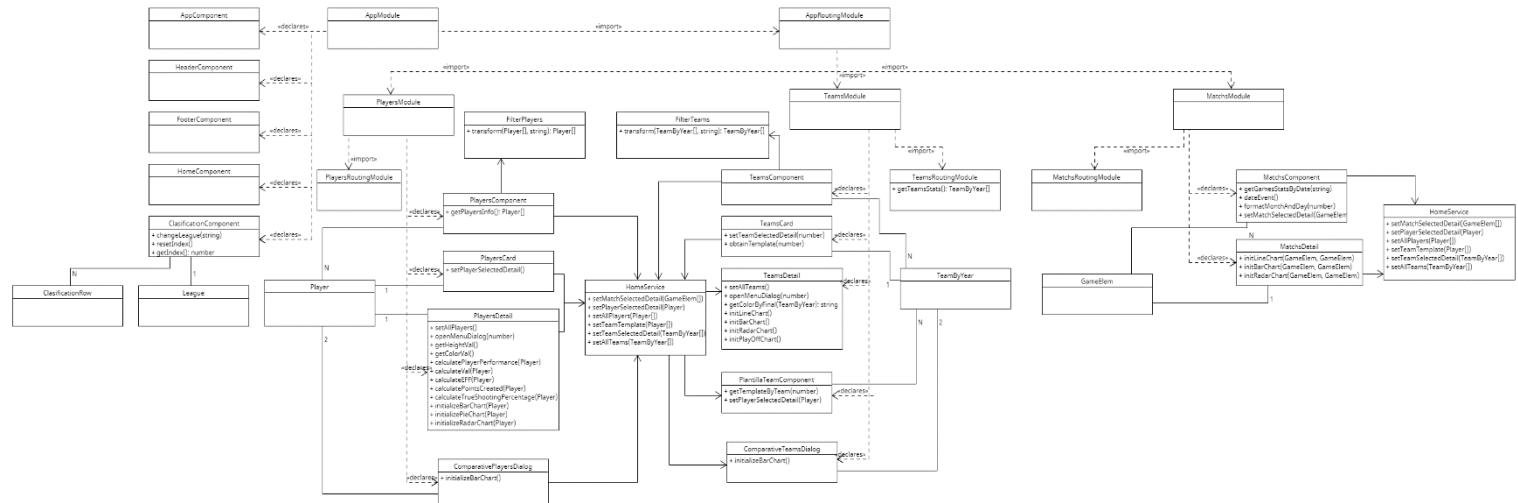


Ilustración 5-9 Diagrama de clases front

El diagrama anterior se muestra para que se puedan ver todas las relaciones existentes entre los distintos elementos del subsistema, pero como se puede observar, al ser tan grande no se aprecia bien. Por ello, vamos a explicarlo por partes.



Ilustración 5-10 Diagrama de clases Raíz



Como el propio nombre lo indica, el diagrama de encima sería la parte raíz de la aplicación, donde se declaran todos los componentes iniciales (Header, Footer) y aquellos subsistemas que no requieren la creación de módulos (Home, Clasification).

El AppModule importa el AppRoutingModule que es el que va a declarar las rutas a estos dos subsistemas directamente y a los restantes a través de los módulos creados para ellos.

Por último, se puede observar que *ClasificationComponent* utilizar los modelos *League*, pudiendo tener solo 1 objeto de este tipo, y *ClasificationRow* pudiendo tener N.

La siguiente parte es la del módulo de jugadores.

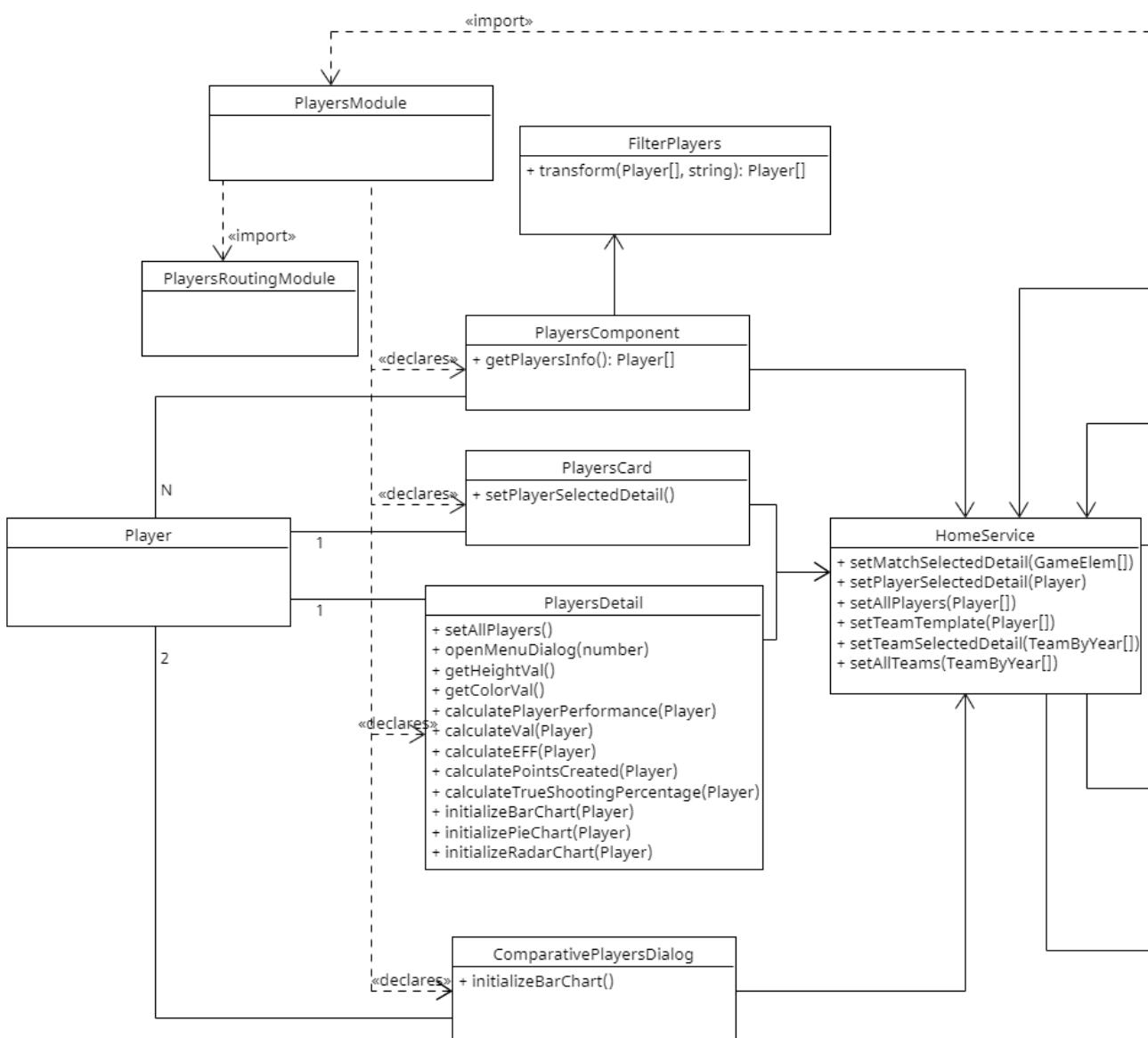


Ilustración 5-11 Diagrama de clases módulo Players



Aquí se puede observar cómo se ha creado un nuevo módulo que hace referencia a otro nuevo routing, *PlayersRoutingModule* que es el encargado de crear las rutas correspondientes a las pantallas del jugador. Todas las pantallas del jugador utilizan el modelo Player y además se comunican entre ellas a través de la clase compartida *HomeService*. *FilterPlayers* es un pipe de Angular creado para poder hacer el filtrado de jugadores según su nombre o equipo.

Un aspecto de mejora en cuanto a la estructuración del frontend sería dividir esta *HomeService* en varias, quedando una para cada módulo, es decir, en este caso tener, por ejemplo, un *PlayerService*.

Esto que se acaba de explicar es aplicable para el resto de las partes del diagrama, que son las siguientes.

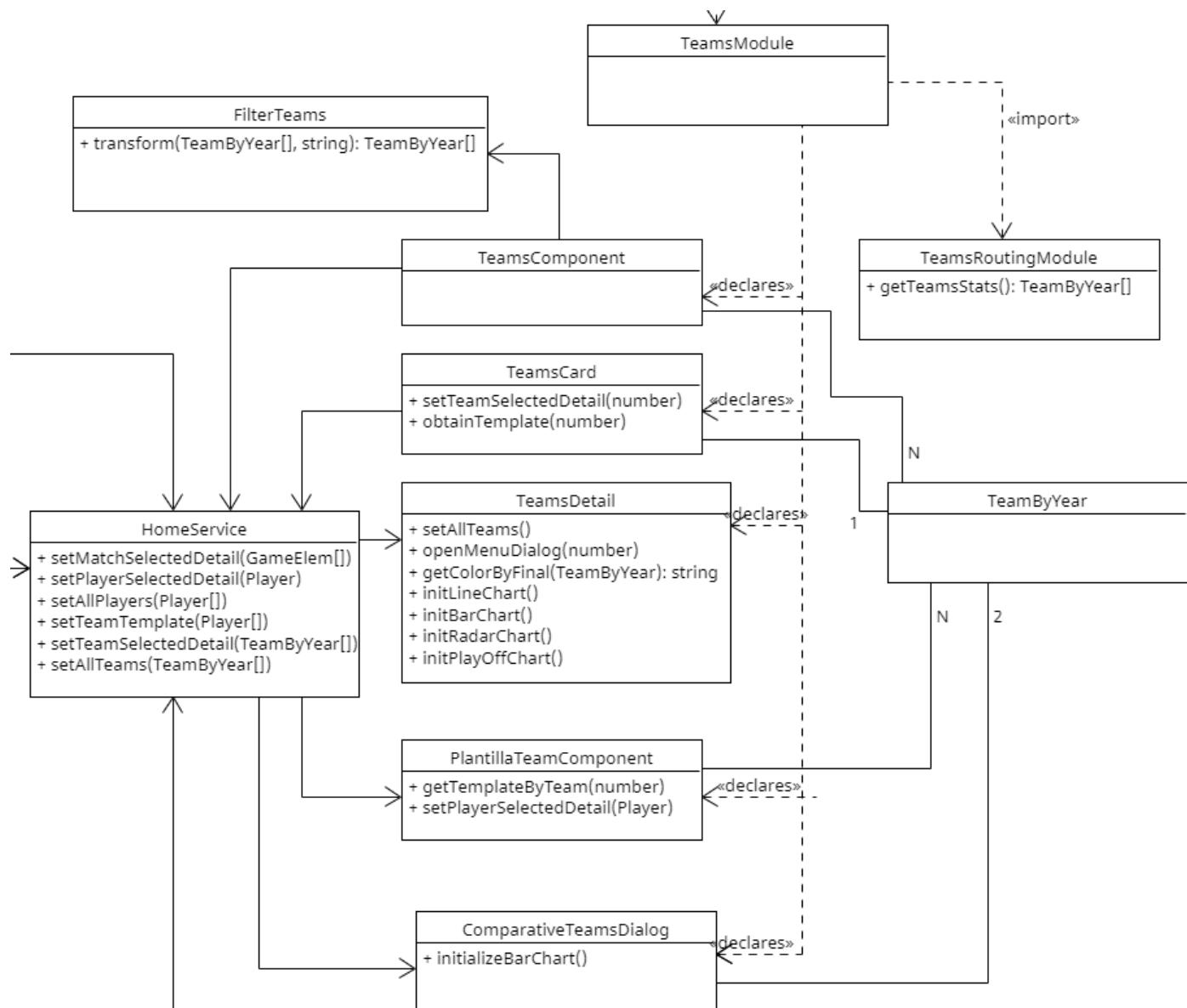


Ilustración 5-12 Diagrama de clases módulo Teams

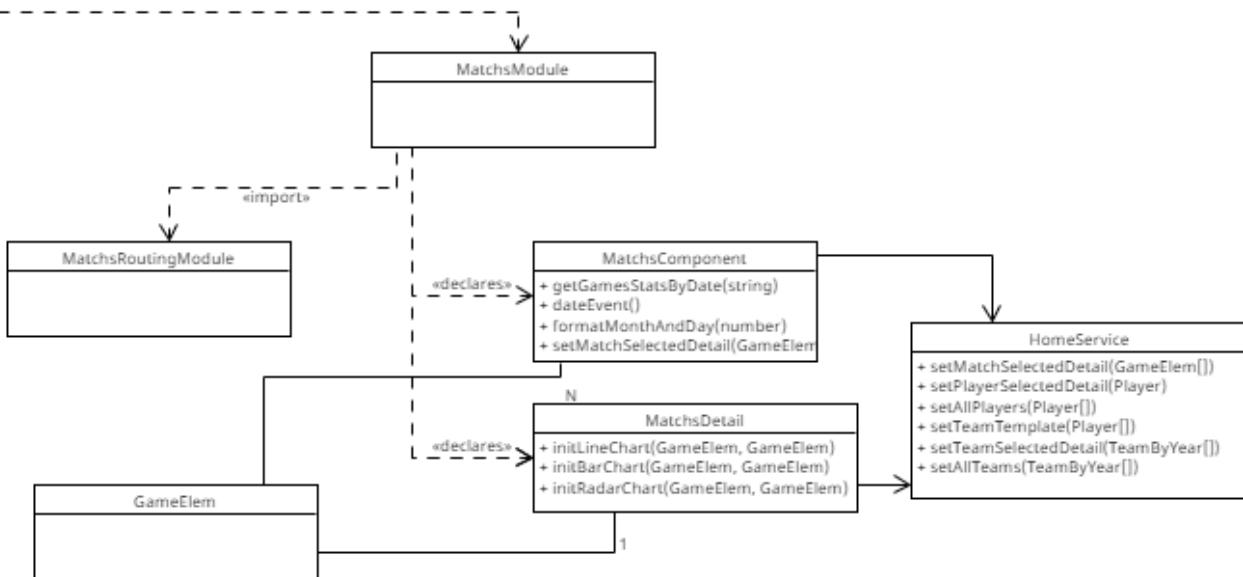


Ilustración 5-13 Diagrama de clases módulo Matches

Además de todas estas clases del diagrama existe otra llamada *DataService* que es la encargada de definir los métodos que realizan las peticiones a las URL definidas en el Backend, en definitiva, la que permite la conexión con el Backend para la extracción de los datos.

5.3 Diagramas de Interacción

Vamos a estudiar los diagramas de secuencia de un par de escenarios. No son los más complejos que presenta la aplicación, ya que estos abarcan una gran cantidad de clases y serían diagramas que no se apreciarían demasiado bien en el documento.

5.3.1 Visualizar partido

Este escenario consiste en acceder a la sección de ‘Partidos’ a través del menú y entrar en el detalle de uno de los partidos disponibles en la fecha por defecto. A continuación, se muestra el diagrama de secuencia entre los distintos objetos que interviene para la consecución de esta funcionalidad.

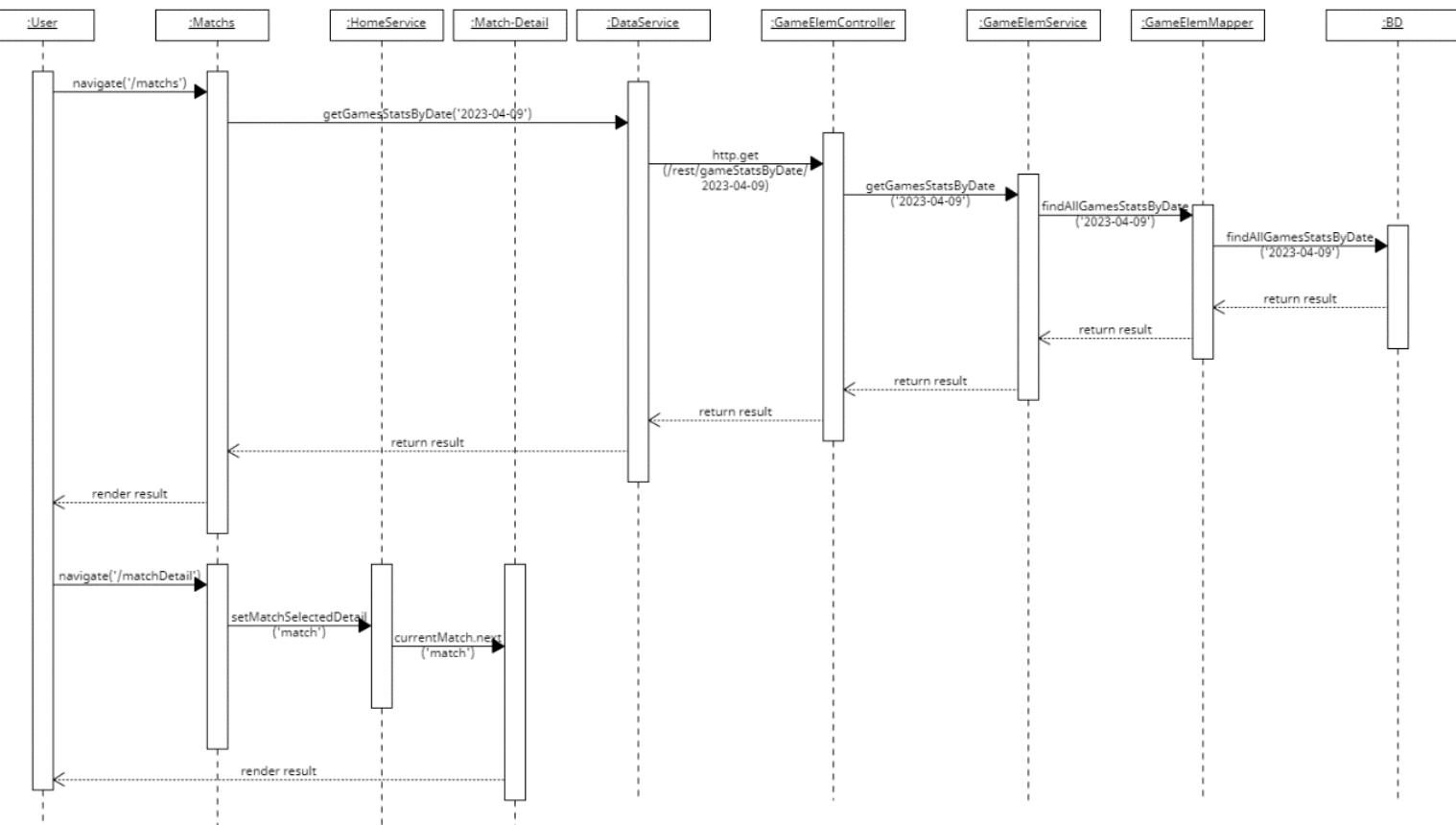


Ilustración 5-14 Diagrama de secuencia Visualizar Partido

En primer lugar, el usuario navega hacia la ventana de partidos a través del menú de navegación. En este punto, el componente de partidos envía la petición al controlador a través de la clase que comunica frontend y backend, el `dataService`. Esta petición se manda con la fecha '2023-04-09' que se corresponde con el último día en el que se disputaron partidos de la temporada regular.

El controlador se comunica con el servicio y este con el mapeador. Este último recupera los datos de la Base de Datos y los va devolviendo hasta que lleguen de nuevo al componente de partidos. Así es como se renderizan todos los partidos en esa fecha.

Ahora el usuario navega hacia el detalle de un partido presionando sobre uno de ellos. La `homeService`, clase que contiene la información compartida dentro del frontend, establece el partido seleccionado y avisa al componente del detalle del partido. Este recibe la información y la renderiza para el usuario.



5.3.2 Visualizar jugador específico tras búsqueda

Este escenario consiste en acceder a la sección de menú 'Jugadores' y en la búsqueda indicar el nombre de un jugador cualquiera. A continuación, se muestra el diagrama de secuencia entre los distintos objetos que interviene para la consecución de esta funcionalidad.

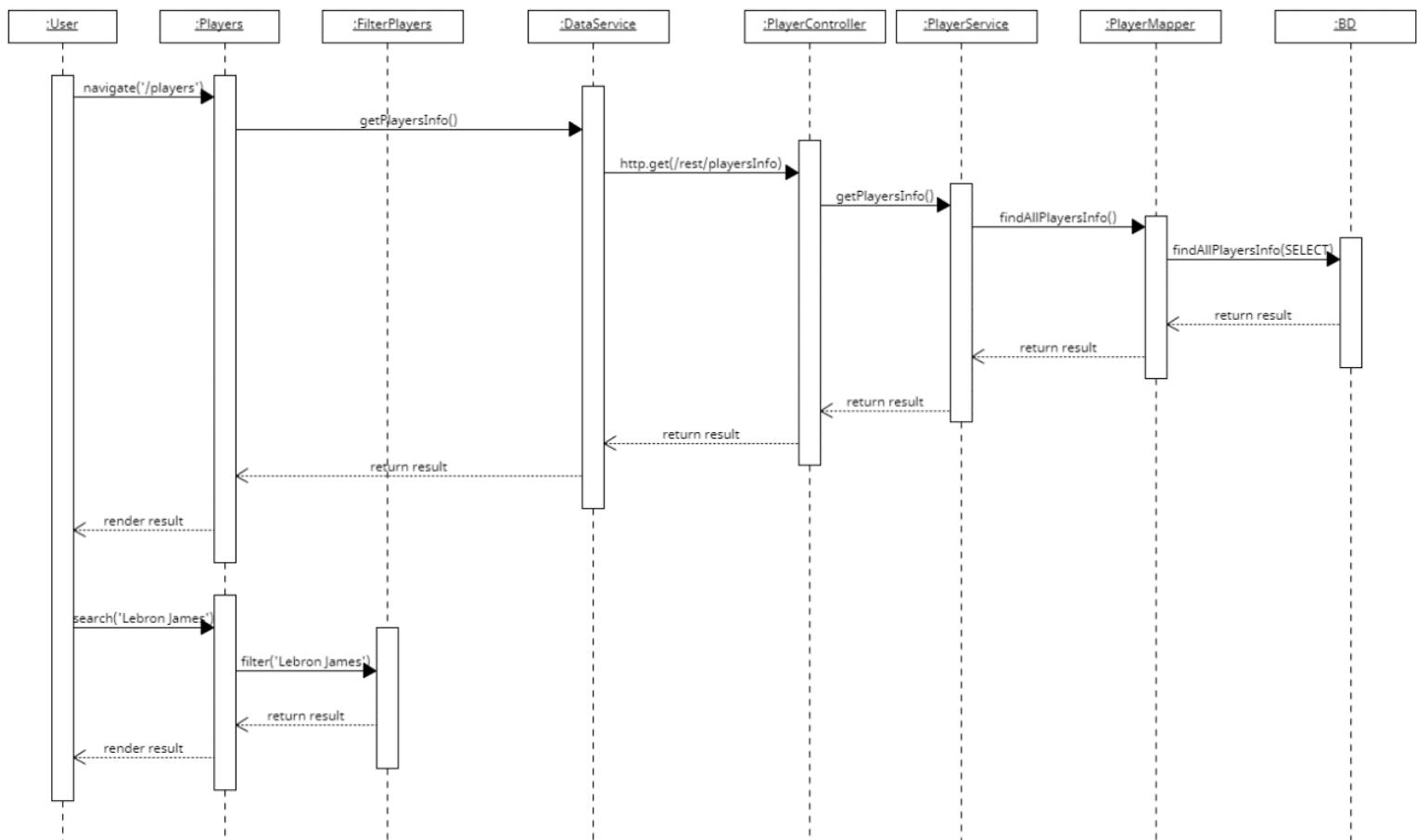


Ilustración 5-15 Diagrama de secuencia Visualizar Jugador buscado

En primer lugar, el usuario navega hacia la ventana de jugadores a través del menú de navegación. En este punto, el componente de jugadores envía la petición al controlador a través de la clase que comunica frontend y backend, el dataService.

El controlador se comunica con el servicio y este con el mapeador. Este último recupera los datos de la Base de Datos y los va devolviendo hasta que lleguen de nuevo al componente de jugadores. Así es como se renderizan todos los jugadores de la temporada.

Ahora el usuario realiza la búsqueda escribiendo el nombre 'Lebron James'. La clase FilterPlayers se encarga de filtrar los jugadores, devolviendo al componente de jugadores solo aquellos que coinciden con lo buscado, en este caso un único jugador.

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 112 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	---

Por último, el componente de jugadores vuelve a renderizar la pantalla para el usuario con un único jugador.

5.4 Diseño de la Base de Datos

5.4.1 Descripción del SGBD Usado

Este proyecto dispone de un Sistema Gestor de Base de Datos MySQL, usando la versión 8.0.32 y siendo gestionado a través de la herramienta MySQL Workbench.

MySQL Workbench es un entorno excelente para el tratamiento de la Base de Datos, te permite realizar cualquier operación sobre tablas desde la interfaz gráfica además de dar la posibilidad de crear ficheros SQL donde se realizan las consultas. También te permite visualizar la estructura de la Base de Datos creada además de otras funciones.

En este proyecto será muy útil al permitir importar los datos para una tabla a partir de ficheros CSV, documentos que generaremos tras la extracción de datos de las APIs.

