



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Bappsket: Herramienta para la administración de un club y equipos de baloncesto

Autor:
Sergio Casado García

Tutora:
Yania Crespo González-Carvajal

Agradecimientos

A mi familia, sin cuyo apoyo, paciencia y esfuerzo nunca hubiese llegado hasta aquí.

A Alba, por haber estado siempre a mi lado y haberme ayudado a levantarme después de cada caída en este largo camino.

Al GCC (Gran Comando Cafetería), que con su apoyo y su ayuda hicieron que este viaje de aprendizaje fuese increíble e hicieron que nunca tirase la toalla a lo largo de estos años.

A mis compañeros de Luce IT y de la Consejería de Agricultura y Ganadería por apoyarme en este proceso durante el año y medio que llevo trabajando con ellos.

A la Escuela de Ingeniería Informática en su conjunto, por ofrecernos todas las oportunidades que hemos tenido durante estos años y que después de tanto tiempo se ha acabado convirtiendo en una segunda casa.

Resumen

El objetivo de este Trabajo de Fin de Grado es el desarrollo de una aplicación web que permita principalmente dos cosas: por un lado la administración de un club de baloncesto desde el punto de vista de un director deportivo y, por otro lado, la administración de un equipo de baloncesto desde el punto de vista de un entrenador.

Abstract

The objective of this Final Degree Project is the development of a web application that allows mainly two things: on the one hand the administration of a basketball club from the point of view of a sports director and, on the other hand, the administration of a basketball team from the point of view of a coach.

Índice general

1. Introducción	17
1.1. Motivación	17
1.2. Objetivos	17
1.3. Estructura de la memoria	18
2. Planificación	19
2.1. Alcance, propósito y objetivos	19
2.2. Plan de proceso	19
2.3. Vista general	19
2.4. Fases e iteraciones	20
2.5. Gestión del proceso	22
2.5.1. Plan de puesta en marcha	22
2.6. Plan de control	22
2.6.1. Control de requisitos	22
2.6.2. Control del calendario	22
2.6.3. Plan de gestión de recursos	23
2.6.4. Plan de trabajo	23
2.6.5. Calendarización inicial	27
2.6.6. Presupuesto	28
2.6.7. Plan de gestión de riesgos	29
2.7. Planes de procesos de soporte	31
2.7.1. Plan de gestión de configuraciones	31
3. Tecnologías utilizadas	33
3.1. Estado del arte	33
3.2. Tecnologías web por parte del cliente	34
3.2.1. Navegador web	34
3.2.2. HTML	35
3.2.3. CSS	36
3.2.4. Materialize	36
3.2.5. JavaScript	37
3.2.6. JQuery	38
3.3. Tecnologías web por parte del servidor	39
3.3.1. Java	39
3.3.2. Apache Derby	39

4. Requisitos y Análisis	41
4.1. Introducción	41
4.2. Entorno	41
4.2.1. Conceptos previos	41
4.3. Funcionalidad requerida para la aplicación	42
4.3.1. Requisitos funcionales	42
4.3.2. Requisitos no funcionales	43
4.3.3. Requisitos de información	43
4.4. Roles	45
4.5. Casos de uso	45
4.6. Matriz de trazabilidad de requisitos/Casos de uso	62
4.7. Modelo de dominio	62
4.8. Máquinas de estado	64
4.9. Diagramas de secuencia	65
5. Diseño	71
5.1. Introducción	71
5.2. Arquitectura	71
5.2.1. Patrón Modelo Vista Controlador	71
5.3. Diseño de Base de Datos Relacionales	73
5.3.1. Tipos de consultas	75
5.3.2. Script de la base de datos	75
5.4. Diagrama de despliegue	76
6. Implementación	77
6.1. Clases controlador	77
6.2. Vista	78
6.3. Clases Modelo	81
7. Pruebas	83
8. Conclusiones	91
8.1. Trabajo futuro	91
A. Contenido del CD	95
B. Manual de mantenimiento	97
B.1. Introducción	97
B.2. Requisitos	97
B.2.1. Servidor web	97
B.2.2. Base de datos	99
B.2.3. Código	101
C. Manual de usuario.	103
C.1. Menú de inicio, ingreso y registro	103
C.2. Registrar equipo y modificar un equipo	108
C.3. Añadir y eliminar entrenadores del club.	112
C.4. Registrar, modificar y ver una jornada.	113