5.4.2 Integración del SGBD en Nuestro Sistema

La Base de Datos será lanzada localmente en el puerto 3306. Nuestro Backend se conectará a ella a través de este puerto y las credenciales configuradas. Esto permitirá que se realicen consultas desde este último y que la Base de Datos devuelva los datos solicitados, es decir, desde el Backend se realizarán tan solo operaciones de obtención de datos (SELECT).

Las operaciones de añadir o actualizar (INSERT, UPDATE) se realizan manualmente tras la importación de archivos CSV (INSERT) o a través del propio entorno MySQL Workbench a través de consultas.



5.4.3 Diagrama E-R

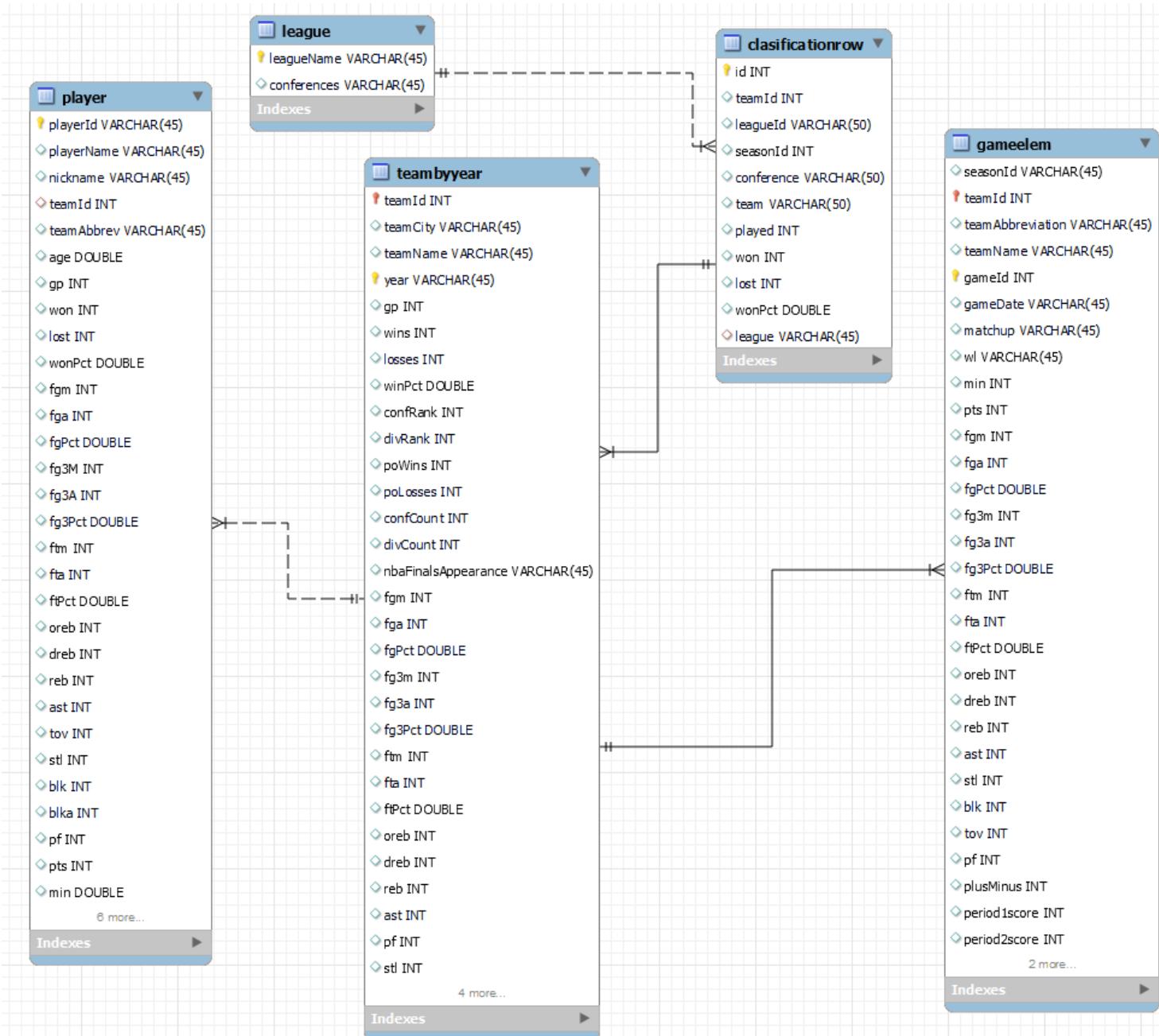


Ilustración 5-16 Diagrama Entidad Relación

Las relaciones establecidas entre las entidades de la Base de Datos, como se puede ver en [Ilustración 5-16 Diagrama Entidad Relación](#) se van a explicar a continuación, así como lo que representa cada clase

5.4.3.1 *Player*

Representa a los jugadores, almacenando toda su información, tanto global, como puede ser la edad, el peso o la altura, como estadísticas concretas de la temporada.

5.4.3.2 *TeamByYear*

Representa a los equipos, se llama así porque no solo guarda la información de la última temporada, sino que vamos a tener por cada equipo la información de sus últimas 5 temporadas.

5.4.3.3 *League*

Representa las ligas.

5.4.3.4 *ClasificationRow*

Representa cada una de las filas que aparecen en la clasificación, es decir, los datos de cada uno de los equipos que aparecerán en la clasificación.

5.4.3.5 *GameElem*

Representa cada elemento de un partido, es decir, es la referencia a un equipo concreto dentro de un partido específico, almacenando las estadísticas que definen su actuación en el mismo.

5.4.3.6 *Relaciones*

- Player/TeamByYear: relación de 1 a N, un jugador solo tiene un equipo, pero un equipo puede tener muchos jugadores.
- TeamByYear/GameElem: relación de 1 a N, es decir, un equipo puede estar en muchos partidos, pero en un partido en concreto ese equipo solo estará una vez.
- ClasificationRow /TeamByYear: relación de 1 a N, ya que una clasificación tendrá varios equipos, pero un equipo solo puede estar en una clasificación.
- League/ClasificationRow: relación de 1 a N, un elemento de la clasificación será de una sola liga, pero una liga tendrá varios elementos de una clasificación.

5.5 Diseño de la Interfaz

La interfaz de esta aplicación ha sido realizada a través de los ficheros HTML y CSS de Angular, además de utilizar Bootstrap para facilitar este trabajo y poder hacer la aplicación responsive.

Vamos a visualizar ahora las distintas pantallas definitivas que componen la aplicación. Hay que mencionar que todo lo que se vea en este apartado es una evolución de lo explicado en [5.7 Análisis de Interfaces de Usuario](#) siendo ya este el resultado final de la interfaz de usuario.

En primer lugar, la ventana de inicio, con la pequeña explicación sobre la aplicación y con el menú de navegación para poder acceder a las demás pantallas.

Professional Basketball Analytics

Inicio Clasificación Jugadores Equipos Partidos

Professional Basketball Analytics

Bienvenido al sitio web en el que podrás consultar las estadísticas en diferentes ámbitos del baloncesto, desde la clasificación de la última temporada, hasta la actuación de jugadores y equipos en la misma.

Podrás acceder también a un análisis más detallado de equipos, jugadores y partidos que han intervenido en la temporada de la NBA.

El acceso a toda esta información se consigue a través de la barra de navegación que se encuentra en la parte superior de la pantalla. Actualmente te encuentras en el 'inicio', pero voy a explicarte la información que puedes visualizar en cada sección.

En la sección de 'clasificación', como el propio nombre indica, podrás acceder a las posiciones de los equipos en la temporada 2022-2023, separando dicha clasificación en función de la conferencia este y oeste en que se dividen los equipos de la nba.

Copyrights by Alejandro Galán Freire

Ilustración 5-17 Pantalla Inicio

Ahora procedemos a explicar la ventana de clasificación. Como se puede ver en XXX se dispone de un botón desplegable para seleccionar la liga de la que se quiere visualizar la clasificación, que por defecto será la NBA. Se muestra una tabla con el posicionamiento de los equipos en esa liga además de aportar una serie de estadísticas.

Professional Basketball Analytics

Inicio Clasificación Jugadores Equipos Partidos

Seleccione la liga

NBA ACB EAST

Posición	Nombre	V	D	P	V%
1	 Milwaukee	58	24	82	0.707
2	 Boston	57	25	82	0.695
3	 Philadelphia	54	28	82	0.659
4	 Cleveland	51	31	82	0.622
5	 New York	47	35	82	0.573
6	 Brooklyn	45	37	82	0.540

Copyrights by Alejandro Galán Freire

Ilustración 5-18 Ventana Clasificación

Continuamos con la sección de jugadores, donde aparecen todas las tarjetas de todos los jugadores inicialmente. En la parte superior se dispone del buscador que nos permite filtrar los jugadores que aparecen por nombre o equipo.

Professional Basketball Analytics

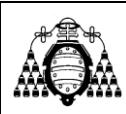
Inicio Clasificación Jugadores Equipos Partidos

[leb]

 Maxi Kleber País Altura Peso Experiencia Número Posición Equipo Germany 1.86 108.86 5 42 A DAL	 LeBron James País Altura Peso Experiencia Número Posición Equipo USA 2.1 113.4 19 6 A LAL	 Caleb Martin País Altura Peso Experiencia Número Posición Equipo USA 1.98 92.99 3 16 A MIA
 Caleb Houston País Altura Peso Experiencia Número Posición Equipo Canada 2.07 92.99 0 2 B/E ORL		

Copyrights by Alejandro Galán Freire

Ilustración 5-19 Pantalla Jugadores



Lo mismo sucede con los equipos, diferenciándose lógicamente la información que representa cada tarjeta.

Professional Basketball Analytics

Inicio Clasificación Jugadores Equipos Partidos

 Atlanta Hawks Estadísticas Plantilla	 Boston Celtics Estadísticas Plantilla	 Brooklyn Nets Estadísticas Plantilla
 Charlotte Hornets Estadísticas Plantilla	 Chicago Bulls Estadísticas Plantilla	 Cleveland Cavaliers Estadísticas Plantilla
 Dallas Mavericks Estadísticas Plantilla	 Denver Nuggets Estadísticas Plantilla	 Detroit Pistons Estadísticas Plantilla

Copyrights by Alejandro Galán Freire

Ilustración 5-20 Pantalla Equipos

En la pantalla de partidos la peculiaridad es que en vez de un buscador se dispone de un calendario que nos permite seleccionar la fecha de la que queremos ver los partidos.

Ilustración 5-21 Pantalla Partidos

Ahora pasamos con el detalle de los jugadores, pantalla accesible al presionar sobre el nombre de un jugador desde su tarjeta o desde la plantilla de un equipo, donde se visualizan los datos del jugador en forma de tablas y gráficos.

Ilustración 5-22 Pantalla Detalle Jugador



Lo mismo sucede con la ventana de detalle de los equipos, accesible desde el nombre del equipo en su tarjeta o desde el botón de estadísticas que esta presenta, y con la ventana de detalle de partidos.

Professional Basketball Analytics

Inicio Clasificación Jugadores Equipos Partidos

CONFERENCIA DIVISION
Celtics EAST ATLANTICO

Estadísticas últimos 5 años

AÑO	PTS	FG_PCT	FG3_PCT	FT_PCT	REB	AST	PER	ROB	BLQ	FAL
1º	9216	0.465	0.365	0.802	3653	2155	1052	706	435	1670
2º	8183	0.461	0.364	0.801	3317	1655	995	595	406	1553
3º	8109	0.466	0.374	0.775	3186	1689	1012	556	383	1471
4º	9164	0.466	0.356	0.816	3778	2036	1118	591	478	1521
5º	9671	0.475	0.377	0.812	3717	2186	1095	521	430	1542

Progreso 5 años

Victorias Derrotas Clasificación conferencia Clasificación división

Asistencias, rebotes, bloqueos, robos y pérdidas

1º Año 2º Año 3º Año 4º Año 5º Año

Copyrights by Alejandro Galán Freire

Ilustración 5-23 Pantalla Detalle Equipo

Professional Basketball Analytics

Inicio Clasificación Jugadores Equipos Partidos

Comparación estadísticas

Equipo	PTS	FG_PCT	FG3_PCT	FT_PCT	REB	AST	PER	ROB	BLQ	FAL
Memphis Grizzlies	100	0.435	0.189	0.722	43	25	12	8	4	16
Oklahoma City Thunder	115	0.5	0.333	0.813	49	30	14	5	2	17

Puntos por cuartos

Memphis Grizzlies Oklahoma City Thunder

Asistencias, rebotes, robos, pérdidas, faltas, bloqueos

Memphis Grizzlies Oklahoma City Thunder

Porcentajes

Tiros Intentados Tiros anotados

Copyrights by Alejandro Galán Freire

Ilustración 5-24 Pantalla Detalle Partido

Los equipos tienen además en la tarjeta la sección de plantilla, que nos lleva a su correspondiente pantalla.

Professional Basketball Analytics

Inicio Clasificación Jugadores Equipos Partidos

Celtics - Plantilla

Nombre	País	Altura	Peso	Experiencia	Número	Posición	Equipo
Jaylen Brown	USA	2.01	101.15	6	7	A/AP	BOS
Malcolm Brogdon	USA	1.95	103.87	6	13	B/E	BOS
Jayson Tatum	USA	2.07	95.25	5	0	E/A	BOS
Justin Jackson	USA	2.07	99.79	5		A	BOS

Copyrights by Alejandro Galán Freire

Ilustración 5-25 Pantalla Plantilla Equipo



Y, por último, las ventanas de comparación de jugadores y equipos accesibles desde las ventanas de detalle de estos.

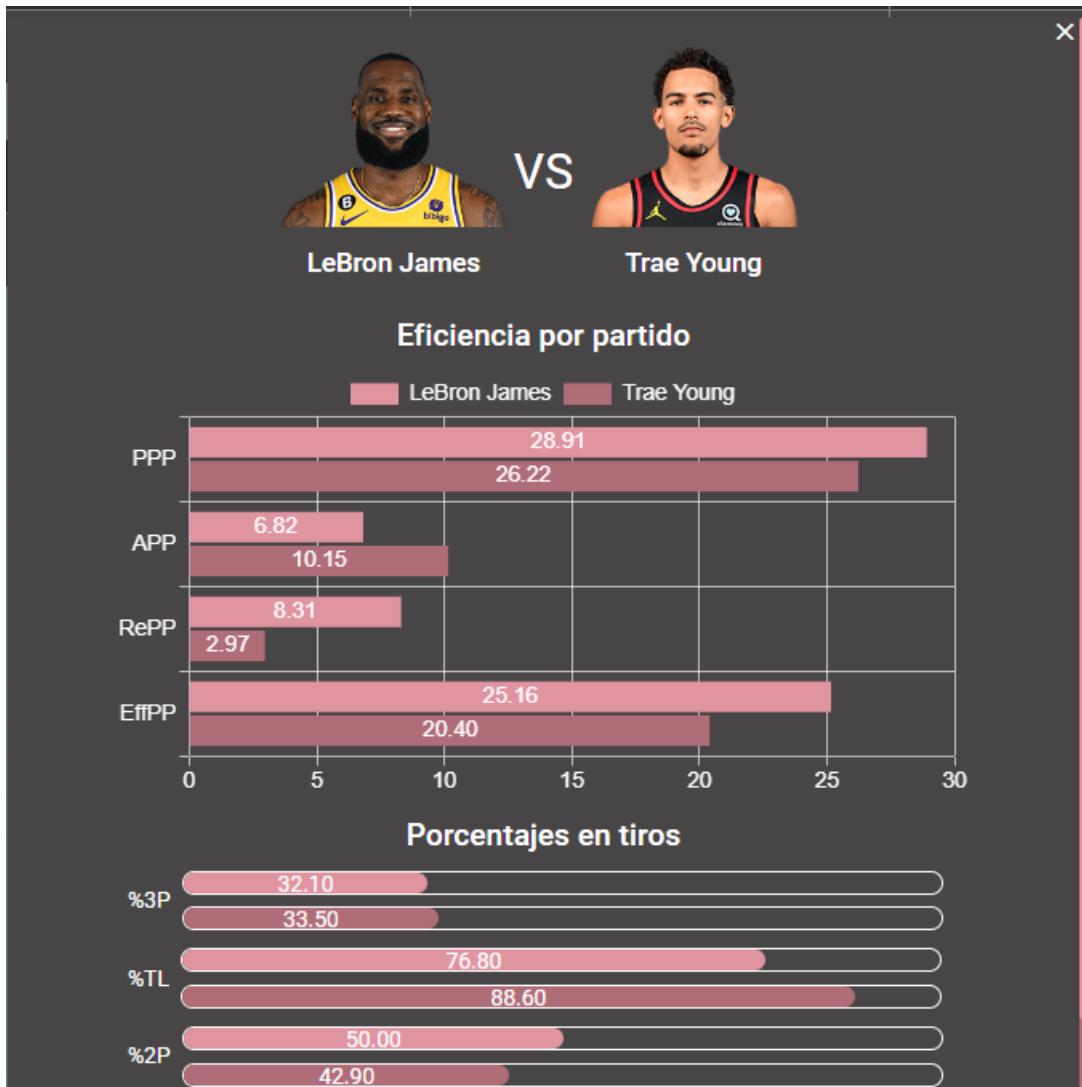


Ilustración 5-26 Pantalla Comparación Jugadores

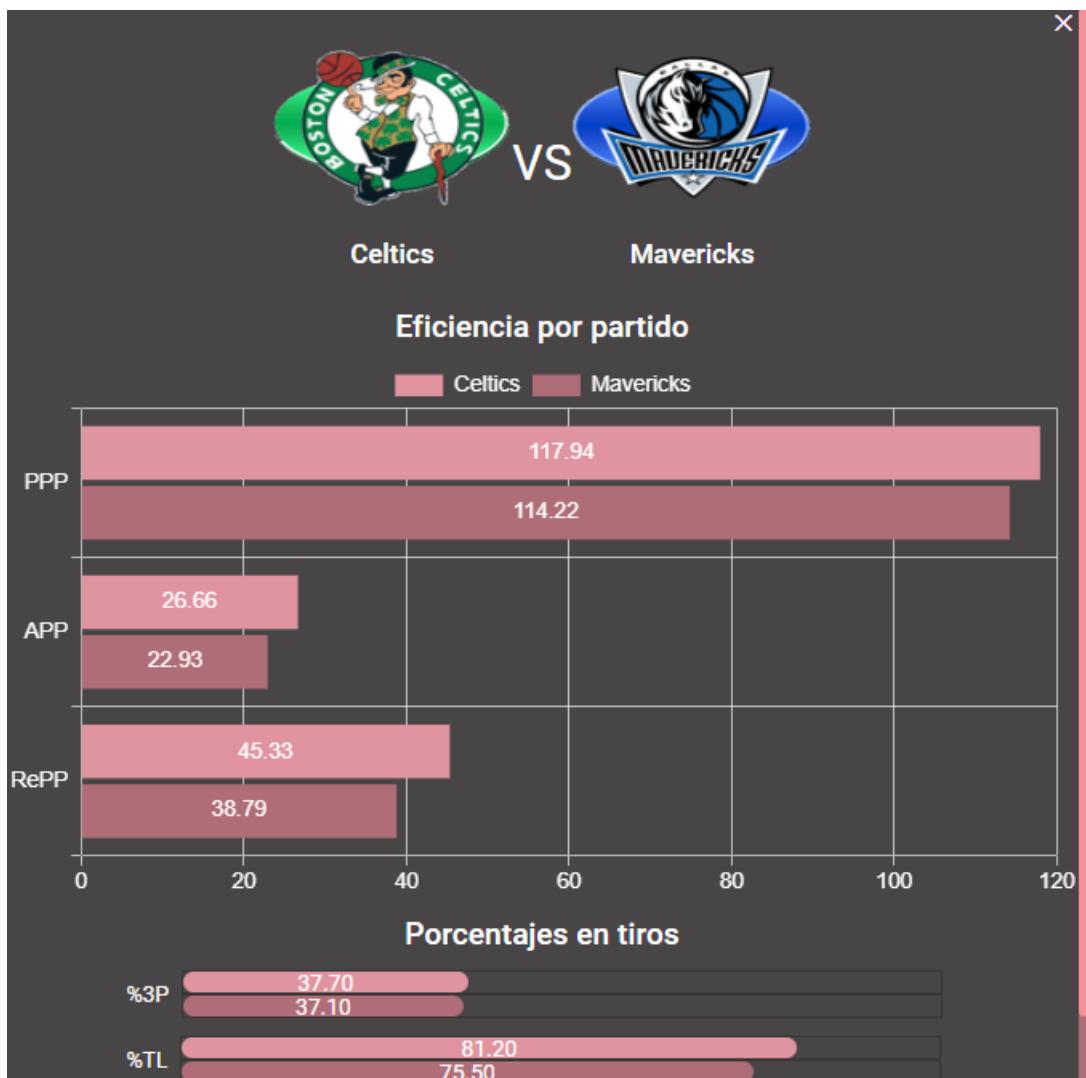
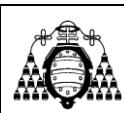


Ilustración 5-27 Pantalla Comparación Equipos

5.6 Especificación Técnica del Plan de Pruebas

Vamos a establecer el proceso técnico que se llevará a cabo en las pruebas especificadas en [4.8 Especificación del Plan de Pruebas](#).

5.6.1 Pruebas Unitarias y de Calidad

Estas pruebas son las primeras que se ejecutan para asegurarnos que cada módulo individual funciona correctamente y que podemos pasar a probarlos integrados con el resto del sistema.

Como se comenta en el apartado [Pruebas unitarias y de calidad](#) las pruebas en el frontend se realizan a través de Karma y Jasmine y en el backend a través de JUnit 4.

La forma de comprobar que estas pruebas pasan es ejecutándolas localmente, y el procedimiento llevado a cabo en caso de error simplemente es realizar una depuración para detectar el error y corregirlo para su correcto funcionamiento.

Después de todo esto lanzamos un análisis de código a través de la herramienta SonarQube que nos permite identificar malas prácticas, bugs, duplicación de código y la cobertura de este.

El procedimiento seguido con esta herramienta es ir corrigiendo los errores que nos indica y repitiendo los análisis hasta que este sea positivo.

5.6.2 Pruebas de Integración y del Sistema

Una vez finalizadas las pruebas unitarias de un módulo procedemos a probar ese módulo manualmente con el sistema lanzado, para ver si el funcionamiento es el esperado. Cuando conseguimos hacer este proceso con todos los subsistemas pasamos al siguiente nivel, que es realizar dichas pruebas manuales sobre todo el sistema para asegurar que todos los módulos se comunican correctamente y que el sistema no presenta ningún fallo.

En nuestro caso por ejemplo es de vital importancia comprobar que los datos de una pantalla a otra, como puede ser del listado de los jugadores al detalle de uno de ellos, se envían de forma correcta, o probar valores correctos e incorrectos en los campos de entrada como son los buscadores, para ver si el sistema se comporta de la manera esperada en todos los casos.

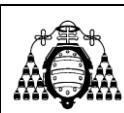
5.6.3 Pruebas de Usabilidad

Las pruebas de usabilidad son excesivamente importantes, sería como realizar las pruebas de sistema, pero con usuarios reales. Esto es clave, ya que nosotros como desarrolladores no nos damos cuenta de ciertos aspectos que cualquier otro usuario sí pudiera hacerlo, debido a que damos por hecho un montón de cosas al ser los autores de lo probado, como, por ejemplo, algo tan simple como la navegación entre las distintas ventanas diseñadas.

Por tanto, el proceso a seguir es el siguiente:

1. Realizar un cuestionario con una serie de preguntas al usuario para saber su conocimiento en la informática.
2. Plantear una serie de actividades que el usuario tendrá que hacer con nuestra aplicación.
3. Observar el responsable de las pruebas como hacen dichas tareas para apuntar ciertos aspectos rellenando un cuestionario.
4. Cuestionario de preguntar cortas para el usuario sobre la aplicación, donde podrá dar una serie de aportaciones según lo que haya podido aportar.

Vamos ahora a definir los distintos cuestionarios que serán utilizados para llevar a cabo las pruebas de usabilidad.



5.6.3.1 Preguntas de carácter general

¿Usa un ordenador frecuentemente?

1. Todos los días
2. Varias veces a la semana
3. Ocasionalmente
4. Nunca o casi nunca

¿Qué tipo de actividades realiza con el ordenador?

1. Es parte de mi trabajo o profesión
2. Lo uso básicamente para ocio
3. Solo empleo aplicaciones estilo Office
4. Únicamente leo el correo y navego ocasionalmente

¿Ha usado alguna vez software como el de esta prueba?

1. Sí, he empleado software similar
2. No, aunque si empleo otros programas que me ayudan a realizar tareas similares
3. No, nunca

¿Qué busca Vd. Principalmente en un programa?

1. Que sea fácil de usar
2. Que sea intuitivo
3. Que sea rápido
4. Que tenga todas las funciones necesarias

¿Qué escogería entre que el programa fuera con tan solo texto o que presentara elementos visuales como gráficos?