C.5. Crear y modificar un entrenamiento 116
C.6. Guardar y modificar la asistencia de los jugadores 120
C.7. Añadir resultado 122
C.8. Añadir nuevo ejercicio 123
C.9. Cambiar club/equipo 124

Índice de figuras

2.1. Arquitectura UPedu. Tomada de su página web [8]	20
2.2. Diagrama de Gantt con la calendarización planificada.	27
2.3. Diagrama de Gantt con la calendarización planificada.	28
3.1. Porcentaje uso navegadores según StatCounter, Global Stats. Imagen tomada de su web [12]	35
3.2. Ejemplo de código HTML	36
3.3. Ejemplo de código CSS	36
3.4. Estructura básica Materialize	37
3.5. Ejemplo de código JavaScript	38
3.6. Ejemplo de código JQuery	38
4.1. Diagrama de Casos de Uso del administrador	46
4.2. Diagrama de casos de uso de un entrenador sin club	47
4.3. Diagrama de casos de uso de un entrenador con club	48
4.4. Matriz trazabilidad	62
4.5. Modelo de dominio	63
4.6. Máquina de estado de un usuario	65
4.7. Diagrama de secuencia del ingreso de un entrenador con club	66
4.8. Diagrama de secuencia de añadir un entrenador	67
4.9. Diagrama de secuencia de eliminar un entrenador	68
5.1. Diagrama de patrón modelo vista controlador.	72
5.2. Composición del modelo la vista y el controlador del proyecto.	73
5.3. Diagrama de bases de datos relacional.	74
5.4. Diagrama de despliegue.	76
6.1. Ejemplo de un servlet	78
6.2. Comparación de Modelos de Aplicación Web clásico y basado en Ajax. Tomada de [7]	79
6.3. Ejemplo de una página HTML con Materialize	80
B.1. Añadimos un nuevo server.	98
B.2. Seleccionamos la carpeta donde tenemos los archivos de GlassFish.	98
B.3. Dejamos la pantalla de dominio tal cual.	99
B.4. Añadimos una nueva base de datos.	100

B.5. Rellenamos los datos	100
B.6. Abrimos la página para editar SQL	100
B.7. Ejecutamos Script	101
C.1. Pagina de login	104
C.2. Pantalla principal de un administrador	105
C.3. Pantalla elección club/equipo	106
C.4. Pantalla principal Entrenador	106
C.5. Pantalla de registro	107
C.6. Pantalla de añadir nombre al club	108
C.7. Pantalla de crear equipo	109
C.8. Pantalla de añadir jugadores	110
C.9. Pantalla modificar equipo	111
C.10.Pantalla de modificar jugadores	111
C.11.Pantalla de añadir un entrenador	112
C.12.Pantalla de eliminar un entrenador	113
C.13.Pantalla de crear una jornada	114
C.14.Pantalla de modificar una jornada	115
C.15.Pantalla de ver una jornada	116
C.16.Pantalla de elegir fecha para entrenamiento	117
C.17.Pantalla de elegir ejercicio	118
C.18.Pantalla de descripción de ejercicio	119
C.19.Pantalla de resumen de entrenamiento	120
C.20.Pantalla de añadir asistencia	121
C.21.Pantalla de modificar asistencia	122
C.22.Pantalla de añadir resultado	123
C.23.Pantalla de añadir un nuevo ejercicio	124
C.24.Pantalla de cambiar club/equipo	125