1. Texto
2. Gráficos

5.6.3.2 Actividades guiadas

El conjunto de actividades que los usuarios tendrán que realizar en la aplicación es:

1. Visualizar la clasificación de la NBA
2. Acceder al detalle del jugador Lebron James
3. Comparar a Lebron James con Trae Young
4. Acceder al detalle del equipo Golden State Warriors
5. Ver la plantilla de Golden State Warriors
6. Comparar los Golden State Warrios con Denver Nuggets
7. Ver el detalle de un partido jugado el 08/04/2023

5.6.3.3 Cuestionario para el responsable de las pruebas

El responsable deberá llenar este cuestionario mientras está observando la realización de tareas por parte del usuario:

Aspecto Observado	Notas
<i>El usuario comienza a trabajar de forma rápida por las tareas</i>	
<i>Tiempo en realizar cada tarea</i>	
<i>Preguntas</i>	
<i>Dificultades</i>	
<i>Errores</i>	

5.6.3.4 Cuestionario de preguntas cortas

Por último, se le pedirá al usuario responder las siguientes preguntas:

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca		
<i>¿Sabe dónde está dentro de la aplicación?</i>						
<i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i>						
<i>¿Le resulta sencillo el uso de la aplicación?</i>						
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca		
<i>¿Funciona cada tarea como Vd. Espera?</i>						
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>						
Calidad del Interfaz						
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado		
<i>El tipo y tamaño de letra es</i>						
<i>Los iconos e imágenes usados son</i>						
<i>Los colores empleados son</i>						
Diseño de la Interfaz	Si	No	A veces			
<i>¿Le resulta fácil de usar?</i>						
<i>¿El diseño de las pantallas es claro y atractivo?</i>						
<i>¿Cree que el programa está bien estructurado?</i>						
Observaciones						
Cualquier comentario del usuario						

Capítulo 6. Implementación del Sistema

6.1 Estándares y Normas Seguidos

Este proyecto ha seguido el patrón Modelo Vista Controlador (MVC), siendo la parte frontend la vista y el backend el modelo y el controlador. Dentro del primer subsistema implementado en Angular nos limitamos a seguir la estructura clásica de un proyecto en este lenguaje, partiendo del AppComponent y comunicándose con el resto de los componentes organizados en sus carpetas con sus correspondientes ficheros.

Por otro lado, el backend sí que sigue una estructuración con el objetivo de separar cada elemento perteneciente a un módulo, es decir, dentro de este separar aquellos componentes con responsabilidades distintas. Cada módulo, por ejemplo, el de jugadores, tiene un modelo, un controlador, un servicio y un mapeador. Ninguno de ellos conoce la implementación concreta de la clase con la que se relaciona si no que se comunican a través de interfaces. Esto nos permite que las clases sean independientes, de forma que los cambios en la implementación de los mensajes definidos por una clase no afecten a los clientes que los utilicen.

También se sigue la convención de nombres establecida para proyectos MVC, donde los controladores acaban en *Controller*, los servicios en *Service* y *ServiceImpl*, y los mapeadores en *Mapper*.

En cuanto a las pruebas unitarias, se ha considerado apto tener como mínimo un 80% de cobertura de código y menos de un 3% de repetición de código en cada subsistema principal (frontend y backend). No se permitía ninguna mala práctica (Code smell) o BUG.

6.2 Lenguajes de Programación

Descripción de los lenguajes de programación usados, su versión y distribución, los módulos o complementos de los mismos empleados, etc.

Vamos a explicar los lenguajes de programación y frameworks empleados en el desarrollo del proyecto.

6.2.1 Angular

Lenguaje utilizado para la realización del frontend. Versión 15.2.0.

Se basa en la realización de la lógica de negocio a través de *typescript*, y la generación de las ventanas con *HTML* y *CSS*. Además, se utilizaron complementos adaptables al lenguaje como *Angular Material* para la creación de componentes por defecto o ampliación de funcionalidades y *Bootstrap* para el diseño responsivo de la aplicación.

Para la realización de los gráficos se utilizó la librería *Chart.js*.

Para la realización de pruebas Angular se ha utilizado *Karma* y *Jasmine*.

Jasmine es un framework que proporciona una sintaxis clara para la realización de las pruebas unitarias del proyecto, permitiendo estructuras las mismas de la forma más adecuada además de ofrecer una amplia gama de métodos y aserciones.

Karma es el runner de las pruebas unitarias. Se encarga de ejecutar las pruebas unitarias en varios navegadores y entornos.

Si se desea más información se dispone de ella en [3.5 Angular](#).

6.2.2 SpringBoot

Java es el lenguaje utilizado para realizar la parte backend, pero se ha añadido el framework *Spring Boot* ya que simplifica la creación de aplicaciones web. Tiene integrada la herramienta de gestión de proyectos Maven, que proporciona un enfoque estructurado y automatizado para la construcción de proyectos y gestión de dependencias.

La versión de Java utilizada es la 11.

Para facilitar la interacción con la base de datos relacional se incorporó el framework *MyBatis* que proporciona una capa de persistencia que permite mapear objetos Java a sentencias SQL y viceversa. Dichas sentencias SQL se realizan en ficheros XML con una configuración específica para poder realizar el mapeo correctamente.

Para las pruebas unitarias se ha utilizado el framework JUnit, versión 4. Proporciona un conjunto de anotaciones, clases y métodos que facilitan la creación y ejecución de pruebas unitarias de manera automatizada.

Si se desea más información se dispone de ella en [3.8 SpringBoot](#).

6.2.3 MySQL

Se ha utilizado un Sistema Gestor de Base de Datos relacional, MySQL versión 8.0.32. Ha sido una opción muy cómoda, que ha permitido una integración excesivamente sencilla con el resto de las partes de la aplicación. En la sección [7.3 Herramientas y Programas Usados para el Desarrollo](#) se explicará la herramienta utilizada para gestionar dicho gestor.

Si se desea más información se dispone de ella en [3.10 MySQL](#).

6.2.4 Python

La versión utilizada de este lenguaje ha sido la 3.9.13. Se ha utilizado para el proceso de extracción de información de las APIs utilizadas, para posteriormente darle formato a los datos recogidos y almacenarlos en ficheros CSV. Para ello se utilizaron las librerías *Pandas* y *JSON*, además de la librería *HTTP* y otras necesarias para poder conectarte a los endpoint definidos por las APIs.

Si se desea más información se dispone de ella en [3.9 Python](#).

6.3 Herramientas y Programas Usados para el Desarrollo del Sistema

6.3.1 Visual Studio Code

Visual Studio Code se ha utilizado como herramienta de desarrollo de la parte frontend. Versión 1.79.2. Es una herramienta con una amplia gama de extensiones muy útiles para el desarrollo de proyectos Angular (y otros muchos) además de permitir la ejecución de comandos necesarios a través de su terminal.

También se ha usado la terminal de esta herramienta para ejecutar los comandos para la incorporación de archivos al repositorio del proyecto.

Si se desea tener más información de esta herramienta se dispone de ella en [3.13 Visual Studio Code](#).

6.3.2 Eclipse – Spring Tool Suite

Se ha utilizado Spring Tool Suite que es un entorno de desarrollo integrado (IDE) basado en Eclipse. Se utiliza la versión 4.

Este entorno está diseñado para el desarrollo de aplicaciones basadas en el framework Spring, proporcionando un conjunto de herramientas y características que facilitan el desarrollo.

Si se desea más información acerca de este entorno visita el apartado [3.14 Spring Tool Suite](#).

6.3.3 MySQL Workbench

MySQL Workbench es el entorno utilizado para la gestión de la Base de Datos. Se utiliza la versión 8.0.

Facilita mucho el trabajo al permitir la creación de tablas, así como cualquier modificación sobre ellas, o la ejecución de consultas o la importación de los datos a través de la interfaz gráfica que presenta, bastante intuitiva. También dispone de funciones no tan intuitivas como la que te permite visuales el diagrama Entidad – Relación de cualquier base de datos que tengas cargada en el entorno.

Si se desea obtener más información sobre este entorno visite [3.15 MySQL Workbench](#).

6.3.4 Spyder (Anaconda 3)

Anaconda es una distribución de Python, y es la que entre otros muchos proporciona el entorno de desarrollo Spyder, donde se programa en el lenguaje Python. La versión utilizada es la 5.2.2.

Si se desea obtener más información visite [3.16 Anaconda – Spyder](#).

6.3.5 Microsoft Excel

Por último, Excel se ha utilizado para crear y gestionar todos los ficheros CSV que contenían los datos que posteriormente serían almacenados en la base de datos. La versión utilizada ha sido la 2305.

6.3.6 SonarQube

Herramienta utilizada para analizar la calidad del código. Se lanzaba localmente en el puerto 9000 y se accedía a través de la web. La versión utilizada fue la 9.9.1.

Cada vez que se hacía algún cambio en el código se ejecutaba el comando correspondiente en el subsistema del cambio para que se realizara el análisis.

6.4 Herramientas y Programas Usados para el Desarrollo de la Documentación

6.4.1 Microsoft Word

Para la realización de toda la documentación se ha utilizado Word, versión 2305.

6.4.2 Microsoft Project

Para la realización de las planificaciones del proyecto he utilizado Microsoft Project, la versión del 2019.

Nos permite gestionar proyectos creando el conjunto de tareas que lo componen, estableciendo relaciones entre ellas como las predecesoras, asignándoles recursos y haciendo un seguimiento del avance de estas.

6.4.3 UMLetino

Ha sido la herramienta usada para la realización de todos los diagramas que aparecen en esta documentación.

Se ha usado la versión web, disponible en este enlace <https://www.umletino.com/umletino.html>.

6.5 Otras herramientas

6.5.1 GitHub

Para la gestión de la evolución del proyecto se ha utilizado la herramienta por excelencia para el control de versiones GitHub, actualizándose a través de la web para la creación de los proyectos, wikis, issues y a través de los comandos para la incorporación o actualización de los elementos almacenados en el repositorio.

6.6 Creación del Sistema

6.6.1 Documentación sistemas

El backend o REST API de este proyecto define una serie de endpoints para poder extraer los datos de la Base de Datos. Como el propio nombre lo dice, es una API de forma que es independiente del resto de módulos, definiendo una serie de mensajes que pueden ser consumidos por distintos clientes si así se desea. Es decir, el backend de este proyecto no se limita al mismo, sino que si existiera otro frontend o Webapp que quisiera consumirlo sería posible. Por ello, es fundamental la correcta documentación de esta API.

La documentación de este subsistema ha sido generada a través de la herramienta automática que presenta Eclipse, Javadoc. Los ficheros de la documentación se entregarán junto con este proyecto, en la carpeta doc y dentro de la misma la subcarpeta backend.

Para generar la documentación del proyecto frontend se ha usado la librería compodoc, también se aportan los ficheros generados junto con este proyecto, pero en este caso dentro de la subcarpeta frontend que se encuentra dentro de la carpeta doc.

6.6.2 Problemas Encontrados

6.6.2.1 *Configuración MyBatis*

Son varias las configuraciones que el framework admite y conseguir la deseada fue un proceso costoso en el que se tuvo que investigar bastante además de probar numerosas veces configuraciones ineficientes.

6.6.2.2 *Configuración SobarQube*

Sobre todo, para la parte del backend fue un proceso complejo, debido a que se tuvieron que probar varios comandos Maven hasta dar con los correctos. Proceso similar a la configuración de MyBatis.

6.6.2.3 *Despliegue*

El despliegue de todas las partes de la aplicación en la nube resultó ser un proceso imposible, pero al que se le dedicó un tiempo. Se considera una opción de mejora a futuro, pero debido al tiempo limitado del que se dispone es un proceso al que no se le puede dedicar más tiempo del ya gastado.

6.6.2.4 *Excesivos modelos*

Llegó un momento en el que el proyecto tenía excesivos modelos de objetos que se encargaban de almacenar la información. Eso provocaba una mala estructuración, con relaciones sin sentido o engorrosas.

La solución a esto fue unificar las estadísticas en clases más idóneas y de tal manera tener menos modelos en el sistema.

6.6.2.5 *Sobrecarga del AppModule*

En el frontend el esqueleto inicial te viene con este módulo como raíz. Si el desarrollador lo desea puede utilizarlo para todas la configuración de los componentes del proyecto, pero cuando este crece se vuelve incontrolable y con una estructura incorrecta.

La solución para este problema ha sido dividir este AppModule en más módulo, creando uno por cada módulo del sistema que tuviera más de una clase que relacionar.

6.6.2.6 *Mala estructuración de la Base de Datos*

En un inicio la base de datos no estaba bien estructurada, simplemente se creaban entidades sin ningún tipo de relación con el objetivo de que guardaran la información necesaria.

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 132 de 203
Alejandro Galán Freire UO277346@uniovi.es				

Posteriormente se empezaron a relacionar las entidades según lo que era lógico.

6.6.2.7 *Conocimiento escaso*

Uno de los problemas más importantes ha sido el poco conocimiento en las tecnologías utilizadas. Este era un proyecto que tenía ganas de desarrollar debido a la curiosidad por el desarrollo web y lleva un gran proceso de aprendizaje en lenguajes como Python o Angular.

La única solución a esto ha sido invertir tiempo en ello para aprender.



Capítulo 7. Desarrollo de las Pruebas

7.1 Pruebas Unitarias y de Calidad

Llegados a este punto, todas las pruebas definidas en [5.8.1 Pruebas unitarias y de calidad](#) obtienen el resultado esperado. La realidad es que no han sido muchos los problemas que han surgido.

Hay que destacar dos, y ambos son de configuración del sistema.

El primero pertenece a la configuración del backend, y es que teníamos que mockear ciertos datos como por ejemplo la conexión entre el service y el mapper, ya que el mapper es una simple interfaz que no se podía instanciar. Era necesario entonces añadir esta línea en la configuración de la clase de pruebas:

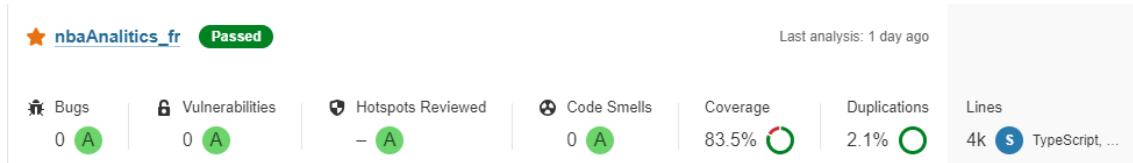
```
@MockitoSettings(strictness = Strictness.LENIENT)
```

Si no se añadía lo que sucedía es que detectaba el mapper como null y no teníamos solución al tratarse de una interfaz. Esta era, finalmente, la solución.

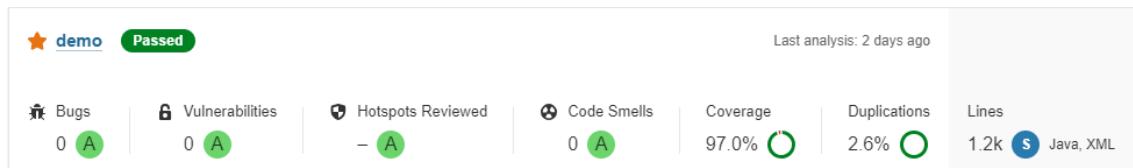
El segundo problema consistía en criterios de configuración en las clases concretas de pruebas en el frontend. No era nada técnico, simplemente las importaciones que se tenían que realizar para poder ejecutar las pruebas, que eran tantas como la clase probada tenía. Esto era un aspecto que nos generaba una gran cantidad de errores, pero de solución sencilla, tan solo había que importar lo necesario.

Como ya se comentó en el apartado [6.6.1 Pruebas Unitarias y de Calidad](#), una vez las pruebas pasaban localmente se ejecutaba el análisis en Sonarqube en ambos proyectos. Estos serían los resultados:

FRONTEND



BACKEND



7.2 Pruebas de Integración y del Sistema

Caso de Uso VC: Ver Clasificación	
Prueba	Resultado Esperado
Acceder a la pantalla de clasificación	El menú de navegación resalta la opción correspondiente y muestra por defecto la clasificación de la NBA
	Resultado Obtenido
	El menú de navegación resalta la opción correspondiente y muestra por defecto la clasificación de la NBA
Prueba	Resultado Esperado
Cambiamos la liga seleccionada	El sistema actualiza la tabla con la clasificación de la liga seleccionada (ACB será la única alternativa)
	Resultado Obtenido
	Problema ya que la ACB no tiene conferencias como la NBA. Acción tomada: Refactorización para que no dependiera de una liga concreta. Resultado final correcto.
Prueba	Resultado Esperado
Liga sin datos por cualquier error	Se muestra un mensaje al usuario avisando de que la liga seleccionada no tiene datos
	Resultado Obtenido
	Se muestra un mensaje al usuario avisando de que la liga seleccionada no tiene datos

Tabla 71 Resultado Pruebas Clasificación

Caso de Uso VJ, DJ, CJ: Jugadores	
Prueba	Resultado Esperado
Acceder a la pantalla de jugadores	El menú de navegación resalta la opción correspondiente y muestra por defecto las tarjetas de todos los jugadores
	Resultado Obtenido
	El menú de navegación resalta la opción correspondiente y muestra por defecto las tarjetas de todos los jugadores
Prueba	Resultado Esperado
Filtrado por nombre	El sistema muestra tan solo aquellos jugadores que coincidan con el nombre o texto introducido ya que la búsqueda comienza desde la primera letra escrita
	Resultado Obtenido
	El menú de navegación resalta la opción correspondiente y muestra por defecto las tarjetas de todos los jugadores
Prueba	Resultado Esperado

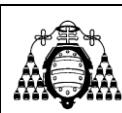


Filtrado por abreviatura de equipo	El sistema muestra tan solo aquellos jugadores que coincidan con la abreviatura del equipo o texto introducido ya que la búsqueda comienza desde la primera letra escrita
	Resultado Obtenido
	El sistema muestra tan solo aquellos jugadores que coincidan con la abreviatura del equipo o texto introducido ya que la búsqueda comienza desde la primera letra escrita
Prueba	Resultado Esperado
Filtrado por nombre no existente	Se avisa al usuario a través de un mensaje de que no existen coincidencias con esa búsqueda
	Resultado Obtenido
	Aparece la lista vacía. Acción tomada: parámetro de control que indicara si el mensaje de aviso al usuario se mostraba o no. Resultado final correcto.
Prueba	Resultado Esperado
Acceso al detalle del jugador	Se presiona sobre el título de la tarjeta de un jugador y se visualiza la pantalla de detalle del mismo
	Resultado Obtenido
	Se presiona sobre el título de la tarjeta de un jugador y se visualiza la pantalla de detalle del mismo
Prueba	Resultado Esperado
Interacción con gráficas en el detalle del jugador	Las gráficas muestran datos y son interactivas
	Resultado Obtenido
	Las gráficas muestran datos y son interactivas
Prueba	Resultado Esperado
Buscar jugador para comparar	Se muestran todos los jugadores cuyo nombre coincide con el texto introducido en el buscador de comparación de jugadores
	Resultado Obtenido
	Se muestran todos los jugadores cuyo nombre coincide con el texto introducido en el buscador de comparación de jugadores
Prueba	Resultado Esperado
Jugador buscado para comparar no existe	Se avisa al usuario a través de un mensaje de que no existen coincidencias con esa búsqueda
	Resultado Obtenido
Prueba	Resultado Esperado
Comparar jugador	Se muestra un diálogo con gráficas comparativas entre el jugador del detalle y el seleccionado en el buscador de comparación
	Resultado Obtenido
	Se muestra un diálogo con gráficas comparativas entre el jugador del detalle y el seleccionado en el buscador de comparación
Prueba	Resultado Esperado

Interacción con gráficas en el diálogo de comparación de jugadores	Las gráficas muestran datos y son interactivas
	Resultado Obtenido
	Las gráficas muestran datos y son interactivas

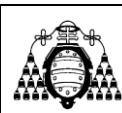
Tabla 72 Resultado Pruebas Jugadores

Caso de Uso VE, PE, DE, CE: Equipos	
Prueba	Resultado Esperado
Acceder a la pantalla de equipos	El menú de navegación resalta la opción correspondiente y muestra por defecto las tarjetas de todos los equipos
	Resultado Obtenido
	El menú de navegación resalta la opción correspondiente y muestra por defecto las tarjetas de todos los equipos
Prueba	Resultado Esperado
Filtrado por nombre	El sistema muestra tan solo aquellos equipos que coincidan con el nombre o texto introducido ya que la búsqueda comienza desde la primera letra escrita
	Resultado Obtenido
	El sistema muestra tan solo aquellos equipos que coincidan con el nombre o texto introducido ya que la búsqueda comienza desde la primera letra escrita
Prueba	Resultado Esperado
Filtrado por nombre no existente	Se avisa al usuario a través de un mensaje de que no existen coincidencias con esa búsqueda
	Resultado Obtenido
	Se avisa al usuario a través de un mensaje de que no existen coincidencias con esa búsqueda
Prueba	Resultado Esperado
Acceso a la plantilla de equipo	Se muestra la plantilla del equipo al presionar la opción correspondiente en la tarjeta del mismo
	Resultado Obtenido
	Se muestra la plantilla del equipo al presionar la opción correspondiente en la tarjeta del mismo
Prueba	Resultado Esperado
Acceso al detalle del jugador a través de plantilla	Se muestra la pantalla de detalle del jugador seleccionado desde la plantilla
	Resultado Obtenido
	Se muestra la pantalla de detalle del jugador seleccionado desde la plantilla pero no encuentra resultados si desde ahí se quiere comparar. Acción tomada: settear todos los jugadores al clickar un jugador sobre la plantilla.



	Resultado final correcto.
Prueba	Resultado Esperado
Acceso al detalle del equipo a través del título de la tarjeta	Se muestra la pantalla de detalle del equipo
	Resultado Obtenido
	Se muestra la pantalla de detalle del equipo
Prueba	Resultado Esperado
Acceso al detalle del equipo a través del botón estadísticas de la tarjeta	Se muestra la pantalla de detalle del equipo
	Resultado Obtenido
	Se muestra la pantalla de detalle del equipo
Prueba	Resultado Esperado
Interacción con gráficas en el detalle del equipo	Las gráficas muestran datos y son interactivas
	Resultado Obtenido
	Las gráficas muestran datos y son interactivas
Prueba	Resultado Esperado
Buscar equipo para comparar	Se muestran todos los equipos cuyo nombre coincide con el texto introducido en el buscador de comparación de equipos
	Resultado Obtenido
	Se muestran todos los equipos cuyo nombre coincide con el texto introducido en el buscador de comparación de equipos
Prueba	Resultado Esperado
Equipo buscado para comparar no existe	Se avisa al usuario a través de un mensaje de que no existen coincidencias con esa búsqueda
	Resultado Obtenido
	Se avisa al usuario a través de un mensaje de que no existen coincidencias con esa búsqueda
Prueba	Resultado Esperado
Comparar equipo	Se muestra un diálogo con gráficas comparativas entre el equipo del detalle y el seleccionado en el buscador de comparación
	Resultado Obtenido
	Se muestra un diálogo con gráficas comparativas entre el equipo del detalle y el seleccionado en el buscador de comparación
Prueba	Resultado Esperado
Interacción con gráficas en el diálogo de comparación de equipos	Las gráficas muestran datos y son interactivas
	Resultado Obtenido
	Las gráficas muestran datos y son interactivas

Tabla 73 Resultado Pruebas Equipos

**Caso de Uso VP, DP: Partidos**

Prueba	Resultado Esperado
Acceder a la pantalla de partidos	El menú de navegación resalta la opción correspondiente y muestra por defecto las tarjetas de todos los partidos con la fecha por defecto del último partido
	Resultado Obtenido
	El menú de navegación resalta la opción correspondiente y muestra por defecto las tarjetas de todos los partidos con la fecha por defecto del último partido
Prueba	Resultado Esperado
Seleccionar fecha con partidos	El sistema muestra tan solo aquellos partidos que se hayan jugado en la fecha seleccionada en el calendario
	Resultado Obtenido
	El sistema muestra tan solo aquellos partidos que se hayan jugado en la fecha seleccionada en el calendario
Prueba	Resultado Esperado
Seleccionar fecha sin partidos	El sistema muestra un mensaje al usuario avisando de que no existen partidos en la fecha seleccionada
	Resultado Obtenido
	El sistema muestra un mensaje al usuario avisando de que no existen partidos en la fecha seleccionada
Prueba	Resultado Esperado
Acceso al detalle del partido	Se muestra la pantalla de detalle del partido seleccionado
	Resultado Obtenido
	Se muestra la pantalla de detalle del partido seleccionado
Prueba	Resultado Esperado
Interacción con gráficas en el detalle del partido	Las gráficas muestran datos y son interactivas
	Resultado Obtenido
	Las gráficas muestran datos y son interactivas

Tabla 74 Resultado Pruebas Partidos

7.3 Pruebas de Usabilidad

Se han llevado a cabo las pruebas de usabilidad diseñadas en el apartado [6.6.3 Pruebas de Usabilidad](#) con tres usuarios de diferentes edades y profesiones.

7.3.1 Sesión 1

Edad: 46

Profesión: Ama de casa



En primer lugar, el usuario responde las preguntas iniciales.

¿Usa un ordenador frecuentemente?

1. Todos los días
2. **Varias veces a la semana**
3. Ocasionalmente
4. Nunca o casi nunca

¿Qué tipo de actividades realiza con el ordenador?

1. Es parte de mi trabajo o profesión
2. **Lo uso básicamente para ocio**
3. Solo empleo aplicaciones estilo Office
4. Únicamente leo el correo y navego ocasionalmente

¿Ha usado alguna vez software como el de esta prueba?

1. Sí, he empleado software similar
2. No, aunque si empleo otros programas que me ayudan a realizar tareas similares
3. **No, nunca**

¿Qué busca Vd. Principalmente en un programa?

1. Que sea fácil de usar
2. **Que sea intuitivo**
3. Que sea rápido
4. Que tenga todas las funciones necesarias

¿Qué escogería entre que el programa fuera con tan solo texto o que presentara elementos visuales como gráficos?

1. Texto
2. **Gráficos**

A continuación, procede a realizar las actividades establecidas para estas pruebas y definidas en [6.6.3.2 Actividades guiadas](#) mientras que el responsable de las pruebas le observa.

Estas son las anotaciones que el responsable ha tomado.

Aspecto Observado	Notas
<i>El usuario comienza a trabajar de forma rápida por las tareas</i>	Sí
<i>Tiempo en realizar cada tarea</i>	Generalmente rápido
<i>Preguntas</i>	¿Dónde está la sección de comparar?
<i>Dificultades</i>	Actividad de comparación de jugadores
<i>Errores</i>	No

Y por último se le aporta este cuestionario para que responda y de sus aportaciones.



Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Sabe dónde está dentro de la aplicación?	X			
¿Existe ayuda para las funciones en caso de que tenga dudas?			X	
¿Le resulta sencillo el uso de la aplicación?		X		
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Funciona cada tarea como Vd. Espera?	X			
¿El tiempo de respuesta de la aplicación es muy grande?				X
Calidad del Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
El tipo y tamaño de letra es	X			
Los iconos e imágenes usados son	X			
Los colores empleados son	X			
Diseño de la Interfaz	Si	No	A veces	
¿Le resulta fácil de usar?	X			
¿El diseño de las pantallas es claro y atractivo?	X			
¿Cree que el programa está bien estructurado?	X			
Observaciones				
Creo que sería necesaria una explicación de dónde se encuentra la funcionalidad de comparar los jugadores o equipos. Si no es eso, al menos poner un sección propia para ello en el menú.				

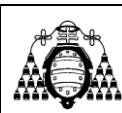
7.3.2 Sesión 2

Edad: 22

Profesión: Ingeniero informático

En primer lugar, el usuario responde las preguntas iniciales.

¿Uso un ordenador frecuentemente?
1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca
¿Qué tipo de actividades realiza con el ordenador?
1. Es parte de mi trabajo o profesión 2. Lo uso básicamente para ocio 3. Solo empleo aplicaciones estilo Office



4. Únicamente leo el correo y navego ocasionalmente

¿Ha usado alguna vez software como el de esta prueba?

1. Sí, he empleado software similar
2. No, aunque si empleo otros programas que me ayudan a realizar tareas similares
3. No, nunca

¿Qué busca Vd. Principalmente en un programa?

1. Que sea fácil de usar
2. Que sea intuitivo
3. Que sea rápido
4. Que tenga todas las funciones necesarias

¿Qué escogería entre que el programa fuera con tan solo texto o que presentara elementos visuales como gráficos?

1. Texto
2. Gráficos

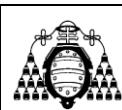
A continuación, procede a realizar las actividades establecidas.

Estas son las anotaciones que el responsable ha tomado.

Aspecto Observado	Notas
El usuario comienza a trabajar de forma rápida por las tareas	Sí
Tiempo en realizar cada tarea	Muy rápido
Preguntas	Se puede volver atrás desde el detalle del jugador
Dificultades	Ninguna
Errores	No

Y por último se le aporta este cuestionario para que responda y de sus aportaciones.

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Sabe dónde está dentro de la aplicación?	X			
¿Existe ayuda para las funciones en caso de que tenga dudas?	X			
¿Le resulta sencillo el uso de la aplicación?	X			
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Funciona cada tarea como Vd. Espera?	X			
¿El tiempo de respuesta de la aplicación es muy grande?				X
Calidad del Interfaz				



Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
<i>El tipo y tamaño de letra es</i>	X			
<i>Los iconos e imágenes usados son</i>	X			
<i>Los colores empleados son</i>		X		
Diseño de la Interfaz		Si	No	A veces
<i>¿Le resulta fácil de usar?</i>	X			
<i>¿El diseño de las pantallas es claro y atractivo?</i>	X			
<i>¿Cree que el programa está bien estructurado?</i>	X			
Observaciones				
Cuando están cargando los jugadores por ejemplo muestra al principio un mensaje de que no es posible realizar la petición cuando realmente está cargando, se debería cambiar ese mensaje. Por otro lado, quizás es buena idea poner una flecha de "atrás" en alguna parte de la aplicación.				

7.3.3 Sesión 3

Edad: 14

Profesión: Estudiante de secundaria

En primer lugar, el usuario responde las preguntas iniciales.

¿Usa un ordenador frecuentemente?
1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca
¿Qué tipo de actividades realiza con el ordenador?
1. Es parte de mi trabajo o profesión 2. Lo uso básicamente para ocio 3. Solo empleo aplicaciones estilo Office 4. Únicamente leo el correo y navego ocasionalmente
¿Ha usado alguna vez software como el de esta prueba?
1. Sí, he empleado software similar 2. No, aunque si empleo otros programas que me ayudan a realizar tareas similares 3. No, nunca
¿Qué busca Vd. Principalmente en un programa?
1. Que sea fácil de usar 2. Que sea intuitivo 3. Que sea rápido

4. Que tenga todas las funciones necesarias

¿Qué escogería entre que el programa fuera con tan solo texto o que presentara elementos visuales como gráficos?

1. Texto
2. **Gráficos**

A continuación, procede a realizar las actividades establecidas.

Estas son las anotaciones que el responsable ha tomado.

Aspecto Observado	Notas
<i>El usuario comienza a trabajar de forma rápida por las tareas</i>	Sí
<i>Tiempo en realizar cada tarea</i>	Muy rápido
<i>Preguntas</i>	¿Dónde está el apartado de comparar?
<i>Dificultades</i>	Ninguna
<i>Errores</i>	No

Y por último se le aporta este cuestionario para que responda y de sus aportaciones.

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca		
<i>¿Sabe dónde está dentro de la aplicación?</i>	X					
<i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i>		X				
<i>¿Le resulta sencillo el uso de la aplicación?</i>	X					
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca		
<i>¿Funciona cada tarea como Vd. Espera?</i>	X					
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				X		
Calidad del Interfaz						
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado		
<i>El tipo y tamaño de letra es</i>	X					
<i>Los iconos e imágenes usados son</i>	X					
<i>Los colores empleados son</i>	X					
Diseño de la Interfaz	Si	No	A veces			
<i>¿Le resulta fácil de usar?</i>	X					
<i>¿El diseño de las pantallas es claro y atractivo?</i>	X					
<i>¿Cree que el programa está bien estructurado?</i>	X					
Observaciones						
Me gusta mucho la aplicación, lo único que me costó un poco es el apartado de comparar. Una vez que sabes dónde está es fácil pero al principio te puedes equivocar.						

7.3.4 Conclusiones

Como conclusiones nos damos cuenta de que el apartado de comparar debe mejorar un poco. Lo que se va a hacer es poner en el inicio la parte explicativa sobre esta funcionalidad, ya que añadir otra sección sería trabajo para futuro.

Por otro lado, la recomendación de la segunda sesión acerca del mensaje de cargando es muy adecuada, también debería mejorarse para el futuro.

En cuanto a los colores, aunque 2/3 hayan marcado la opción de “muy adecuados” considero que podrían mejorar en las gráficas, ya que algunas tienen tantos datos que no se sabía exactamente que colores combinar. Haría falta tener en cuenta una paleta de colores más amplia.

Según la elección de los usuarios de si prefieren la representación de la información en texto o en gráficos se puede considerar un acierto la representación de los datos de esta aplicación.

En definitiva, el uso de la aplicación no causa problemas, presenta aspectos de mejora, pero no impiden la correcta usabilidad de la aplicación, los tiempos de respuesta son aceptables, es fácil de usar y también intuitiva.

Capítulo 8. 8 Manuales de Sistema

8.1 Manual de Ejecución

Este proyecto está compuesto por distintos sistemas que deben arrancarse de manera independiente. Estos sistemas son la Base de Datos, el backend y el frontend.

El procedimiento es el siguiente, sin ser necesario seguir un orden concreto ya que todos han de estar lanzados para que la aplicación funcione correctamente:

Lanzar localmente la Base de Datos

La herramienta utilizada para ello es MySQL Workbench. Aquí tenemos una instancia de nuestra Base de Datos en el puerto 3306:

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 145 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	---

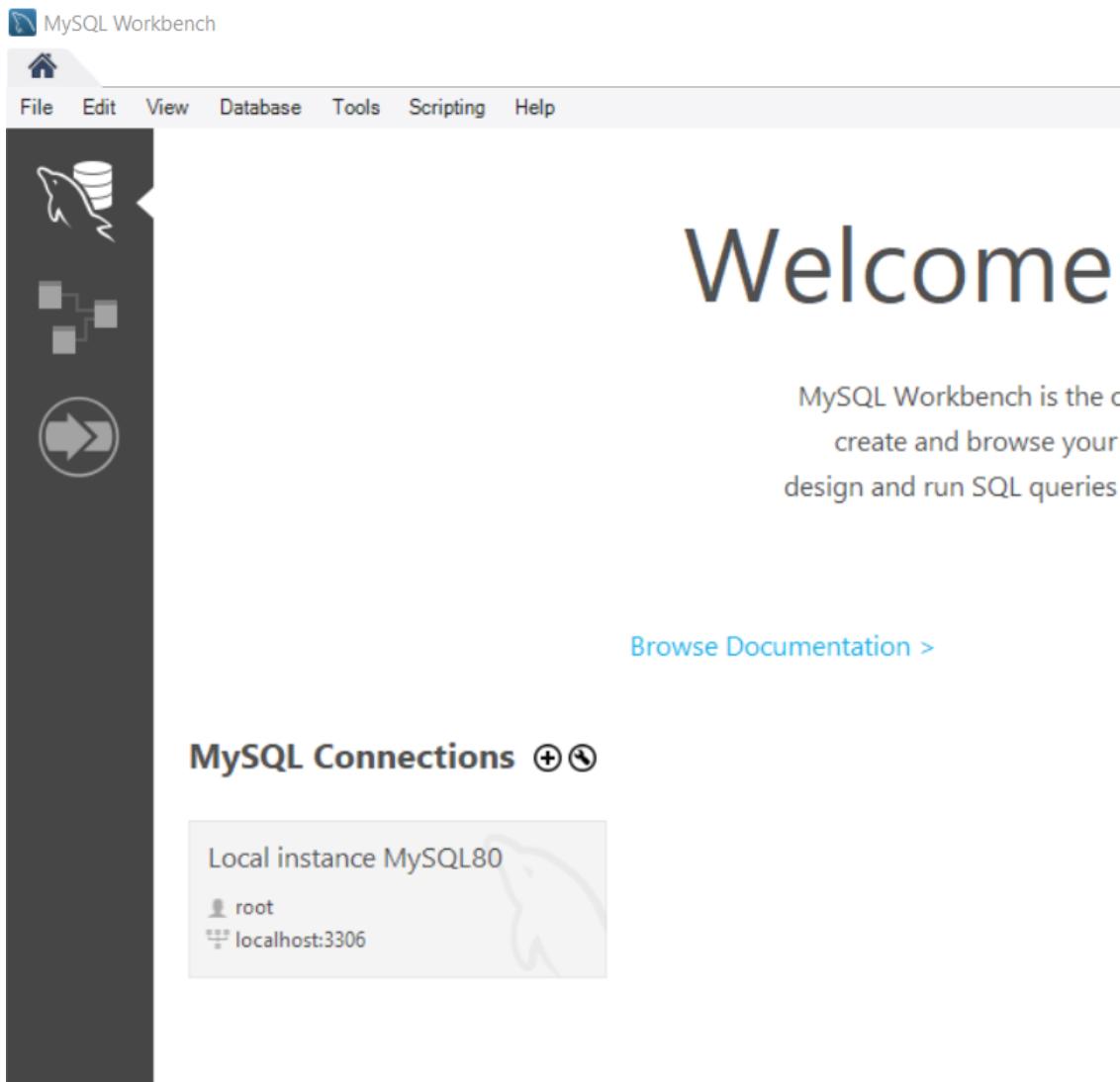


Ilustración 8-1 Lanzamiento Base de Datos

Presionamos sobre la instancia y la Base de Datos se ejecuta, quedando está disponible para trabajar con ella:

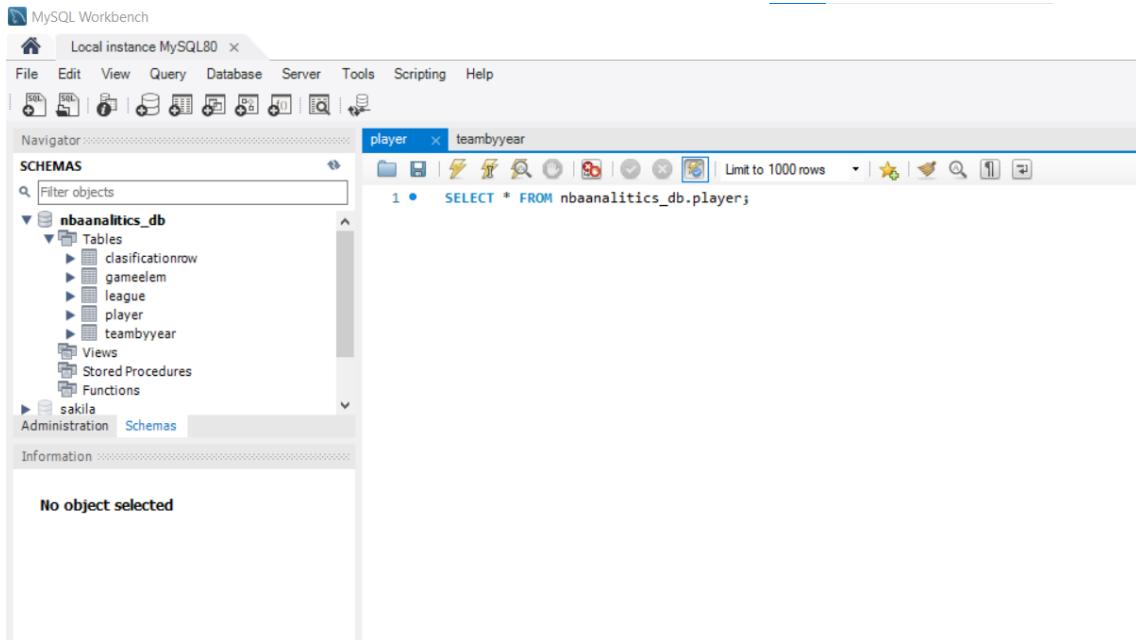


Ilustración 8-2 Base de Datos en ejecución

Lanzar localmente el Backend

Para lanzar el backend necesitamos la herramienta Spring Tool Suite. En ella cargamos el proyecto y lo ejecutamos como una aplicación de Spring Boot.

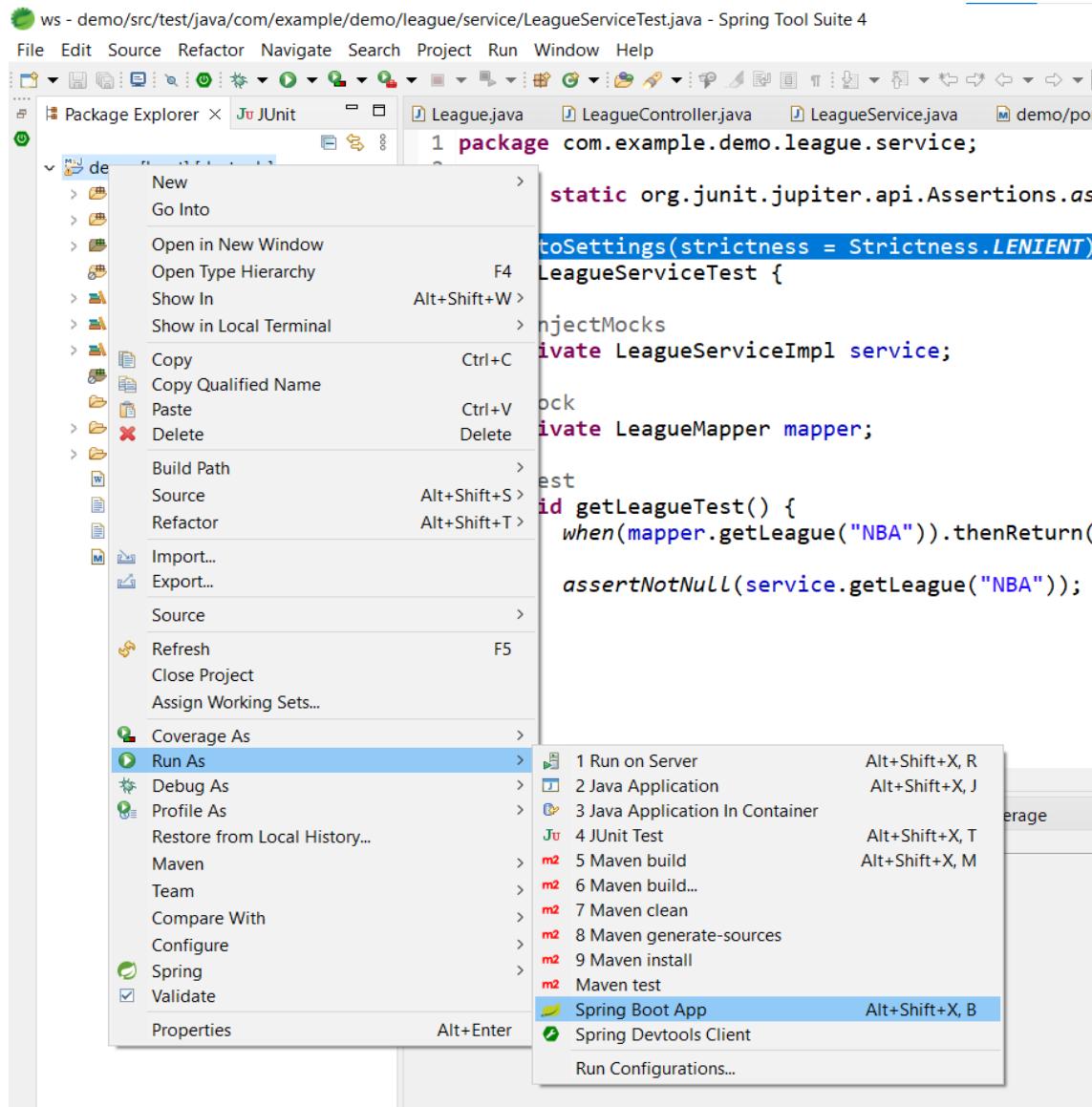
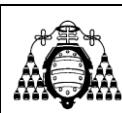


Ilustración 8-3 Ejecución del backend

En caso de que no haya errores la consola tendrá que aparecer de esta manera:

```
2023-06-28T18:30:03.182+02:00 INFO 7364 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scan
2023-06-28T18:30:03.709+02:00 INFO 7364 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080
2023-06-28T18:30:03.721+02:00 INFO 7364 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-06-28T18:30:03.721+02:00 INFO 7364 --- [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.62]
2023-06-28T18:30:03.812+02:00 INFO 7364 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApp
2023-06-28T18:30:03.813+02:00 INFO 7364 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialized [Root WebApplicationContext]
2023-06-28T18:30:04.035+02:00 INFO 7364 --- [ restartedMain] com.mysql.jdbc.Driver : This is deprecated. The new driver class is 'com.mysql.cj.jdbc.Driver'. The driver is automatically registered.
2023-06-28T18:30:04.039+02:00 INFO 7364 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2023-06-28T18:30:04.039+02:00 WARN 7364 --- [ restartedMain] c.zaxxer.hikari.util.DriverDataSource : Registered driver with driverClassName=HikariDriver
2023-06-28T18:30:04.570+02:00 INFO 7364 --- [ restartedMain] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection com.zaxxer.hikari.HikariDataSource
2023-06-28T18:30:04.572+02:00 INFO 7364 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2023-06-28T18:30:04.640+02:00 INFO 7364 --- [ restartedMain] o.h.b.dialect.Dialect : HHH000204: Processing PersistenceUnitProperties
2023-06-28T18:30:04.735+02:00 INFO 7364 --- [ restartedMain] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.2.1.Final
2023-06-28T18:30:05.341+02:00 INFO 7364 --- [ restartedMain] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
2023-06-28T18:30:05.669+02:00 INFO 7364 --- [ restartedMain] o.h.e.t.j.p.JtaPlatformInitiator : HHH000498: Using JtaPlatform implementation
2023-06-28T18:30:05.680+02:00 INFO 7364 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'Default'
2023-06-28T18:30:06.012+02:00 WARN 7364 --- [ restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, by default, JPA queries executed within this transaction might be executed within 'AUTOCOMMIT' mode.
2023-06-28T18:30:06.373+02:00 INFO 7364 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 3571
2023-06-28T18:30:06.413+02:00 INFO 7364 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http)
2023-06-28T18:30:06.421+02:00 INFO 7364 --- [ restartedMain] com.example.demo.DemoApplication : Started DemoApplication in 5.138 seconds (JVM running for 6.138)
```

Ilustración 8-4 Consola del backend tras ejecución

Aquí se puede observar que el backend está lanzado en el puerto 8080.

Lanzar localmente el Frontent

Por último, es necesario tener lanzado también el frontend. En este caso, utilizamos la herramienta Visual Studio Code en la que tenemos que cargar nuestro proyecto de Angular y ejecutar el comando 'npm start' definido en la sección 'scripts' del package.json del proyecto:

The screenshot shows the Visual Studio Code interface. On the left, the package.json file is open, showing the configuration for the 'scripts' section, which includes 'ng serve' for the 'start' command. On the right, a terminal window is running the command 'npm start'. The output shows the Angular build process, including the generation of browser bundles and the start of the development server on port 4200.

```

package.json
package.json > {} dependencies
1  {
2    "name": "nba-analitics-fr",
3    "version": "0.0.0",
4    "scripts": {
5      "ng": "ng",
6      "start": "ng serve",
7      "build": "ng build",
8      "watch": "ng build --watch --configuration development",
9      "test": "ng test --no-watch --code-coverage",
10     "sonar": "sonar-scanner"
11   },
12   "private": true
13 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS F:\4ºCarrera\SEGUNDO CUATRI\TFG\repositorio\Proyecto\frontend\nbaAnalitics_fr> npm start
> nba-analitics-fr@0.0.0 start
> ng serve

✓ Browser application bundle generation complete.

Initial Chunk Files | Names      | Raw Size
vendor.js           | vendor     | 4.93 MB
styles.css, styles.js | styles    | 730.42 kB
main.js             | main       | 358.85 kB
polyfills.js        | polyfills  | 314.28 kB
scripts.js          | scripts    | 166.37 kB
runtime.js          | runtime    | 6.53 kB

| Initial Total | 6.47 MB

Build at: 2023-06-28T16:35:15.371Z - Hash: 290f1848dc364522 - Time: 25469ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.
✓ Browser application bundle generation complete.

6 unchanged chunks

Build at: 2023-06-28T16:35:19.118Z - Hash: 290f1848dc364522 - Time: 3097ms

✓ Compiled successfully.