Índice de cuadros

2.1. Cálculo presupuesto del proyecto	29
2.2. Matriz Impacto/Probabilidad	29
4.1. Caso de uso : Registrarse	49
4.2. Caso de uso : Ingresar	50
4.3. Caso de uso : Registrar equipo	51
4.4. Caso de uso : Modificar equipo	52
4.5. Caso de uso : Añadir entrenador	52
4.6. Caso de uso : Eliminar entrenador	53
4.7. Caso de uso : Registrar jornada	53
4.8. Caso de uso : Modificar jornada	54
4.9. Caso de uso : Descargar resultados	55
4.10. Caso de uso : Crear entrenamiento	56
4.11. Caso de uso :Modificar un entrenamiento	57
4.12. Caso de uso :Añadir nuevo ejercicio a la base de datos	58
4.13. Caso de uso :Registrar asistencia	58
4.14. Caso de uso :Modificar asistencia	59
4.15. Caso de uso :Modificar un resultado	60
4.16. Caso de uso :Ver resultado	60
4.17. Caso de uso :Cambiar club/equipo	61
4.18. Caso de uso : Descargar asistencia	62
7.1. CP-01: Añadir entrenador	83
7.2. CP-02: Eliminar entrenador	83
7.3. CP-013: Añadir entrenador ya existente	84
7.4. CP-04: Crear equipo con mismo 1er y 2º entrenador	84
7.5. CP-05: Crear equipo	84
7.6. CP-06: Crear equipo	85
7.7. CP-07: Modificar equipo	86
7.8. CP-08: Crear jornada	86
7.9. CP-09: Crear jornada	87
7.10. CP-10: Modificar una jornada	87
7.11. CP-11: Modificar una jornada	87
7.12. CP-12: Añadir asistencia	87
7.13. CP-13: Modificar asistencia	88
7.14. CP-14: Crear entrenamiento	88

7.15. CP-15: Modificar entrenamiento	88
7.16. CP-16: Añadir un comentario	89
7.17. CP-17: Crear entrenamiento	89
7.18. CP-18: Crear entrenamiento	89
7.19. CP-19: Ver resultados	89
7.20. CP-20: Ver resultados	90
7.21. CP-21: Cambiar club/equipo trabajo	90

Capítulo 1

Introducción

1.1. Motivación

Desde los cinco años he desarrollado de forma continuada en el club deportivo San Agustín la actividad de baloncesto. A partir de los dieciocho años, además de ser solo un jugador, empecé a formar parte del grupo de entrenadores del club. Durante estos últimos seis años he ido observando carencias que tanto yo como mis compañeros teníamos a la hora de administrar a nuestros equipos. Durante varios años, para intentar mejorar y tener todo más ordenado, utilizamos unos cuadernos que nosotros mismos hacíamos donde apuntábamos los entrenamientos, nuestros jugadores y sus asistencias a los entrenamientos, así como apuntes a cada ejercicio que se realizaba. Esto hizo que mejorásemos en la forma de administrar nuestros equipos pero muchas veces este cuaderno se acababa estropeando o perdiendo, ocasionando la pérdida de mucho trabajo y mucha información acerca de nuestro equipo.

De la misma forma durante este tiempo he observado cómo la administración del club muchas veces no tenía los datos de cada equipo organizados correctamente y, adicionalmente, el encargado de subir los resultados a la web del colegio en el que se desarrolla el mencionado club deportivo, tenía que obtener dichos resultados de un grupo de Whatsapp donde todos poníamos nuestros resultados.

Todo esto, y el afán de mejorar como entrenador, así como las ganas de facilitar las cosas a mis compañeros generaron la idea de desarrollar una aplicación para administrar tanto nuestros equipos como el club de una forma mas eficiente.

1.2. Objetivos

El objetivo principal es desarrollar una aplicación que mejore la gestión y administración tanto del club desde el punto de vista del director deportivo, así como la administración y el día a día de un equipo de baloncesto desde el punto de vista de un entrenador.