```

Ilustración 8-5 Ejecución del frontend

Conectarte mediante un explorador al frontend

Una vez están lanzados todos los sistemas lo único que tenemos que hacer para usar la app es conectarnos a través de un navegador al puerto 4200 de la siguiente manera:

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 149 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	---

localhost:4200

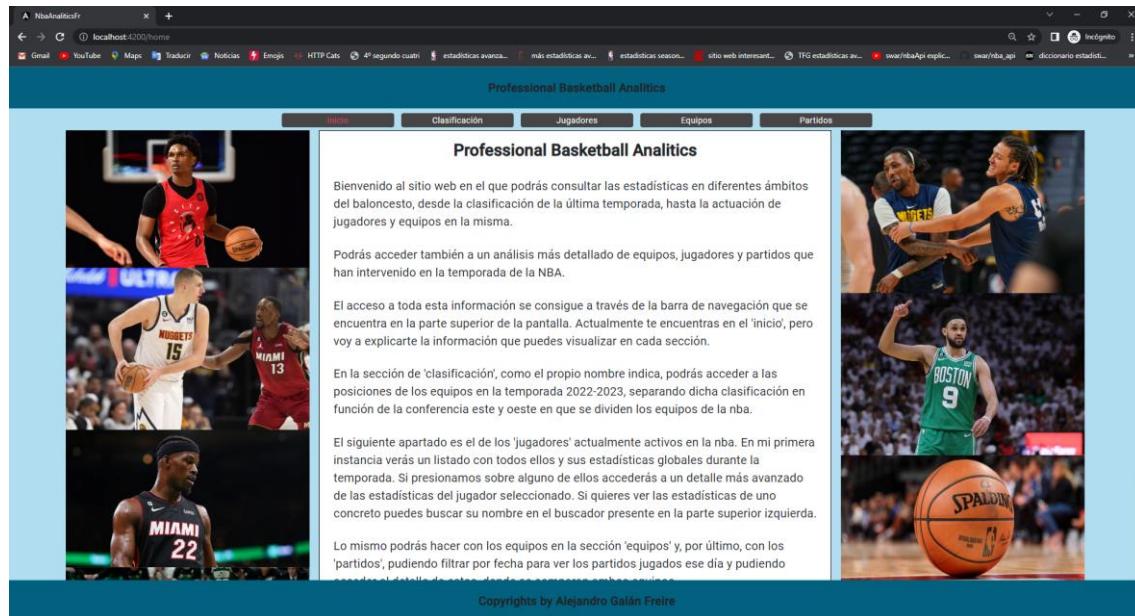


Ilustración 8-6 Acceso a la app a través del puerto del frontend

Esta es la manera de ejecutar la aplicación.

8.2 Manual de Usuario

El manual de usuario es algo muy importante debido a que es el documento que servirá a los usuarios de nuestro sistema para saber cómo funciona cada una de las partes de nuestra aplicación. Debemos pues describir cómo funcionan todas las opciones de la misma, que parámetros tiene, que cosas debemos hacer para que todas las operaciones funcionen correctamente y cualquier otro aspecto que consideremos oportuno para explicar el funcionamiento del sistema.

No debemos escatimar detalles en este manual ya que es la herramienta para que los usuarios comprendan nuestro sistema. También debemos hacer el mayor uso posible de capturas de pantalla para mejorar nuestras explicaciones.

Una vez la aplicación ha sido lanzada como se explica en el apartado anterior se puede empezar a interaccionar con la misma. A continuación, se van a explicar las diferentes pantallas y qué funcionalidades se pueden hacer con cada una de ellas, comenzando desde la pantalla inicial del sitio.

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 150 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	---

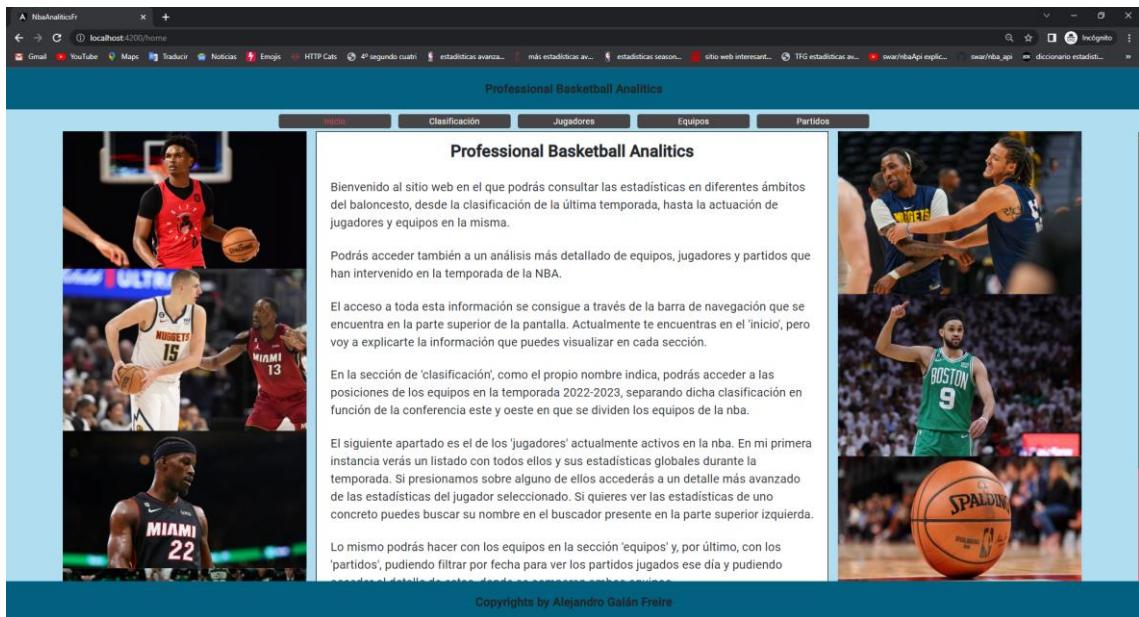


Ilustración 8-7 Pantalla de inicio

Esta es la primera pantalla que aparece al iniciar la aplicación. Es posible navegar hacia cualquier otra ventana accesible a través del menú ubicado en la parte superior en el centro:

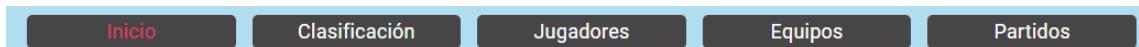


Ilustración 8-8 Menú de navegación

Si navegamos hacia la primera opción (Clasificación) navegamos hacia esta pantalla:

Professional Basketball Analitics

Inicio
Clasificación
Jugadores
Equipos
Partidos

Seleccione la liga

EAST

Posición	Nombre	V	D	P	V%
1	 Atlanta	41	41	82	0.5
2	 Boston	57	25	82	0.695
3	 Cleveland	51	31	82	0.622
4	 Chicago	40	42	82	0.488
5	 Miami	44	38	82	0.537

Copyrights by Alejandro Galán Freire

Ilustración 8-9 Pantalla Clasificación

Por defecto, visualizaremos la clasificación de la NBA, en sus distintas conferencias, pero podemos también visualizar la clasificación de la ACB seleccionando esta liga en el botón desplegable de arriba a la izquierda:



Ilustración 8-10 Selector de liga

La siguiente sección es la de jugadores, donde nos aparecería la siguiente ventana:

Professional Basketball Analytics

Inicio Clasificación **Jugadores** Equipos Partidos

Buscar jugador...

Clint Capela País Edad Altura Peso Experiencia Número Posición Equipo Switzerland 28 1.86 116.12 8 15 P	Bogdan Bogdanovic País Edad Altura Peso Experiencia Número Posición Equipo Serbia 30 1.98 102.06 5 13 B/E A
Dejounte Murray País Edad Altura Peso Experiencia Número Posición Equipo USA 26 1.98 81.65 5 5 B/E ATL	John Collins País Edad Altura Peso Experiencia Número Posición Equipo USA 25 2.1 102.51 5 20 AP/P ATI
Bruno Fernando	Aaron Holiday

Copyrights by Alejandro Galán Freire

Ilustración 8-11 Pantalla listado de jugadores

Aquí visualizaremos todos los jugadores en unas tarjetas que muestran sus datos generales. Es posible filtrar el jugador o jugadores que queramos a través del buscador:

lebron

LeBron James País Edad Altura Peso Experiencia Número Posición Equipo USA 38 2.1 113.4 19 6 A LAL
--

Ilustración 8-12 Jugador filtrado por búsqueda

Si presionamos sobre el nombre del jugador podremos acceder al detalle de este:



Ilustración 8-13 Pantalla de detalle de jugador

Dentro de esta pantalla además existe la posibilidad de escoger otro jugador para ser comparado con el del detalle, en el caso de la imagen, con Lebron James. Esta opción se encuentra en un buscador ubicado en la parte inferior de la pantalla:

Buscar jugador para comparar

Ilustración 8-14 Botón de comparación de jugadores

Si escribimos se mostrará un listado con todas las coincidencias encontradas:

Trae Young	Trent Forrest	Patrick Beverley	Patrick Williams	Patrick Baldwin Jr.	Trevor Hudgins
Trevelin Queen	Troy Brown Jr.	Max Strus	Trey Murphy III	Trevor Keels	Tre Mann
Montrezl Harrell	Trendon Watford	Trey Lyles	Tre Jones	Gary Trent Jr.	

Ilustración 8-15 Listado tras búsqueda

Si presionamos sobre algún jugador de la lista se desplegará el dialogo de comparación de ambos jugadores:

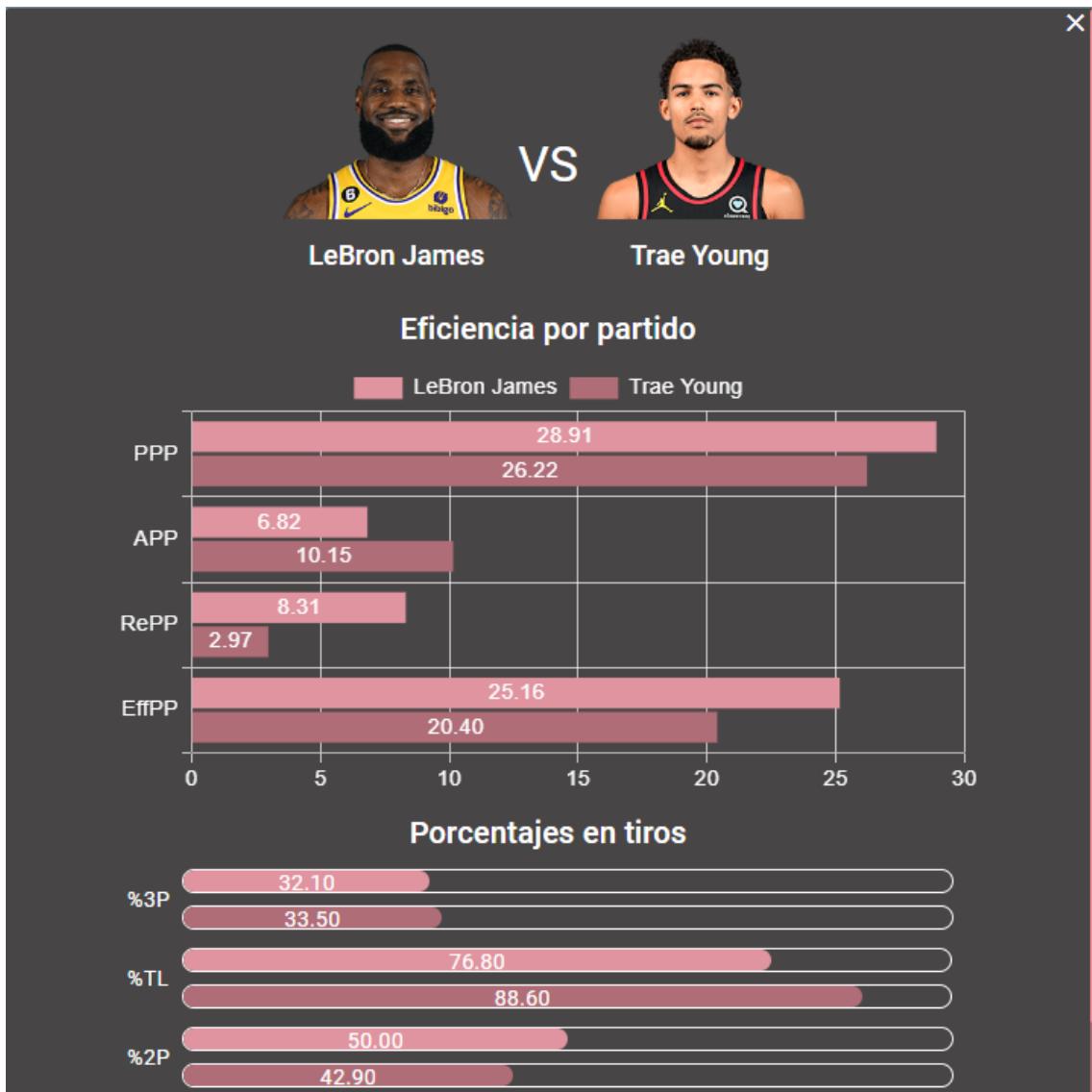


Ilustración 8-16 Pantalla comparación jugadores

Lo mismo que con los jugadores sucede con los equipos. Aporto a continuación las imágenes resultantes de hacer la misma operación partiendo de la sección de menú 'Equipos':



Ilustración 8-17 Equipos filtrados tras búsqueda

Pero antes de continuar, como se puede observar las tarjetas de los equipos permiten visualizar la plantilla de los estos. Esta opción mostrará una tabla con los jugadores activos en el equipo:

Professional Basketball Analytics								
Cleveland Cavaliers - Plantilla		Cleveland Cavaliers - Plantilla						
Nombre	País	Altura	Peso	Experiencia	Número	Posición	Equipo	
 Robin Lopez	USA	2.13	127.46	14	33	P	CLE	
 Ricky Rubio	Spain	1.89	86.18	11	13	B/E	CLE	
 Danny Green	USA	2.01	97.52	13	14	B/E	CLE	
 Raul Neto	Brazil	1.89	81.65	7	19	B/E	CLE	
 Cedi Osman	Turkey	2.04	104.33	5	16	A	CLE	
 Caris LeVert	USA	2.01	92.99	6	3	B/E	CLE	

Copyrights by Alejandro Galán Freire

Ilustración 8-18 Pantalla plantilla de equipo

Los nombres de los jugadores son clickables, permitiendo también acceder al detalle del jugador seleccionado desde aquí.



Ilustración 8-19 Pantalla de detalle de equipo

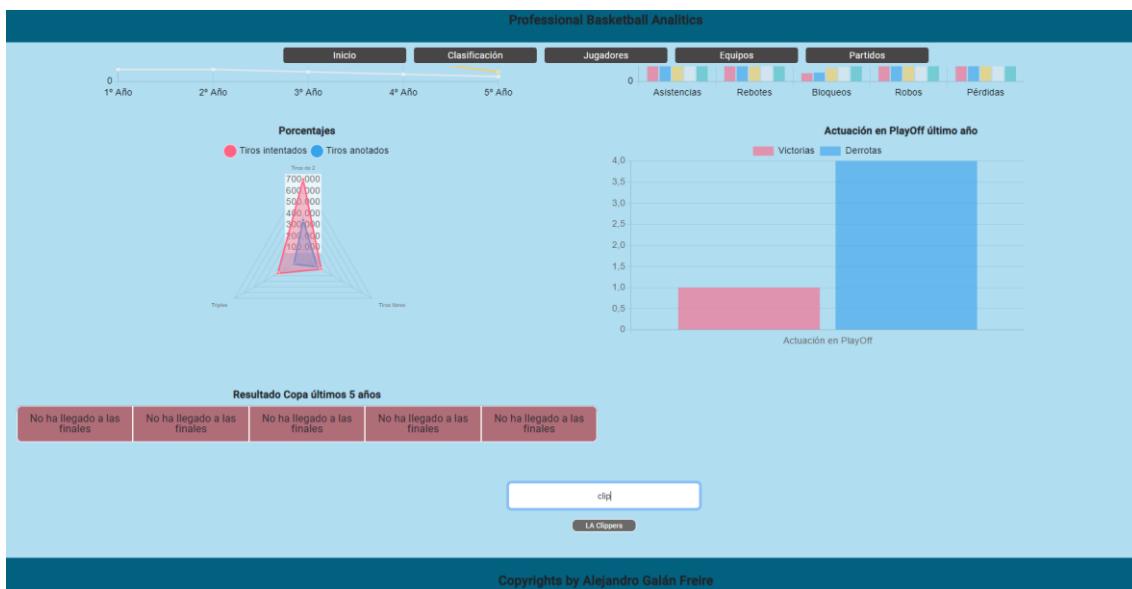
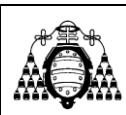


Ilustración 8-20 Búsqueda de equipo para comparar

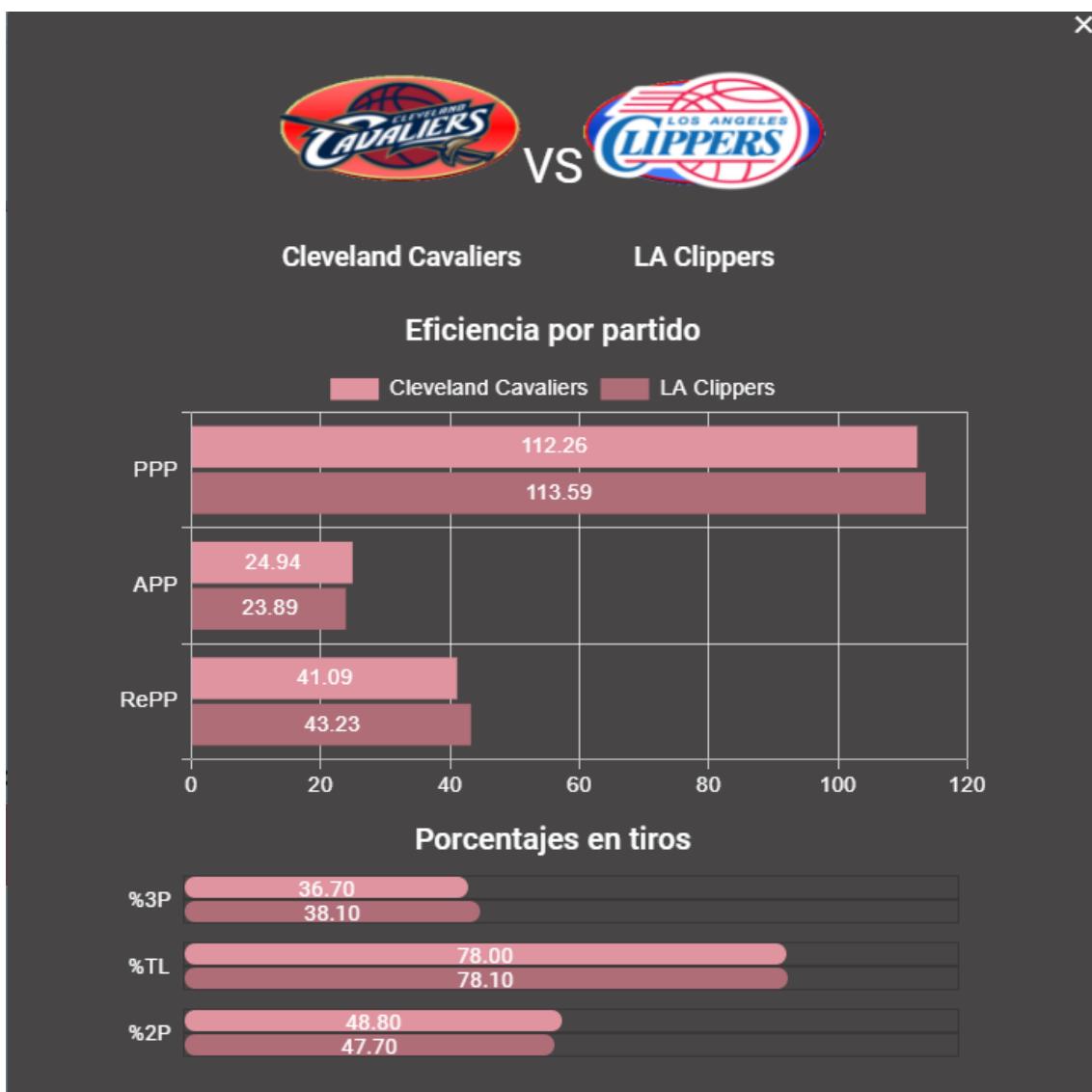
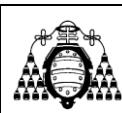


Ilustración 8-21 Pantalla de comparación de equipos

Por último, explicar la sección de 'Partidos'. La pantalla que aparece tras seleccionar esta opción de menú es la siguiente:

Ilustración 8-22 Pantalla de visualización de partidos

A través del calendario ubicado en la parte superior central podemos seleccionar una fecha concreta para ver los partidos jugados ese día:

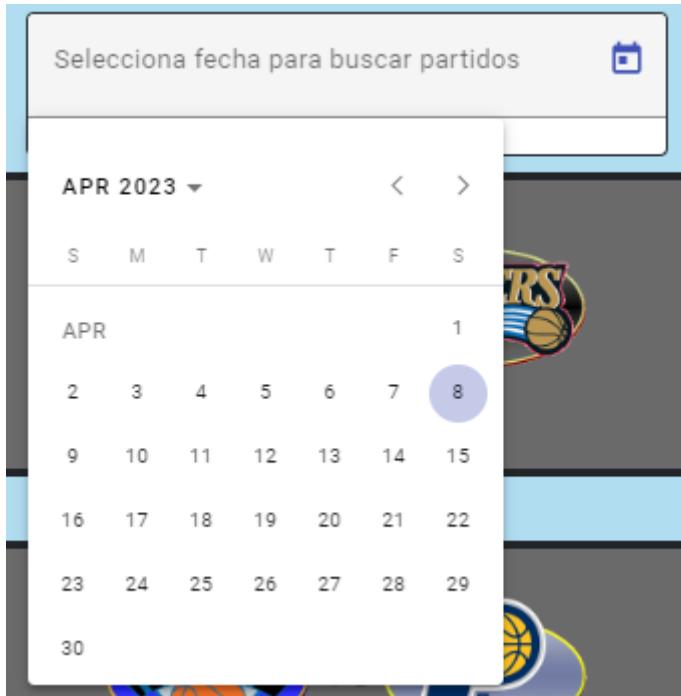


Ilustración 8-23 Calendario para seleccionar fecha de partidos

Para finalizar la explicación de uso de la aplicación cabe destacar qué es lo que ocurre cuando se elige una fecha en la que no existen partidos, o cuando se trata de filtrar por un jugador o equipo que no existe, ya sea en el listado de jugadores y equipos o en la opción de comparar.

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 159 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	---

Pues lo que ocurre es que se muestra un mensaje al usuario como los siguientes:

No encuentra coincidencias en el buscador:

Professional Basketball Analytics

Inicio Clasificación Jugadores Equipos

afsañ

NO SE ENCUENTRAN COINCIDENCIAS

Ilustración 8-24 Búsqueda sin coincidencias

No existen partidos en la fecha seleccionada:

Professional Basketball Analytics

Inicio Clasificación Jugadores Equipos Partidos

Selecciona fecha para buscar partidos
6/30/2023

M/D/YYYY

No existen partidos para la fecha seleccionada

Ilustración 8-25 Fecha sin partidos

8.3 Manual del Programador

Vamos a continuación a indicar como configurar el entorno y cuál es la estructura seguida en los sistemas para realizar modificaciones o ampliaciones.

8.3.1 Backend

En primer lugar, el backend. En este proyecto se utiliza Spring Tool Suite para trabajar con el proyecto ya que es en Spring Boot. Son necesarias configurar las siguientes dependencias en el fichero 'pom.xml':

Dependencia web (necesaria para que pueda ser un sitio Web)

```
<!-- Dependencia web -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
</dependency>
```

Ilustración 8-26 Dependencia web

Spring JDBC (necesaria para las anotaciones correspondientes)

```
<!-- Spring JDBC -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

Ilustración 8-27 Dependencia Spring

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 161 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	---

MySQL (Necesario para poder interactuar con bases de datos relacionales)

```
<!-- JDBC Driver con MySQL -->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.32</version>
</dependency>
```

Ilustración 8-28 Dependencia MySQL

Jacoco (necesario para SonarQube)

```
<dependency>
    <groupId>org.jacoco</groupId>
    <artifactId>jacoco-maven-plugin</artifactId>
    <version>0.8.7</version>
</dependency>
```

Ilustración 8-29 Dependencia Jacoco

MyBatis (Necesario para el tratamiento de la Base de Datos)

```
<!-- Starter MyBatis -->
<dependency>
    <groupId>org.mybatis.spring.boot</groupId>
    <artifactId>mybatis-spring-boot-starter</artifactId>
    <version>3.0.1</version>
</dependency>
```

Ilustración 8-30 Dependencia MyBatis

MyBatis, el framework utilizado para interactuar con la Base de Datos, además presenta un fichero de configuración propio, 'mybatis-config.xml' que es donde se declaran las ubicaciones de los ficheros XML donde se definen las consultas a la base de datos, para que los mapper puedan acceder a estos ficheros. Este archivo de configuración tiene el siguiente contenido:

```

-//mybatis.org//DTD Config 3.0//EN (doctype)
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
  "https://mybatis.org/dtd/mybatis-3-config.dtd">

> <configuration>
>   <mappers>
    <mapper resource="mapper/ClasificationRowMapper.xml"/>
    <mapper resource="mapper/TeamByYearMapper.xml"/>
    <mapper resource="mapper/PlayerMapper.xml"/>
    <mapper resource="mapper/GameElemMapper.xml"/>
    <mapper resource="mapper/LeagueMapper.xml"/>
  </mappers>
</configuration>

```

Ilustración 8-31 Configuración myBatis

Un ejemplo de Mapper es el siguiente:

```

-//mybatis.org//DTD Mapper 3.0//EN (doctype)
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "https://mybatis.org/dtd/mybatis-3-mapper.dtd">

> <mapper namespace="com.example.demo.clasification.mapper.ClasificationRowMapper">
>   <select id="getClasification" resultType="com.example.demo.clasification.model.ClasificationRow">
    SELECT *
    FROM clasificationrow
    WHERE league = #{league}
  </select>
</mapper>

```

Ilustración 8-32 Archivo XML de un Mapper

En él se indica la ubicación del Mapper para saber el nombre del método que se ejecuta para obtener la información, y el modelo para almacenar en él los datos obtenidos.

Existe otro archivo de configuración, que es necesario para indicar el puerto que se utilizará, realizar la conexión a la Base de Datos y finalmente indicar donde se encuentra el fichero de configuración de MyBatis. Este archivo se llama application.properties y su contenido es el siguiente:

server.port=8080

```
spring.datasource.driverClassName = com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/nbaanalytics_db
spring.datasource.username=root
spring.datasource.password=1234
spring.datasource.tomcat.test-while-idle=true
spring.datasource.tomcat.validation-query=SELECT 1

mybatis.config-location=classpath:mybatis-config.xml
```

Ilustración 8-33 Fichero application.properties

Por último, para la configuración local de la herramienta Sonarqube es necesario definir los siguientes comandos:

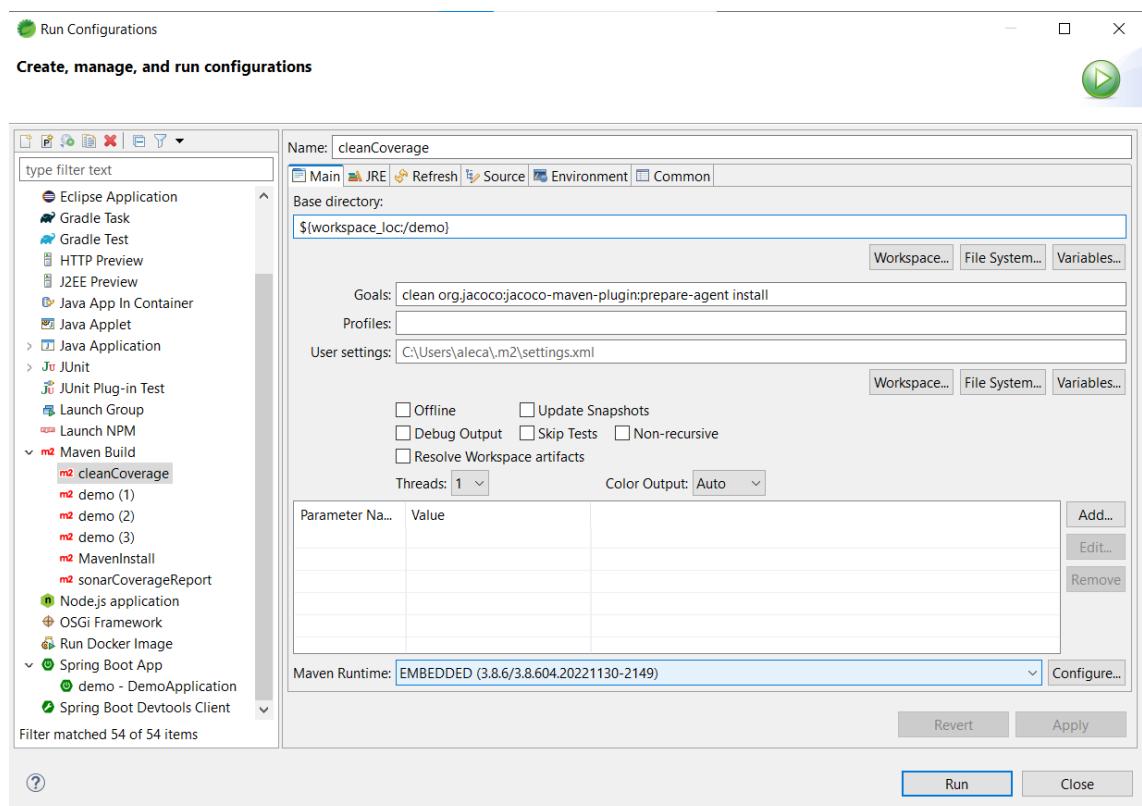


Ilustración 8-34 Configuración Sonarqueba del backend

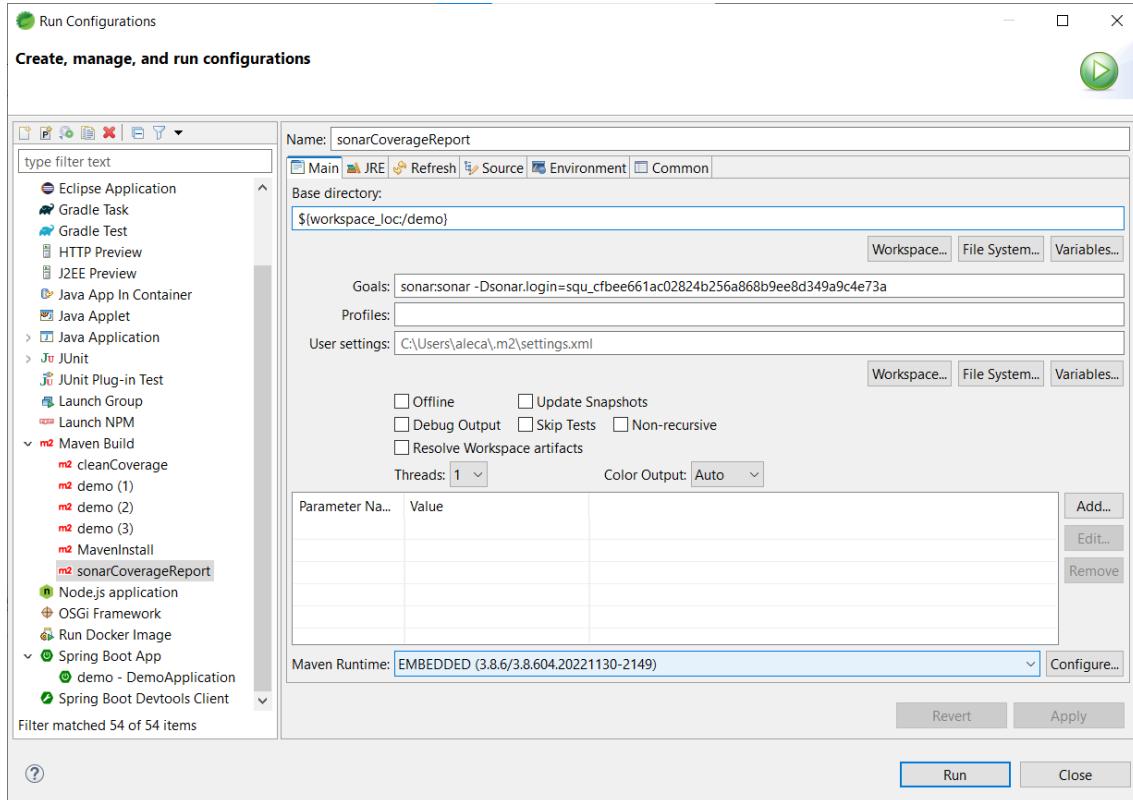


Ilustración 8-35 Configuración Sonarqube del backend (2)

Para configurarlos le damos click derecho al proyecto -> Run as -> Run Configurations, y definicimos los dos nuevos comandos.

Si se quiere añadir funcionalidad a un módulo ya existente simplemente se tiene que definir un nuevo endpoint en el 'controller' y sus correspondencias en el 'service' y 'mapper' adecuados, o simplemente modificar la lógica de un endpoint previamente definido.

Sin embargo, si se quiere añadir un módulo independiente, deberemos crear un paquete nuevo que contenga el modelo de la entidad nueva, y su controlador, servicio y mapper. Por supuesto, añadir el fichero XML para el mapper e indicar su existencia en 'mybatis-config.xml'.

8.3.2 Frontend

Continuamos con el frontend. Son necesarias tener las siguientes dependencias en el fichero Package.json:

```

"dependencies": {
  "@angular/animations": "^15.2.0",
  "@angular/cdk": "~15.2.5",
  "@angular/common": "^15.2.0",
  "@angular/compiler": "^15.2.0",
  "@angular/core": "^15.2.0",
  "@angular/forms": "^15.2.0",
  "@angular/material": "~15.2.5",
  "@angular/platform-browser": "^15.2.0",
  "@angular/platform-browser-dynamic": "^15.2.0",
  "@angular/router": "^15.2.0",
  "@fortawesome/fontawesome-free": "^6.4.0",
  "@material/tooltip": "^14.0.0",
  "@popperjs/core": "^2.11.7",
  "bootstrap": "^5.2.3",
  "chart.js": "^4.3.0",
  "chartjs-plugin-datalabels": "^2.2.0",
  "jquery": "^3.6.4",
  "ng2-charts": "^4.1.1",
  "rxjs": "~7.8.0",
  "tslib": "^2.3.0",
  "zone.js": "~0.12.0"
},
  
```

Ilustración 8-36 Dependencias de proyecto

Y estas dependencias de desarrollo:

```

"devDependencies": {
  "@angular-devkit/build-angular": "^15.2.0",
  "@angular/cli": "~15.2.0",
  "@angular/compiler-cli": "^15.2.0",
  "@types/chart.js": "^2.9.37",
  "@types/jasmine": "~4.3.0",
  "jasmine-core": "~4.5.0",
  "karma": "~6.4.0",
  "karma-chrome-launcher": "~3.1.0",
  "karma-coverage": "~2.2.0",
  "karma-jasmine": "~5.1.0",
  "karma-jasmine-html-reporter": "~2.0.0",
  "sonar-scanner": "^3.1.0",
  "typescript": "~4.9.4"
}
  
```

Ilustración 8-37 Dependencias de desarrollo

Todas ellas instaladas a través de la terminal de Visual Studio Code, con el comando 'npm install' y la correspondiente continuación según lo que se quiera instalar.

Por otra parte, para la ejecución del proyecto, las pruebas y el análisis con la herramienta SonarQube es necesario definir en el mismo archivo los siguientes scripts:

```
"scripts": {  
  "ng": "ng",  
  "start": "ng serve",  
  "build": "ng build",  
  "watch": "ng build --watch --configuration development",  
  "test": "ng test --no-watch --code-coverage",  
  "sonar": "sonar-scanner"  
},
```

Ilustración 8-38 Comandos ejecutables (scripts)

Para poder hacer las pruebas y configurar la cobertura necesaria se necesita añadir el fichero 'karma.conf.json' que presenta el siguiente contenido:



```
module.exports = function (config) {
  ...
  config.set({
    basePath: '',
    frameworks: ['jasmine', '@angular-devkit/build-angular'],
    plugins: [
      require('karma-jasmine'),
      require('karma-chrome-launcher'),
      require('karma-jasmine-html-reporter'),
      require('karma-coverage'),
      require('@angular-devkit/build-angular/plugins/karma')
    ],
    client: {
      // leave Jasmine Spec Runner output visible in browser
      clearContext: false
    },
    coverageReporter: {
      dir: require('path').join(__dirname, 'coverage'),
      fixWebpackSourcePaths: true,
      thresholds: {
        statements: 60,
        lines: 60,
        branches: 60,
        functions: 60
      },
      subdir: '..',
      reporters: [
        { type: 'html' },
        { type: 'lcov' },
        { type: 'text-summary' }
      ]
    },
    reporters: ['progress', 'kjhtml'],
    port: 9876,
    colors: true,
    logLevel: config.LOG_INFO,
    autoWatch: true,
    browsers: ['Chrome'],
    singleRun: false,
    restartOnFileChange: true
  });
};
```

Ilustración 8-39 Configuración Karma

8.3.3 Base de Datos

Para conectarte a la base de datos simplemente desde el MySQL Workbench debes introducir los datos y credenciales necesarias:

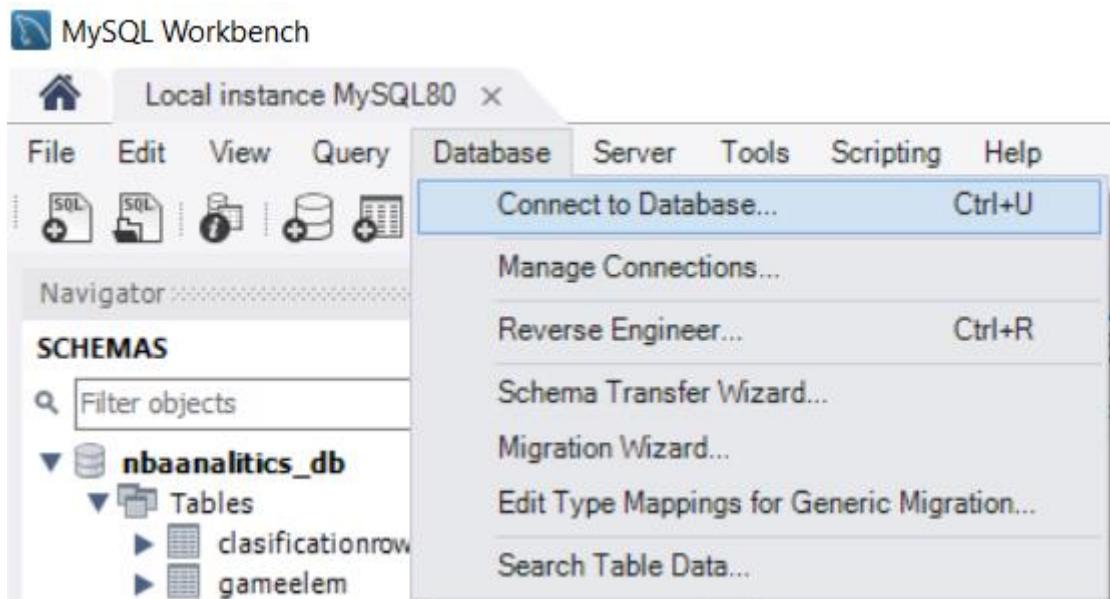


Ilustración 8-40 Opción de conexión a base de datos

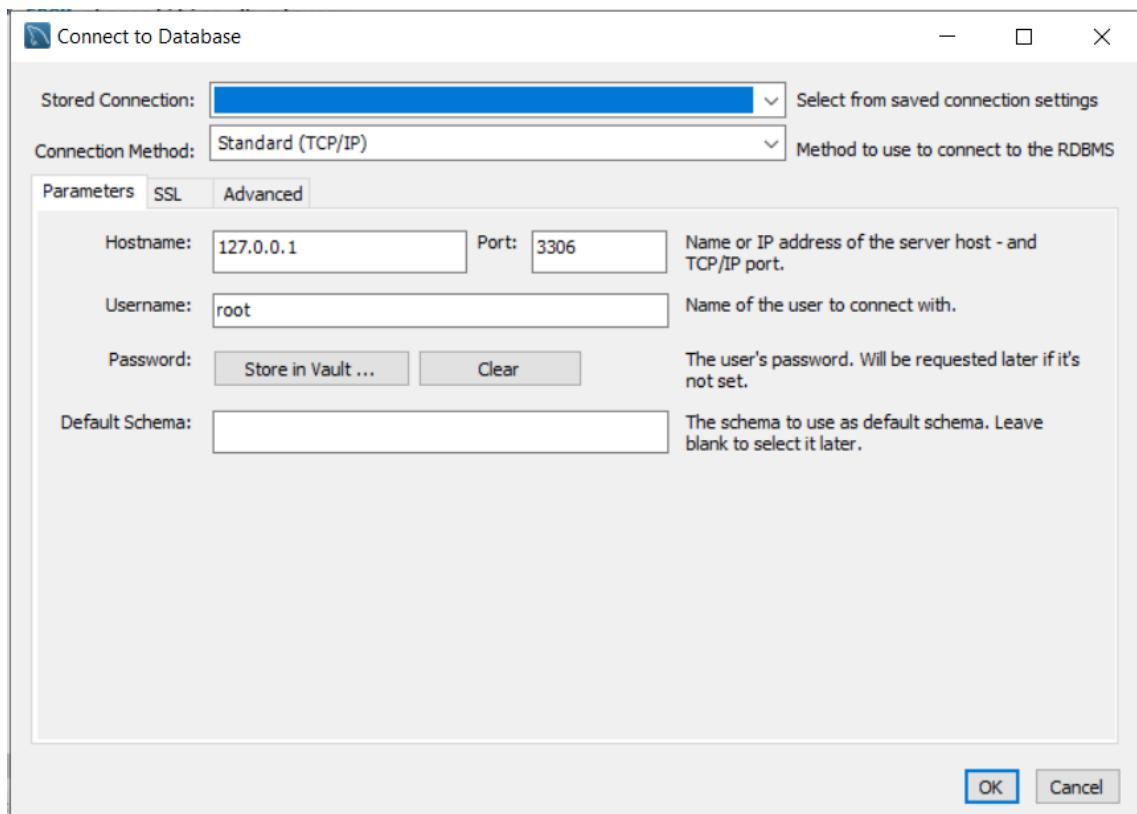


Ilustración 8-41 Conexión a base de datos

8.3.4 Extracción de datos

Para la parte de extracción de datos en Python no es necesaria ninguna configuración previa. Simplemente se tienen que hacer las importaciones necesarias para poder conectarte a los endpoints de las APIs a través de HTTP, importación del paquete json para leer los datos y de la librería 'pandas' para procesar los datos y enviarlos a ficheros .CSV.