Como subobjetivos específicos se pueden reseñar:

- La descarga de resultados en archivo EXCEL fácil de consultar.
- La utilización de dicha aplicación sin la necesidad de pertenecer a ningún club.

- Aplicar los conocimientos obtenidos durante el desarrollo del grado de ingeniería informática.

1.3. Estructura de la memoria

Esta memoria está estructurada por capítulos. En el primer capítulo se realiza una breve introducción del trabajo realizado durante el desarrollo del proyecto, donde explicamos lo que ha motivado este proyecto y los objetivos que queremos cumplir con su desarrollo. En el segundo capítulo se explicará la planificación del proyecto y la metodología que se ha utilizado para el desarrollo de éste. En el tercer capítulo se detallarán las tecnologías que han sido utilizadas para el desarrollo de este proyecto. El cuarto capítulo está dedicado a la explicación del análisis que se ha realizado del proyecto, explicando los requisitos obtenidos, los casos de uso, el modelo de dominio, las máquinas de estado y algunos diagramas de secuencia. En el quinto capítulo, explicaremos todo lo relacionado con el diseño y la arquitectura del sistema, el diagrama de despliegue y el diseño de la base de datos. En el sexto se explicará la implementación y se explicarán aspectos relevantes sobre ésta. En el séptimo capítulo se expondrán las pruebas realizadas al proyecto una vez el desarrollo de este fue terminado. En el octavo capítulo se expondrán las conclusiones del proyecto. Después en el primer anexo tendremos una explicación del contenido del CD. En el segundo anexo tendremos un manual para el mantenimiento de la aplicación. Por último en el tercer anexo tendremos el manual de usuario de la aplicación.

Capítulo 2

Planificación

A continuación se expondrá la planificación del proyecto. Es necesario realizar una planificación para determinar el orden de las tareas y poder estimar el tiempo que supondrá la realización de cada una de ellas antes de empezar con la parte de análisis, diseño e implementación del proyecto. También hay que evaluar los riesgos y realizar un plan de gestión de estos para poder evitarlos o minimizar su impacto. Por último realizaremos el cálculo del presupuesto del proyecto.

2.1. Alcance, propósito y objetivos

El objetivo de este proyecto es elaborar una aplicación web que facilite la administración de un club y de sus diferentes equipos ya sea como director deportivo o como entrenador.

2.2. Plan de proceso

Para el desarrollo de este proyecto se ha optado por utilizar la metodología de UPedu [8], que es un proceso de ingeniería de software que nos proporciona un enfoque disciplinado para asignar tareas y responsabilidades de una organización de desarrollo. Su objetivo es garantizar la producción de software de alta calidad que satisfaga las necesidades de los usuarios dentro de un calendario y un presupuesto predecible.

2.3. Vista general

La arquitectura general de UPedu se puede ver en la siguiente figura:

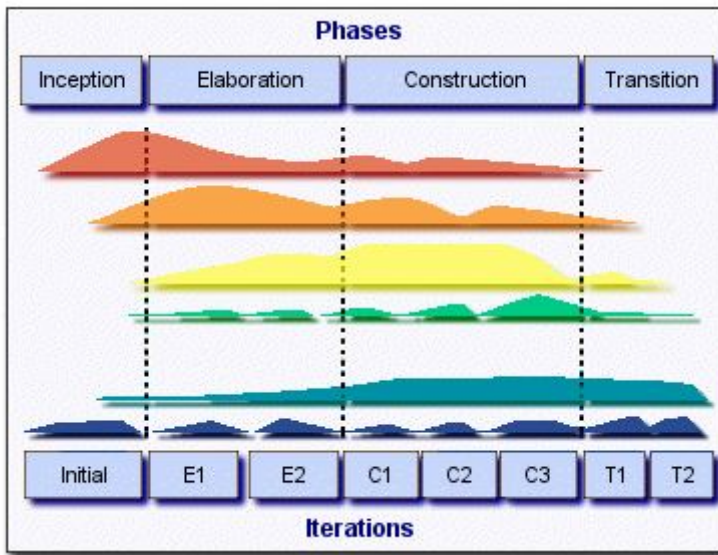


Figura 2.1: Arquitectura UPedu. Tomada de su página web [8]

En la foto el eje horizontal representa el tiempo y muestra los aspectos del ciclo de vida del proceso a medida que se desarrolla, mientras que el eje vertical representa los flujos de trabajo del proceso central.