```
from nba_api.stats.endpoints._base import Endpoint
from nba_api.stats.library.http import NBAStatsHTTP
from nba_api.stats.library.parameters import LastNGames
import json
import pandas as pd
```

Ilustración 8-42 Ejemplo importaciones fichero extracción de datos

Capítulo 9. Conclusiones y Ampliaciones

9.1 Conclusiones

Tras la realización de este proyecto durante estos meses podemos extraer varias conclusiones.

La primera es la importancia que tiene la realización de este proyecto en mi aprendizaje como desarrollador web gracias a la cantidad de cosas aprendidas, especialmente los lenguajes y herramientas utilizadas, la mayoría de ellos nuevos para mí. El entender la importancia de la modularidad en un sistema, del uso de frameworks y librerías que te faciliten el trabajo y la importancia de una correcta configuración para realizar el trabajo de la forma más eficiente posible.

Por otro lado, aprender a poner enfoque en todos los aspectos destacables en el desarrollo de un sistema, planificación y presupuesto, diseño, análisis, implementación y pruebas y la correcta documentación de estos para su buena comprensión.

Además, el resultado del trabajo se podría considerar que cumple con los objetivos pre establecidos al principio del proyecto, consiguiendo generar una API REST independiente de la interfaz de usuario para así poder ser reutilizada, conseguir también la escalabilidad del sistema, y hacer un procesamiento previo de los datos necesarios en nuestro sistema para no depender directamente de una API desplegada en internet.

Destacad por último la importancia dedicada a la limpieza del código y pruebas para conseguir que todo el sistema funcione correctamente, procurando tener una cobertura de código óptima y eliminando todas las malas prácticas y duplicidades de código identificadas por herramientas.

9.2 Ampliaciones

Pese a considerar haber cumplido con los objetivos principales establecidos en el proyecto, son varias las ampliaciones que se pueden considerar como trabajo a futuro.

Despliegue de la aplicación

Que menos en el desarrollo de un sitio web que conseguir el despliegue de este. No ha sido posible debido a distintos problemas que han surgido en los intentos llevados a cabo. Se necesitaría de más tiempo para lograrlo.

Implementación de las recomendaciones de los usuarios

En las pruebas de usabilidad realizadas a los usuarios, estos dieron sugerencias de mejores como crear un apartado específico para la comparación de jugadores y equipos, esto quedaría pendiente de desarrollar.

Automatizar procesamiento de datos

El procesamiento previo de todos los datos que se toman de las APIs para posteriormente almacenarlos en nuestra Base de Datos es manual, sería conveniente automatizar dicho proceso para que cada x tiempo se hiciera una petición a la API y nuestros ficheros Python se conectaran directamente a la Base de Datos almacenando la información procesada.

Internacionalización

Como mejora se podría internacionalizar la aplicación en otros idiomas.

Análisis más exhaustivo de datos

Es posible hacer análisis o comparativas más complejas de las estadísticas de jugadores, equipos y partidos.

Sistema de integración continua

Automatizar el proceso de ejecución de pruebas y análisis del código con un sistema de integración continua cada vez que, por ejemplo, se realizara un commit en el repositorio.

Incorporación de los DTOs necesarios

Para lograr la completa separación de las capas de la API REST aunque en la mayoría de los modelos se utilizan todos los datos aportados.

Capítulo 10. Referencias Bibliográficas

10.1 Referencias libros y plantillas

[Redondo] J. M. Redondo López, "Documentos-modelo para Trabajos de Fin de Grado/Master de la Escuela de Informática de Oviedo," Jun. 17, 2019. https://www.researchgate.net/publication/327882831_Plantilla_de_Proyectos_de_Fin_de_Carrera_de_la_Escuela_de_Informatica_de_Oviedo (accessed Feb. 23, 2023).

[Oliver] Dean Oliver, "Basketball On Paper". Potomac Books. 2011.

[Para planificación, riesgos y presupuesto] DPPI, "Guía de Aprendizaje de la asignatura de Dirección y Planificación de Proyectos Informáticos"

10.2 Referencias internet

[Angular] "Angular"

<https://angular.io/docs>

<https://www.tutorialesprogramacionya.com/angularya/>

[Bootstrap] "Bootstrap".

<http://getbootstrap.com>

<http://www.w3schools.com/bootstrap>

[Chart.js] "Chart.js tutorials"

<https://www.chartjs.org/docs/latest/>

<https://www.chartjs.org/docs/latest/getting-started/>

<https://www.chartjs.org/>

<https://www.chartjs.org/docs/latest/getting-started/usage.html>

<https://www.chartjs.org/docs/latest/charts/line.html>

<https://www.chartjs.org/docs/latest/getting-started/integration.html>

<https://www.chartjs.org/docs/latest/configuration/legend.html>

[CSS] "Tutoriales CSS"

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 172 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	---

<https://librosweb.es/libro/css>

<http://www.w3schools.com/css>

[Spring Boot] “Spring Boot tutorials”

<https://spring.io/guides/gs/spring-boot/>

<https://www.javatpoint.com/spring-boot-tutorial>

https://www.tutorialspoint.com/spring_boot/index.htm

<https://www.arquitecturajava.com/spring-boot-jdbc-y-su- configuracion/>

[MyBatis] “MyBatis tutorials”

<https://mybatis.org/spring/es/getting-started.html>

<https://www.baeldung.com/spring-mybatis>

<https://sainimanish700.medium.com/mybatis-with-spring-boot-example-d521f393a742>

<https://www.adictosaltrabajo.com/2016/06/29/integracion-de-mybatis-con-spring-boot-y-cache-con-redis/>

[MySQL Workbench] “MySQL Workbench tutorials”

<https://desarrolloweb.com/articulos/esquema-grafico-base-mysql-workbench.html>

<https://dev.mysql.com/doc/workbench/en/>

[Swar/nbaApi] “Swap Patel, API NBA”

https://github.com/swar/nba_api

[Estadísticas avanzadas y páginas de referencia] “referencias”

<https://spain.id.nba.com/reportajes/medir-rendimiento-jugador-nba>

<https://www.nba.com/stats>

https://www.espn.com.mx/basquetbol/nba/juego/_/juegolid/401468919

<https://www.sportingnews.com/es/nba/news/estadistica-avanzada-diccionario-terminos-que-es-que-sirve-nba-analytics-sabermetrics/1gc73ududvomy1s3mqiahg238c>

[Creación proyecto] “Creación subsistemas e integración”

https://www.youtube.com/watch?v=hIE_uZy9Vpw&list=PLsqnBdXG3GZlr638ZjonocVciM4g8SeLe&index=1

<https://www.youtube.com/watch?v=FHDtxyR6rnc&list=PLsqnBdXG3GZlr638ZjonocVciM4g8SeLe&index=3>

<https://www.youtube.com/watch?v=0fBLGBCfoHE&list=PLsqnBdXG3GZlr638ZjonocVciM4g8SeLe&index=4>

<https://www.youtube.com/watch?v=31DuU-98XtY&list=PLsqnBdXG3GZlr638ZjonocVciM4g8SeLe&index=6>

<https://www.youtube.com/watch?v=73G1xouigU&list=PLsqnBdXG3GZlr638ZjonocVciM4g8SeLe&index=8>

<https://www.youtube.com/watch?v=Wgdg9C8LDZo&list=PLsqnBdXG3GZlr638ZjonocVciM4g8SeLe&index=9>

<https://www.youtube.com/watch?v=fOzX1gtgwwI&list=PLsqnBdXG3GZlr638ZjonocVciM4g8SeLe&index=10>

https://www.youtube.com/watch?v=g_zoy9m0KMs&list=PLsqnBdXG3GZlr638ZjonocVciM4g8SeLe&index=11

https://www.youtube.com/watch?v=ZP8Um12Z_mk&list=PLsqnBdXG3GZlr638ZjonocVciM4g8SeLe&index=12

[Sonarqube] “Sonarqube tutorials”

https://www.youtube.com/results?search_query=sonarqube

<https://www.youtube.com/watch?v=pufcNkJPvxw>

<https://www.youtube.com/watch?v=6pLj3KVglFA>

<https://www.youtube.com/watch?v=ZN5k2prUw4Q&t=345s>

<https://www.youtube.com/watch?v=LpjANAPF2Oo&t=79s>

Capítulo 11. Apéndices

11.1 Detalle del presupuesto

En este apartado vamos a hablar sobre el modelo de la empresa supuesto para la realización del presupuesto, así como el detalle de este.

11.1.1 Definición de la empresa

La empresa supuesta dispone de dos trabajadores, el consultor que se va a encargar de tener las reuniones con el tutor del proyecto y el desarrollador que se ocupará del resto. En Tabla 75 Personal de la empresa se puede ver el coste que suponen para la empresa estos recursos de tipo empleado.

Personal	Cantidad	Sueldo bruto anual	Coste salarial anual	Total
Desarrollador	1	24.000,00 €	30.000,00 €	30.000,00 €
Consultor	1	25.500,00 €	31.875,00 €	31.875,00 €

Tabla 75 Personal de la empresa

Teniendo en cuenta la productividad que aporta cada empleado al proyecto obtenemos los costes directos e indirectos que suponen para la empresa.

Personal	Total	Productividida	Coste Directo	Coste indirecto (%)	Coste indirecto
Desarrollador	30.000,00 €	90,00%	27.000,00 €	10,00%	3.000,00 €
Consultor	32.000,00 €	70,00%	22.400,00 €	30,00%	9.600,00 €
Total			49.400,00 €		12.600,00 €

Tabla 76 Productividad y costes

Además de los empleados, la empresa también ha de contratar ciertos servicios, como costes indirectos.

Servicio	Coste anual
Limpieza	1.200,00 €
Oficina	5.000,00 €
Seguros	2.500,00 €
Material oficina	450,00 €
Agua	600,00 €
Electricidad	1.500,00 €
Mensajería	200,00 €
Impuestos	1.500,00 €
Total	12.950,00 €

Tabla 77 Otros costes

Teniendo en cuenta también la productividad de cada trabajador y las horas que se trabajan al año en España aproximadamente calculamos las horas que realmente son productivas. En base a eso y al precio por hora sin beneficios hemos de obtener un precio por hora con beneficios que nos aporte una facturación positiva a la empresa.

Para hacer el cálculo del precio/hora con beneficios cogemos el coste salarial que el trabajador supone para la empresa y lo dividimos por las horas/año, para suponer que el empleado trabaja todas las horas posibles. El resultado de esta operación lo multiplicamos por 2 y tenemos un precio con hora con beneficios suficiente como para tener una facturación con margen, como se puede observar en *Tabla 78 Facturación* y *Tabla 79 Resumen modelo de la empresa*

Personal	Productividad	Horas/año	Horas productivas/año	Precio hora sin beneficios	Precio hora con beneficios	Facturación
Desarrollador	90,00%	1826	1643,4	13,14 €	32,86 €	54.000,00 €
Consultor	70,00%	1826	1278,2	13,96 €	34,91 €	44.625,00 €
				Total		98.625,00 €

Tabla 78 Facturación

Concepto	IMPORTE
Total de los costes directos	49.400,00 €
Total de los costes indirectos	25.550,00 €
Suma de los costes directos e indirectos	74.950,00 €
Beneficio deseado (25%)	18.737,50 €
Coste total (costes directos, indirectos y beneficios)	93.687,50 €
Facturación posible	98.625,00 €
Margen entre coste total y facturación	5,27%

Tabla 79 Resumen modelo de la empresa

11.1.2 Presupuesto de costes

Vamos ahora a calcular el coste que el proyecto supone para la empresa. Para ello, hemos de calcular el coste de cada partida, teniendo en cuenta los empleados que participan en ellas y cuánto tiempo invierten.

En primer lugar, las tutorías, habiendo un total de 18 reuniones de 1 hora cada una donde el encargado de realizarlas es el consultor. El coste de esta partida es de 251,28€.

Partida 1: Tutorías

I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal(2)	Total
01			Tutorías						251,28 €
	001		Realizar tutorías	18	reuniones			251,28 €	
		01	Consultor	18	horas	13,96 €	251,28 €		
								Total	251,28 €

Tabla 80 Partida Tutorías

La siguiente partida es la de investigación, a partir de aquí ya será siempre el desarrollador el encargado de realizar todas las tareas, y en las correspondientes tablas se ven las horas invertidas en cada una de ellas, calculando así también el coste que suponen para el proyecto.

Partida 2: Investigación

I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal(2)	Total
01			Investigación						279,20 €
	001		Investigar páginas similares					97,72 €	
		01	Desarrollador	7	horas	13,96 €	97,72 €		
	002		Investigar las tecnologías					97,72 €	
		01	Desarrollador	7	horas	13,96 €	97,72 €		
	003		Investigar APIs					83,76 €	
		01	Desarrollador	6	horas	13,96 €	83,76 €		
								Total	279,20 €

Tabla 81 Partida Investigación

Partida 3: Planificación y gestión del proyecto

I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal(2)	Total
01			Planificación y gestión del proyecto						279,20 €
	001		Realizar la planificación					97,72 €	
		01	Desarrollador	7	horas	13,96 €	97,72 €		
	002		Identificar y estudiar riesgos					69,80 €	
		01	Desarrollador	5	horas	13,96 €	69,80 €		
	003		Realizar el presupuesto					111,68 €	
		01	Desarrollador	8	horas	13,96 €	111,68 €		
								Total	279,20 €

Tabla 82 Partida Planificación y gestión del proyecto

Partida 4: Análisis

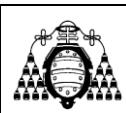
I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal(2)	Total
01			Análisis						893,44 €
	001		Definir el sistema					97,72 €	
		01	Desarrollador	7	horas	13,96 €	97,72 €		
	002		Obtener los requisitos del sistema					125,64 €	
		01	Desarrollador	9	horas	13,96 €	125,64 €		
	003		Identificar los subsistemas					69,80 €	
		01	Desarrollador	5	horas	13,96 €	69,80 €		
	004		Realizar el diagrama de clases preliminar					125,64 €	
		01	Desarrollador	9	horas	13,96 €	125,64 €		
	005		Analizar casos de uso y escenarios					167,52 €	
		01	Desarrollador	12	horas	13,96 €	167,52 €		
	006		Analizar las interfaces de usuario					167,52 €	
		01	Desarrollador	12	horas	13,96 €	167,52 €		
	007		Especificar el plan de pruebas					139,60 €	
		01	Desarrollador	10	horas	13,96 €	139,60 €		
								Total	893,44 €

Tabla 83 Partida Análisis

Partida 5: Diseño

I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal(2)	Total
01			Análisis						698,00 €
	001		Definir la arquitectura del sistema					69,80 €	
		01	Desarrollador	5	horas	13,96 €	69,80 €		
	002		Diseñar las clases					167,52 €	
		01	Desarrollador	12	horas	13,96 €	167,52 €		
	003		Crear diagramas de interacción					27,92 €	
		01	Desarrollador	2	horas	13,96 €	27,92 €		
	004		Diseñar la base de datos					139,60 €	
		01	Desarrollador	10	horas	13,96 €	139,60 €		
	005		Diseñar la interfaz					195,44 €	
		01	Desarrollador	14	horas	13,96 €	195,44 €		
	006		Especificar técnicamente el plan de pruebas					97,72 €	
		01	Desarrollador	7	horas	13,96 €	97,72 €		
								Total	698,00 €

Tabla 84 Partida Diseño



Partida 6: Implementación

I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal(2)	Total
01			Implementación						1.731,04 €
	001		Procesar los datos de las APIs					167,52 €	
		01	Desarrollador	12	horas	13,96 €	167,52 €		
	002		Crear la Base de Datos					139,60 €	
		01	Desarrollador	10	horas	13,96 €	139,60 €		
	003		Crear la interfaz de usuario					167,52 €	
		01	Desarrollador	12	horas	13,96 €	167,52 €		
02			Lógica de negocio					1.256,40 €	
	001		Visualizar la clasificación de distintas ligas						
		01	Desarrollador	12	horas	13,96 €	167,52 €		
	002		Visualizar todos los jugadores						
		01	Desarrollador	10	horas	13,96 €	139,60 €		
	003		Visualizar todos los equipos						
		01	Desarrollador	10	horas	13,96 €	139,60 €		
	004		Visualizar estadísticas avanzadas de un jugador						
		01	Desarrollador	10	horas	13,96 €	139,60 €		
	005		Visualizar estadísticas avanzadas de un equipo						
		01	Desarrollador	10	horas	13,96 €	139,60 €		
	006		Visualizar la plantilla de un equipo						
		01	Desarrollador	8	horas	13,96 €	111,68 €		
	007		Visualizar estadísticas de un partido						
		01	Desarrollador	10	horas	13,96 €	139,60 €		
	008		Comparar dos jugadores						
		01	Desarrollador	10	horas	13,96 €	139,60 €		
	009		Comparar dos equipos						
		01	Desarrollador	10	horas	13,96 €	139,60 €		
								Total	1.731,04 €

Tabla 85 Partida Implementación

Partida 7: Desarrollo de pruebas

I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal(2)	Total
01			Desarrollo de pruebas						530,48 €
	001		Pruebas unitarias					139,60 €	
		01	Desarrollador	10	horas	13,96 €	139,60 €		
	002		Pruebas de calidad					125,64 €	
		01	Desarrollador	9	horas	13,96 €	125,64 €		
	003		Pruebas del sistema					139,60 €	
		01	Desarrollador	10	horas	13,96 €	139,60 €		
	004		Pruebas de usabilidad					125,64 €	
		01	Desarrollador	9	horas	13,96 €	125,64 €		
								Total	530,48 €

Tabla 86 Partida Desarrollo de pruebas

Partida 8: Documentación

I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal(2)	Total
01			Documentación						404,84 €
	001		Realizar documentación de apéndices					195,44 €	
		01	Desarrollador	14	horas	13,96 €	195,44 €		
	002		Realizar revisión de documentación					209,40 €	
		01	Desarrollador	15	horas	13,96 €	209,40 €		
								Total	404,84 €

Tabla 87 Partida Documentación

Una vez calculado el coste de todas las partidas añadimos como costes los desplazamientos que se tienen que realizar debido a las reuniones presenciales que tiene que hacer el consultor con el tutor, es decir, las tutorías, siendo 12 presenciales y 6 online.

Partida 9: Otros costes

I1	I2	I3	Descripción	Cantidad	Unidades	Precio	Subtotal (3)	Subtotal(2)	Total
01			Otros costes						64,80 €
	001		Viajes a la universidad					64,80 €	
		01	Desplazamiento en tren	12	viajes	5,40 €	64,80 €		
								Total	64,80 €

Tabla 88 Partida Otros costes

Por último, añadir el coste debido a la compra del hardware y software necesario para la realización del proyecto.

Hardware y software

Material	Coste
Portátil	500,00 €
Licencia Office 365	28,75 €
Licencia Microsoft Project 2019	14,90 €
Total	543,65 €

Tabla 89 Hardware y Software

Presupuesto de costes

Código	Partida	Total
01	Partida 1: Tutorías	251,28 €
02	Partida 2: Investigación	279,20 €
03	Partida 3: Planificación y gestión del proyecto	279,20 €
04	Partida 4: Análisis	893,44 €
05	Partida 5: Diseño	698,00 €
06	Partida 6: Implementación	1.731,04 €
07	Partida 7: Desarrollo de pruebas	530,48 €
08	Partida 8: Documentación	404,84 €
09	Partida 9: Otros costes	64,80 €
10	Hardware y software	543,65 €
	Total	5.675,93 €

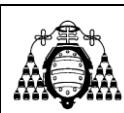
Tabla 90 Presupuesto de costes

11.1.3 Presupuesto de cliente

Al cliente solo le podemos mostrar las partidas que aparecen en *Tabla 92 Presupuesto cliente*, ya que los costes de desplazamiento, hardware y software son responsabilidad de la propia empresa. Aun así, para hacer las ponderaciones añadimos también estos cálculos, que junto con el beneficio del presupuesto de coste (un 25%) nos da la cantidad a compensar.

Con esta cantidad, y con el precio que se suma de las partidas que le presentaremos al cliente hallamos el porcentaje que se ha de aplicar a todas esas partidas para obtener el beneficio deseado.

Es debido a eso que en *Tabla 92 Presupuesto cliente* el coste de las partidas aumenta con respectos a *Tabla 90 Presupuesto de costes*.



Ponderaciones

Presupuesto coste	5.675,93 €
otros costes (hardware + otros costes)	608,45 €
Beneficio (25%)	1.418,98 €
Cantidad a compensar	2.027,43 €
Suma cantidad partidas	5.067,48 €
Porcentaje a compensar	40,01%

Tabla 91 Ponderaciones

Presupuesto cliente

Código	Partida	Total
01	Partida 1: Tutorías	351,81 €
02	Partida 2: Investigación	390,90 €
03	Partida 3: Planificación y gestión del proyecto	390,90 €
04	Partida 4: Análisis	1.250,89 €
05	Partida 5: Diseño	977,26 €
06	Partida 6: Implementación	2.423,61 €
07	Partida 7: Desarrollo de pruebas	742,72 €
08	Partida 8: Documentación	566,81 €
Total		7.094,91 €

Coste sin IVA	7.094,91 €
IVA (21%)	1.489,93 €
Coste con IVA	8.584,84 €

Tabla 92 Presupuesto cliente

11.2 Glosario y Diccionario de Datos

11.2.1 Glosario

Por orden alfabético, se mencionarán y explicarán todos los términos que se consideran importantes en este proyecto.

- **Responsive:** Se refiere a un diseño o desarrollo web que se adapta y visualiza correctamente en diferentes dispositivos y tamaños de pantalla, como computadoras, tablets y teléfonos móviles
- **Escalable:** Se refiere a la capacidad de un sistema o una aplicación para crecer y manejar un aumento en la carga de trabajo o usuarios sin perder rendimiento o funcionalidad.
- **SPA:** Es una aplicación web que se carga completamente en una sola página y no requiere recargarla al interactuar con ella. Proporciona una experiencia de usuario fluida y rápida.
- **Framework:** Es un conjunto de herramientas, bibliotecas y convenciones de programación que facilitan el desarrollo de aplicaciones al proporcionar una estructura predefinida y funcionalidades comunes.

- **Librería:** Es un conjunto de código prescrito y predefinido que se puede utilizar en el desarrollo de software para agregar funcionalidades específicas sin tener que escribir todo el código desde cero.
- **Clave primaria:** Es un campo o conjunto de campos en una base de datos que identifica de manera única cada registro dentro de una tabla. Se utiliza para garantizar la integridad y la relación entre diferentes tablas.
- **Clave externa:** Es un campo o conjunto de campos en una tabla que establece una relación con la clave primaria de otra tabla. Se utiliza para establecer la integridad referencial y las relaciones entre diferentes tablas en una base de datos relacional.
- **IDE:** Es un entorno de desarrollo integrado que proporciona herramientas y funcionalidades para facilitar la escritura, prueba y depuración de código en un solo lugar, como un editor de texto, un depurador y un compilador.
- **API REST:** Es un conjunto de reglas y convenciones para construir y acceder a servicios web. Permite que diferentes aplicaciones se comuniquen entre sí y comparten datos de manera sencilla y estándar a través de HTTP.
- **Backend:** Se refiere a la parte de una aplicación o sistema que se encarga del procesamiento de datos, la lógica de negocio y la comunicación con la base de datos. Es la "parte del servidor" que no es visible para el usuario final.
- **WebApp:** Es una aplicación web que se ejecuta en un navegador web. A diferencia de las aplicaciones tradicionales, no requiere instalación en el dispositivo del usuario y se puede acceder a ella a través de una URL.
- **Frontend:** Se refiere a la parte de una aplicación o sistema que se encarga de la interfaz de usuario y la interacción visual con el usuario. Es la "parte del cliente" que se ejecuta en el navegador web del usuario.
- **CSV:** Es un formato de archivo que se utiliza para almacenar datos tabulares, donde los valores de cada columna se separan por comas. Es ampliamente utilizado para importar y exportar datos en hojas de cálculo y bases de datos.
- **JSON:** Es un formato de intercambio de datos ligero y legible por humanos. Se utiliza ampliamente para representar estructuras de datos y transmitir información entre aplicaciones.
- **SGBD:** Es un software que permite administrar y gestionar bases de datos. Proporciona una interfaz para crear, almacenar, modificar y recuperar datos de manera eficiente y segura.
- **Pipe (Angular):** Función que transforma los datos antes de mostrarlos en la interfaz de usuario. Puedes usar Pipes para realizar filtrado, ordenamiento, formato de fechas, entre otras transformaciones en los datos.
- **Componente (Angular):** es una parte fundamental de la arquitectura de la aplicación. Representa una parte de la interfaz de usuario y contiene lógica relacionada con esa parte específica. Los componentes se componen de una plantilla HTML, estilos CSS y un controlador TypeScript.
- **Servicio (Angular):** Es una clase que proporciona funcionalidades compartidas y datos a través de la aplicación. Los servicios se utilizan para centralizar la lógica de negocio, la comunicación con el servidor y la gestión de datos. Se pueden inyectar en los componentes u otros servicios que los necesiten.

- **Módulo (Angular):** es un contenedor para organizar los componentes, servicios y otros elementos de la aplicación. Cada aplicación de Angular está compuesta por uno o varios módulos. Los módulos ayudan a modularizar y separar la funcionalidad de la aplicación en unidades lógicas y reutilizables.
- **Controller (MVC - Modelo Vista Controlador):** En el patrón de diseño MVC, el Controller es la parte encargada de manejar las interacciones del usuario y coordinar la comunicación entre la Vista y el Modelo. Recibe las solicitudes del usuario, actualiza el Modelo según sea necesario y selecciona la Vista adecuada para mostrar los resultados.
- **Model (MVC - Modelo Vista Controlador):** En el patrón de diseño MVC, el Model representa el objeto que almacenará la información solicitada hasta llegar a su destino.
- **Service (MVC - Modelo Vista Controlador):** En el contexto de MVC, un Service se refiere a una capa de la aplicación que se encarga de la lógica de negocio y la interacción con los datos. Los servicios encapsulan funcionalidades específicas que pueden ser utilizadas por varios componentes del sistema. Se utilizan para mantener un código más modular y reutilizable.
- **Mapper (MVC - Modelo Vista Controlador):** En el contexto de MVC, un Mapper es una capa que se encarga de comunicarse con la Base de Datos y mapear estos en el modelo del backend.

11.2.2 Diccionario de datos

En la aplicación las estadísticas de jugadores, equipos, partidos y demás aparecen abreviadas ya que tienen nombres largos. En este apartado se van a dar el significado de cada una de estas estadísticas, teniendo la propia aplicación además el nombre completo cuando se pasa por encima del dato.

Además de las estadísticas básicas, como pueden ser los puntos realizados por un jugador, existen otro tipo de estadísticas calculadas a través de estas consideradas como básicas, por medio de fórmulas. Estas también serán explicadas en este apartado.

GP: Partidos jugados

Won: Victorias

Lost: Derrotas

WonPct: Porcentaje de victorias

Fgm: Tiros anotados

Fga: Tiros intentados

FgPct: Porcentaje de tiros anotados

Fg3m: Tiros de 3 anotados

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 185 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	---

Fg3a: Tiros de 3 intentados

Fg3Pct: Porcentaje de tiros de 3 anotados

Ftm: Tiros libres anotados

Fta: Tiros libres intentados

FtPct: Porcentaje de tiros libres anotados

Oreb: Rebores ofensivos

Dreb: Rebotes defensivos

Reb: Rebores

Ast: Asistencias

Tov: Pérdidas

Stl: Robos

Blk: Bloqueos realizados

Blka: Bloqueos intentados

Pf: Faltas

Pts: Puntos

Min: Minutos jugados

urlPicture: url de la imagen del jugador u equipo

SeasonExp: Experiencia del jugador en el baloncesto profesional

Jersey: Número de camiseta

ConfRank: Posición de un equipo en su conferencia

DivRank: Posición de un equipo en su división

PoWins: Victorias en PlayOffs

PoLosses: Derrotas en PlayOffs

NbaFinalsAppearance: Variable que indica si ha aparecido o no en las finales del PlayOff en un año en concreto

Period1Score: Puntos en el primer cuarto

Period2Score: Puntos en el segundo cuarto

Period3Score: Puntos en el tercer cuarto

Period4Score: Puntos en el último cuarto

Ahora vamos a explicar algunas de las estadísticas obtenidas por fórmulas, para los jugadores concretamente.

val: Valoración de un jugador calculada a través de la siguiente fórmula:

$$\text{val} = (\text{reb} + \text{reb} + \text{ast} + \text{stl} + \text{blk} + \text{successfulShots} - \text{missedShots} - \text{tov})$$

Siendo **successfulShots** la suma de **fgm**, **fg3m** y **ftm** y **missedShots** la diferencia de acertados e intentados, es decir, los tiros fallados.

EFF_PerGame: Eficiencia del jugador por partido

$$\text{EFF_PerGame} = (\text{points} + \text{rebounds} + \text{ast} + \text{stl} + \text{blk} - \text{missedShots} - \text{tov}) / \text{gp}$$

EFF_PerMinute: Eficiencia del jugador por minuto

$$\text{EFF_PerMinute} = (\text{points} + \text{rebounds} + \text{ast} + \text{stl} + \text{blk} - \text{missedShots} - \text{tov}) / \text{min} * 48$$

PointsCreated: Puntos creados por un jugador

$$\text{pointsCreated} = (\text{points} + (1.42 * \text{ast})) / \text{gp}$$

TrueShootingPercentage: Porcentaje real de tiro de un jugador

$$\text{trueShootingPercentage} = \text{points} / (2 * (\text{fieldShotsAttempted} + 0.44 * \text{freeShotsAttempted}))$$

Siendo **fieldShotsAttempted** la suma de **fga** y **fg3a** y **freeShotsAttempted** **fta**.

11.3 Contenido Entregado en el Archivo adjunto

11.3.1 Contenidos

En este apartado se va a explicar la estructura de la entrega, es decir, que documentos se entregan junto con esta documentación y que contiene cada uno.

Se entregará un archivo zip llamado 'Anexos.zip' con el siguiente contenido:

- **Código fuente:** carpeta que contiene el código realizado para desarrollar el sitio web.
Contiene las siguientes subcarpetas:
 - o **Backend:** código correspondiente a la REST API
 - o **Frontend:** código correspondiente a la Webapp
 - o **Base de datos:** la base de datos del sistema y sus tablas

- **Extracción de datos:** código encargado de extraer y normalizar los datos de las APIs.
- **Documentación subsistemas:** carpeta que contiene la documentación generada del código, tanto del backend como del frontend.
- **PlanificaciónInicial.mpp:** fichero de Project con la planificación realizada al principio del proyecto.
- **PlanificaciónFinal.mpp:** fichero de Project con la planificación final, ajustando la inicial a medida que el proyecto iba avanzando.
- **Presupuesto.xlsx:** fichero Excel con el proceso seguido para calcular el coste del proyecto.
- **Video app:** carpeta que contiene un archivo .mkv, que es una muestra en video de la aplicación desarrollada.

11.4 Código Fuente

Todo el código fuente es adjuntado junto con esta documentación, pero este apartado va a destacar alguna parte del código de cada subsistema para su mejor comprensión. Para hacer esta explicación utilizaremos el módulo de 'Jugadores'.

11.4.1 Procesamiento previo

Para todos los módulos del sistema la primera operación a realizar es la extracción de la información necesaria de la API consumida.

La API usada principalmente en nuestro sitio es https://github.com/swar/nba_api para consumir datos de la NBA. Esta define una serie de endpoints que nos facilitan la información.

Entonces, para el caso de los jugadores, concretamente para obtener los datos generales de los mismos para sus tarjetas se ha realizado un fichero Python que hace la petición a la API y que posteriormente hace el procesamiento de la información obtenida para almacenarla en la Base de Datos.

Esta sería la clase que define el endpoint:

```
class CommonPlayerInfo(Endpoint):
    endpoint = 'commonplayerinfo'
    expected_data = {'AvailableSeasons': ['SEASON_ID'], 'CommonPlayerInfo': ['PERSON_ID', 'FIRST_NAME', 'LAST_NAME', 'DISPLAY_FIRST_NAME', 'DISPLAY_LAST_NAME', 'DISPLAY_NAME', 'BIRTH_DATE', 'BIRTH_PLACE', 'DEATH_DATE', 'DEATH_PLACE', 'GENDER', 'HEIGHT', 'WEIGHT', 'POSITION', 'COLLEGE', 'DRAFT_YEAR', 'DRAFT_ROUND', 'DRAFT_NUMBER', 'DRAFT_TYPE', 'DRAFT_STATUS', 'DRAFT_SEASON', 'DRAFT_TEAM', 'LAST_UPDATE'], 'PlayerHeadlineStats': ['PERSON_ID', 'DISPLAY_NAME', 'LAST_UPDATE'], 'PlayerStats': ['PERSON_ID', 'DISPLAY_NAME', 'LAST_UPDATE'], 'TeamStats': ['TEAM_ID', 'DISPLAY_NAME', 'LAST_UPDATE']}

    nba_response = None
    data_sets = None
    player_stats = None
    team_stats = None
    headers = None

    def __init__(self, player_id, league_id_nullable=LeagueIDNullable.default, proxy=None, headers=None, timeout=30, get_request=True):
        self.proxy = proxy
        if headers is not None:
            self.headers = headers
        self.timeout = timeout
        self.parameters = {
            'PlayerID': player_id,
            'LeagueID': league_id_nullable
        }
        if get_request:
            self.get_request()

    def get_request(self):
        self.nba_response = NBAStatsHTTP().send_api_request(
            endpoint=self.endpoint,
            parameters=self.parameters,
            proxy=self.proxy,
            headers=self.headers,
            timeout=self.timeout,
        )
        self.load_response()

    def load_response(self):
        data_sets = self.nba_response.get_data_sets()
        self.data_sets = [Endpoint.DataSet(data=data_set) for data_set_name, data_set in data_sets.items()]
        self.available_seasons = Endpoint.DataSet(data=data_sets['AvailableSeasons'])
        self.common_player_info = Endpoint.DataSet(data=data_sets['CommonPlayerInfo'])
        self.player_headline_stats = Endpoint.DataSet(data=data_sets['PlayerHeadlineStats'])
```

Ilustración 11-1 Endpoint estadísticas generales jugadores



Y este sería el procesamiento de los datos:

```
listaPlayersInfo = []

for i in range(0, 540):
    playerId = playersStats_data['rowSet'][i][0] #i

    objPlayersInfo = CommonPlayerInfo(playerId)
    objPlayersInfo.load_response()
    playersInfo_string = objPlayersInfo.get_response()
    playersInfo_dictionary = json.loads(playersInfo_string)

    playerName = playersInfo_dictionary['resultSets'][0]['rowSet'][0][3]
    country = playersInfo_dictionary['resultSets'][0]['rowSet'][0][9]
    height = playersInfo_dictionary['resultSets'][0]['rowSet'][0][11]
    weight = playersInfo_dictionary['resultSets'][0]['rowSet'][0][12]
    season_exp = playersInfo_dictionary['resultSets'][0]['rowSet'][0][13]
    jersey = playersInfo_dictionary['resultSets'][0]['rowSet'][0][14]
    position = playersInfo_dictionary['resultSets'][0]['rowSet'][0][15]

    playerInfo = {
        'playerId': playerId,
        'playerName': playerName,
        'country': country,
        'height': height,
        'weight': weight,
        'season_exp': season_exp,
        'jersey': jersey,
        'position': position
    }

    listaPlayersInfo.append(playerInfo)

df = pd.DataFrame(listaPlayersInfo)
df.to_csv('playersInfo2.csv')
```

Ilustración 11-2 Procesamiento datos generales de los jugadores

Como se puede ver, filtramos los datos que deseamos, creando una lista de objetos con estos datos y generamos un fichero CSV con la siguiente estructura:

playerId	playerName	country	height	weight	season_exp	jersey	position
1630639	A.J. Lawson	Canada	06-jun	179	0	9	Guard
1631260	AJ Green	USA	06-may	190	0	20	Guard
1631100	AJ Griffin	USA	06-jun	220	0	14	Forward
203932	Aaron Gordon	USA	06-ago	235	8	50	Forward
1628988	Aaron Holiday	USA	6-0	185	4	3	Guard
1630174	Aaron Nesmith	USA	06-may	215	2	23	Guard-Forward
1630598	Aaron Wiggins	USA	06-may	190	1	21	Guard
1629678	Admiral Schofield	United Kingdom	06-may	241	2	25	Forward
201143	Al Horford	Dominican Repub	06-sep	240	15	42	Center-Forward
202692	Alec Burks	USA	06-jun	214	11	5	Guard
1630197	Aleksej Pokusevski	Serbia	7-0	210	2	17	Forward
1627936	Alex Caruso	USA	06-may	186	5	6	Guard
203458	Alex Len	Ukraine	07-feb	250	9	25	Center
1628993	Alize Johnson	USA	06-ago	212	4		Forward
1631214	Alondes Williams	USA	06-abr	210	0		Guard
1630578	Alperen Sengun	Turkey	06-nov	243	1	28	Center
1629599	Amir Coffey	USA	06-jul	210	3	7	Guard-Forward
203083	Andre Drummond	USA	06-nov	279	10	3	Center
2738	Andre Iguodala	USA	06-jun	215	18	9	Guard-Forward
1629614	Andrew Nembhard	Canada	06-mar	191	0	2	Guard-Forward
203952	Andrew Wiggins	Canada	06-jul	197	8	22	Forward
1629014	Anfernee Simons	USA	06-mar	181	4	1	Guard
203076	Anthony Davis	USA	06-oct	253	10	3	Forward-Center
1630162	Anthony Edwards	USA	06-abr	225	2	1	Guard
1630264	Anthony Gill	USA	06-ago	230	2	16	Forward
1630237	Anthony Lamb	USA	06-jun	227	2	40	Forward
1630559	Austin Reaves	USA	06-may	197	1	15	Guard
203085	Austin Rivers	USA	06-abr	200	10	25	Guard
1630245	Ayo Dosunmu	USA	06-may	200	1	12	Guard
1628389	Bam Adebayo	USA	06-sep	255	5	13	Center-Forward
1627732	Ben Simmons	Australia	06-oct	240	4	10	Guard-Forward
1631097	Benedict Mathurin	Canada	06-may	210	0	0	Guard-Forward
202687	Bismack Biyombo	DRC	06-ago	255	11	18	Center
201933	Blake Griffin	USA	06-sep	250	12	91	Forward
1631104	Blake Wesley	USA	06-abr	185	0	14	Guard
1626246	Boban Marjanovic	Serbia	07-abr	290	7	51	Center
1626171	Bobby Portis	USA	06-nov	250	7	9	Forward

Ilustración 11-3 Fichero CSV con datos generales de los jugadores

Ahora procederíamos a la carga de estos datos en la base de datos.

11.4.2 Base de Datos

La carga de los datos, previamente procesados, en la base de datos es muy sencilla. El entorno utilizado, MySQL Workbench permite la importación directa de ficheros CSV. Para ello, previamente se ha de haber creado la entidad correspondiente con los datos necesarios.

La entidad del jugador contiene más datos de los que se pueden ver en el ejemplo porque se dispone de otro procesamiento de datos que se almacenará en la misma entidad.



The screenshot shows the MySQL Workbench interface. The left pane displays the 'SCHEMAS' tree, with 'nbaanalytics_db' expanded to show its 'Tables' (clasificationrow, gameelem, league, player, teambyyear). The 'player' table is selected. The right pane shows the 'Information' tab for the 'player' table, with the 'Columns' section expanded. The columns listed are: playerId (int PK), playerName (varchar(45)), nickname (varchar(45)), teamId (int), teamAbbrev (varchar(45)), age (double), gp (int), won (int), lost (int), wonPct (double), fgm (int), fga (int), fgPct (double), fg3M (int), fg3A (int), fg3Pct (double), ftm (int), fta (int), ftPct (double), oreb (int), dreb (int), reb (int), ast (int), tov (int), stl (int), blk (int), blka (int), pf (int), pts (int), min (double), country (varchar(45)), height (varchar(45)), weight (varchar(45)), seasonExp (int), jersey (varchar(45)), and position (varchar(45)).

Table: player**Columns:**

playerId	int PK
playerName	varchar(45)
nickname	varchar(45)
teamId	int
teamAbbrev	varchar(45)
age	double
gp	int
won	int
lost	int
wonPct	double
fgm	int
fga	int
fgPct	double
fg3M	int
fg3A	int
fg3Pct	double
ftm	int
fta	int
ftPct	double
oreb	int
dreb	int
reb	int
ast	int
tov	int
stl	int
blk	int
blka	int
pf	int
pts	int
min	double
country	varchar(45)
height	varchar(45)
weight	varchar(45)
seasonExp	int
jersey	varchar(45)
position	varchar(45)

Ilustración 11-4 Entidad Player

La importación es tan sencilla como dar click derecho sobre la entidad player y seleccionar el fichero CSV configurado.

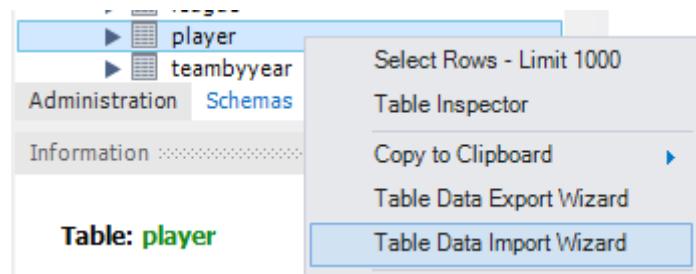


Ilustración 11-5 Importación datos a base de datos

11.4.3 Backend

Este subsistema es el encargado de obtener los datos de la base de datos y gestionarlos o aplicarles la lógica que se deseé. Para cada módulo tendremos el modelo, que tendrá los mismos atributos que la entidad de la base de datos para almacenar la información, un controlador que define el endpoint de nuestra REST API para poder obtener la información y un servicio y mapeador para el flujo de dicha información.

Este sería el modelo de jugador que define todos los atributos de la base de datos así como getters y setters para cada uno.

```
public class Player extends CommonStats {

    private int playerId;
    private String playerName;
    private String nickname;
    private String teamAbbrev;
    private double age;
    private int gp;
    private int blka;
    private int min;
    private int won;
    private int lost;
    private double wonPct;
    private String country;
    private String height;
    private String weight;
    private int seasonExp;
    private String jersey;
    private String position;

    public int getPlayerId() {
        return playerId;
    }

    public void setPlayerId(int playerId) {
        this.playerId = playerId;
    }

    public String getPlayerName() {
        return playerName;
    }
}
```

Ilustración 11-6 Modelo Jugador

```

@RestController
@RequestMapping("/rest")
public class PlayerController {

    @Autowired
    PlayerService psService;

    /**
     * Metodo que inicia la transaccion para obtener la informacion general de todos los jugadores
     * @return todos los jugadores y sus informacion general
     */
    @GetMapping("/playersinfo")
    public List<Player> getPlayersInfo() {
        List<Player> playersInfo = psService.findAllPlayersInfo();

        for (Player player : playersInfo) {
            player.setHeight(getMetros(player.getHeight()));
            player.setWeight(getKG(player.getWeight()));
        }

        return playersInfo;
    }
}

```

Ilustración 11-7 Controlador jugadores

Como se ha dicho el controlador define el endpoint base con `@RequestMapping` y el endpoint final con `@GetMapping` en el caso mostrado. Es el encargado de aplicar la lógica necesaria, en este ejemplo se convierte la altura de pies a metro y el peso de libras a kilos, y devuelve la información al cliente que la haya solicitado.

```

public interface PlayerService {

    /**
     * Metodo que se comunica con el mapper para obtener los datos
     * de los jugadores y se lo devuelve al controlador
     * @return todos los jugadores y sus estadisticas avanzadas
     */
    List<Player> findAllPlayersStats();

    /**
     * Metodo que se comunica con el mapper para obtener los datos
     * del jugador especificado y se lo devuelve al controlador
     * @param playerId
     * @return estadisticas avanzadas del jugador especificado
     */
    Player findPlayerStatsById(int playerId);

    /**
     * Metodo que se comunica con el mapper para obtener los datos generales
     * de los jugadores y se lo devuelve al controlador
     * @return datos generales de todos los jugadores
     */
    List<Player> findAllPlayersInfo();
}

```

Ilustración 11-8 Servicio jugadores (interfaz)

El controlador se comunica con el servicio para obtener los datos, pero con quien se comunica directamente es con la interfaz del servicio, no con la implementación, y esto es para conseguir la independencia entre las distintas capas.



```
@Service
public class PlayerServiceImpl implements PlayerService {

    @Autowired
    PlayerMapper psMapper;

    @Override
    public List<Player> findAllPlayersStats() {
        return psMapper.findAllPlayersStats();
    }

    @Override
    public List<Player> findAllPlayersInfo(){
        return psMapper.findAllPlayersInfo();
    }
}
```

Ilustración 11-9 Servicio jugadores (implementación)

Por último, el servicio se comunica con el mapeador, interfaz que define los métodos finales. Estos métodos serán referenciados desde el correspondiente mapeador XML que define la consulta a realizar a la base de datos, obteniendo de ella la información y devolviéndosela al servicio para que a su vez este se la envíe al controlador.

```
@Mapper
public interface PlayerMapper {

    /**
     * Metodo que extrae los datos de todos los jugadores de la base de datos
     * @return todos los jugadores y sus estadísticas avanzadas
     */
    List<Player> findAllPlayersStats();

    /**
     * Metodo que extrae los datos del jugador especificado de la base de datos
     * @param playerId
     * @return estadísticas avanzadas del jugador especificado
     */
    Player findPlayerStatsById(int playerId);

    /**
     * Metodo que extrae los datos generales de todos los jugadores de la base de datos
     * @return datos generales de todos los jugadores
     */
    List<Player> findAllPlayersInfo();

    /**
     * Metodo que extrae todos los jugadores de un equipo de la base de datos
     * @param teamId
     * @return todos los jugadores del equipo especificado
     */
    List<Player> findPlayersInfoByTeamId(int teamId);
}
```

Ilustración 11-10 Mapeador jugadores (interfaz)

```

-//mybatis.org//DTD Mapper 3.0//EN (doctype)
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "https://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.example.demo.players.mapper.PlayerMapper">
  <select id="findAllPlayersStats" resultType="com.example.demo.players.model.Player">
    SELECT *
    FROM player
  </select>

  <select id="findPlayerStatsById" resultType="com.example.demo.players.model.Player">
    SELECT *
    FROM player
    WHERE playerId = #{playerId}
  </select>

  <select id="findAllPlayersInfo" resultType="com.example.demo.players.model.Player">
    SELECT *
    FROM player
    ORDER BY teamAbbrev
  </select>

  <select id="findPlayersInfoByTeamId" resultType="com.example.demo.players.model.Player">
    SELECT *
    FROM player
    WHERE teamId = #{teamId}
  </select>
</mapper>

```

Ilustración 11-11 Mapeador jugadores (fichero XML)

Luego se realizan las clases de pruebas necesarias para el modelo, el controlador y el servicio.

```

class PlayerTest {

    @Test
    void test() {
        Player player1 = new Player();
        callSets(player1);
        callGets(player1);

        Player player2 = new Player();
        EqualsVerifier.simple().forClass(Player.class)
            .suppress(Warning.STRICT_INHERITANCE, Warning.ALL_NONFINAL_FIELDS_SHOULD_BE_USED, Warning.INHERITED_DIRECTLY_FROM_OBJECT)
            .withRedefinedSuperclass()
            .withPrefabValues(Player.class, player1, player2)
            .verify();
    }

    private void callSets(Player player) {
        player.setPlayerId(1);
        player.setPlayerName("pepe");
        player.setNickname("nickname");
        player.setTeamId(1);
        player.setTeamAbbrev("abbrev");
    }
}

```

Ilustración 11-12 Pruebas modelo



```
@MockitoSettings(strictness = Strictness.LENIENT)
@RunWith(SpringRunner.class)
@SpringBootTest
class PlayerControllerTest {

    @InjectMocks
    private final PlayerController controller = new PlayerController();
    @Mock
    private PlayerService service;

    @Test
    void getPlayerStatsTest() {
        when(service.findAllPlayersStats()).thenReturn(new ArrayList<>());

        assertNotNull(controller.getPlayersStats());
    }

    @Test
    void getPlayerStatsByIdTest() {
        when(service.findPlayerStatsById(1)).thenReturn(new Player());

        assertNotNull(controller.getPlayerStatsById(1));
    }

    @Test
    void getPlayersInfoTest() {
        Player playerMock = new Player();
        playerMock.setHeight("6.6");
        playerMock.setWeight("200");
        ArrayList<Player> listMock = new ArrayList<>();
        listMock.add(playerMock);

        when(service.findAllPlayersInfo()).thenReturn(listMock);

        assertNotNull(controller.getPlayersInfo());
    }
}
```

Ilustración 11-13 Pruebas controlador

```

@MockitoSettings(strictness = Strictness.LENIENT)
class PlayerServiceTest {

    @InjectMocks
    private PlayerServiceImpl service;

    @Mock
    private PlayerMapper mapper;

    @Test
    void findAllPlayersStatsTest() {
        when(mapper.findAllPlayersStats()).thenReturn(new ArrayList<>());

        assertNotNull(service.findAllPlayersStats());
    }

    @Test
    void findAllPlayersInfoTest() {
        when(mapper.findAllPlayersInfo()).thenReturn(new ArrayList<>());

        assertNotNull(service.findAllPlayersInfo());
    }

    @Test
    void findPeriodsOfTeamInGameTest() {
        when(mapper.findPlayerStatsById(1)).thenReturn(new Player());

        assertNotNull(service.findPlayerStatsById(1));
    }

    @Test
    void findPlayersInfoByTeamIdTest() {
        when(mapper.findPlayersInfoByTeamId(1)).thenReturn(new ArrayList<>());

        assertNotNull(service.findPlayersInfoByTeamId(1));
    }
}

```

Ilustración 11-14 Pruebas servicio

11.4.4 Frontend

Por último, la Webapp. Concretamente para el apartado de jugadores creamos un módulo, al igual que con los equipos y los partidos, con su correspondiente módulo encargado de definir las rutas.

```

import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { PlayersComponent } from './players.component';
import { PlayerDetailComponent } from './player-detail/player-detail/player-detail.component';
import { PlayerCardComponent } from './player-card/player-card.component';
import { ComparativePlayersDialogComponent } from '../../comparativeDialog/comparative-dialog/co
import { FilterPlayersPipe } from 'src/app/pipes/playerPipes/filterPlayers.pipe';
import { FormsModule } from '@angular/forms';
import { MatTooltipModule } from '@angular/material/tooltip';
import { RouterModule } from '@angular/router';
import { MatIconModule } from '@angular/material/icon';
import { PlayersRoutingModule } from './players-routing.module';

```



```

@NgModule({
  declarations: [
    PlayersComponent,
    PlayerDetailComponent,
    PlayerCardComponent,
    ComparativePlayersDialogComponent,
    FilterPlayersPipe
  ],
  imports: [
    CommonModule,
    FormsModule,
    MatTooltipModule,
    RouterModule,
    MatIconModule,
    PlayersRoutingModule
  ]
})
export class PlayersModule { }

```

Ilustración 11-15 Módulo de jugadores

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { PlayersComponent } from './players.component';
import { PlayerDetailComponent } from './player-detail/player-detail/player-detail.component';

const routes: Routes =
[
  { path: 'players', component: PlayersComponent, pathMatch: 'prefix' },
  { path: 'playerDetail', component: PlayerDetailComponent },
];

@NgModule({
  imports: [
    RouterModule.forRoot(routes)
  ],
  exports: [RouterModule]
})
export class PlayersRoutingModule { }

```

Ilustración 11-16 Módulo de rutas de los jugadores



Es necesario también tener un modelo para el jugador en este subsistema, que será el encargado de almacenar los datos extraídos.

```
export class Player{
  playerId!: number;
  playerName!: string;
  nickname!: string;
  teamId!: number;
  teamAbbrev!: string;
  age!: number;
  gp!: number;
  won!: number;
  lost!: number;
  wonPct!: number;
  fgm!: number;
  fga!: number;
  fgPct!: number;
  fg3m!: number;
  fg3a!: number;
  fg3Pct!: number;
  ftm!: number;
  fta!: number;
  ftPct!: number;
  oreb!: number;
  dreb!: number;
  reb!: number;
  ast!: number;
  tov!: number;
  stl!: number;
  blk!: number;
  blka!: number;
  pf!: number;
  pts!: number;
  min!: number;
  urlPicture!: string;
```

Ilustración 11-17 Modelo Jugador

El componente o clase encargado de realizar la petición es el componente principal de los jugadores.



```
import { Component, OnDestroy, OnInit } from '@angular/core';
import { DataService } from 'src/app/services/data-service.service';
import { HomeService } from '../../services/home.service';
import { TooltipPosition } from '@angular/material/tooltip';
import { FormControl } from '@angular/forms';
import { Subscription } from 'rxjs';
import { Player } from 'src/app/models/player';

@Component({
  selector: 'app-players',
  templateUrl: './players.component.html',
  styleUrls: ['./players.component.scss']
})
export class PlayersComponent implements OnInit, OnDestroy {

  playersInfo: Player[] = [];
  filterData = '';
  positionOptions: TooltipPosition[] = ['above'];
  position = new FormControl(this.positionOptions[0]);
  playerInfoSubscription!: Subscription;
  emptyData = true;

  constructor(private dataService: DataService,
    private readonly homeService: HomeService){}

  ngOnDestroy(): void {
    this.playerInfoSubscription.unsubscribe();
  }

  ngOnInit(): void {
    this.playerInfoSubscription = this.dataService.getPlayersInfo().subscribe(
      (resp: any) => {
        this.playersInfo = JSON.parse(resp);
        if(this.playersInfo.length > 0){
          this.emptyData = false;
        } else {
          this.emptyData = true;
        }
      }
    );
  }
}
```

Ilustración 11-18 Componente jugadores

Pero no lo hace directamente, sino que existe un servicio encargado de definir todas las peticiones necesarias a la REST API, y que va a ser quien reciba en una primera instancia la información para que el componente principal la pueda recoger.

```

export class DataService {

    private scoreboardUrl = 'http://localhost:8080/rest/scoreboard/clasification';
    private leagueUrl = 'http://localhost:8080/rest/league';
    private teamsStatsUrl = 'http://localhost:8080/rest/teamstats';
    private teamsStatsByYearUrl = 'http://localhost:8080/rest/teamstatsbyyear';
    private playersInfoUrl = 'http://localhost:8080/rest/playersinfo';
    private playerStatsById = 'http://localhost:8080/rest/playerstatsbyid';
    private gamesStatsUrl = 'http://localhost:8080/rest/gamesstatsbydate';
    private teamTemplate = 'http://localhost:8080/rest/teamTemplate';

    constructor(private http: HttpClient) { }

    getPlayersInfo(){
        return this.http.get(this.playersInfoUrl, {responseType: 'text'});
    }
}

```

Ilustración 11-19 Servicio para peticiones

Luego, cada módulo o componente, además de su clase principal tienen un fichero HTML y otro CSS para definir el aspecto gráfico.

```

<input type="text" class="form-control" placeholder="Buscar jugador..." name = "filterData" [(ngModel)]="filterData">
<div *ngIf="emptyData" class="noContent">NO ES POSIBLE ENCONTRAR A LOS JUGADORES</div>
<div *ngIf="!emptyData" class="tableContainer">
    <div class="cardsContainer row">
        <ng-container *ngIf="(playersInfo | filterPlayer:filterData).length > 0; else noResults">
            <app-player-card
                *ngFor="let player of playersInfo | filterPlayer:filterData"
                [player]="player"
                class="col-sm-12 col-md-6 col-xl-4"
            >
            </app-player-card>
        </ng-container>
        <ng-template #noResults>
            <div class="noContent">NO SE ENCUENTRAN COINCIDENCIAS</div>
        </ng-template>
    </div>
</div>

```

Ilustración 11-20 Fichero HTML Jugadores



```
.tableContainer{  
    position: relative;  
    width: 100%;  
    margin: 0;  
    height: 91%;  
    top: 2.5%;  
}  
  
.table{  
    position: relative;  
    top: 1%;  
    margin-left: 2%;  
    width: 95%;  
    height: 100%;  
    overflow: auto;  
    color: black;  
    font-size: 18px;  
}  
  
.tableData{  
    width: 100%;  
    text-align: center;  
}  
}  
  
th{  
    background-color: black;  
    font-weight: 500;  
}  
  
td{  
    background-color: black;  
}  
  
td, th{  
    border: solid;  
    border-color: black;  
    border-width: 2px;  
    padding: 20px;  
}  
  
a{  
    text-decoration: none;  
    color: black;  
}
```

Ilustración 11-21 Fichero CSS Jugadores

	ESCUELA DE INGENIERÍA INFORMÁTICA		SISTEMA DE REPORTING Y ESTADÍSTICA AVANZADA	Página 203 de 203 Alejandro Galán Freire UO277346@uniovi.es
--	--------------------------------------	--	--	---

Por último, se dispone de la clase encargada de realizar las pruebas de todo el componente de jugadores.

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { HttpClientModule } from '@angular/common/http';
import { PlayersComponent } from './players.component';
import { FilterPlayersPipe } from 'src/app/pipes/playerPipes/filterPlayers.pipe';
import { FormsModule } from '@angular/forms';

describe('PlayersComponent', () => {
  let component: PlayersComponent;
  let fixture: ComponentFixture<PlayersComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [
        PlayersComponent,
        FilterPlayersPipe
      ],
      imports: [
        HttpClientModule,
        FormsModule
      ]
    })
    .compileComponents();
  });

  fixture = TestBed.createComponent(PlayersComponent);
  component = fixture.componentInstance;
  fixture.detectChanges();
});

it('should create', () => {
  expect(component).toBeTruthy();
});
```

Ilustración 11-22 Clase de pruebas Jugadores