2.4. Fases e iteraciones

- Inicial:

El objetivo de la fase de inicio es lograr la concurrencia entre todos los interesados sobre los objetivos del ciclo de vida del proyecto. La fase de inicio es importante principalmente en nuevos desarrollos donde hay importantes riesgos comerciales y requisitos que deben abordarse antes de que el proyecto pueda continuar. Por otro lado, en proyectos basados en las mejoras de un sistema, esta fase es mucho más breve, pero se sigue centrando en la viabilidad del proyecto.

Los objetivos principales de esta fase incluyen:

1. Establecer el alcance del software y las condiciones de contorno.
2. Discriminar los casos de uso crítico del sistema.
3. Estimar el coste general y el cronograma del proyecto.
4. Estimar los riesgos potenciales.
5. Preparar el entorno de apoyo para el proyecto.

- Elaboración:

El objetivo de esta fase es proporcionar una base estable para la mayor parte del esfuerzo de diseño e implementación y para la fase de construcción. La arquitectura evoluciona a

partir de de una consideración de los requisitos más importantes y de una evaluación de riesgo.

Los objetivos principales de esta fase incluyen:

1. Garantizar que la arquitectura, los requisitos y los planes sean lo suficientemente estables y los riesgos lo suficientemente mitigados para poder predecir el coste y el cronograma del proyecto.
2. Abordar todos los riesgos arquitectónicamente significativos.
3. Producir un prototipo evolutivo de componentes de calidad de producción, así como uno o más prototipos para . mitigar riesgos específicos.
4. Demostrar que la arquitectura básica soportará los requisitos del sistema.

■ **Construcción:**

El objetivo de esta fase es aclarar los requisitos restantes y completar arquitectura básica. La fase de construcción es en cierto sentido un proceso de fabricación donde se hace hincapié en la gestión de los recursos y el control de las operaciones para optimizar los costes y las calidad.

Los objetivos de esta fase son:

1. Minimizar los costes de desarrollo mediante la optimización de los recursos .
2. Lograr la calidad adecuada tan rápido como sea posible.
3. Lograr versiones útiles lo más rápido posible.
4. Completar el análisis, diseño, desarrollo y prueba de todas las funcionalidades requeridas.
5. Desarrollar de forma iterativa e incremental un producto completo que esté listo.
6. Decidir si el software, los sitios y los usuarios están listos para implementar la aplicación.
7. Lograr cierto grado de paralelismo en el trabajo de los equipos de desarrollo.

■ **Transición:**

El objetivo de la fase de transición es garantizar que el software esté disponible para sus usuarios finales. La fase de transición puede abarcar varias iteraciones incluyendo probar el producto y realizar ajustes menores en función de comentarios de los usuarios (estos comentarios se deben basar en ajustar los problemas del producto, la configuración, la instalación y la usabilidad). Al final de esta fase los objetivos del ciclo de vida deberían haberse cumplido y el proyecto debería de estar en posición de ser cerrado. También puede haber casos en los que esta fase coincida con el comienzo de otro ciclo de vida del producto. En otros casos la transición puede coincidir con la entrega de artefactos a terceros que se serán los responsables de las operaciones, el mantenimiento y las mejoras del sistema entregado.

Los objetivos principales de esta fase son:

1. Probar la beta para validar el nuevo sistema con las expectativas del usuario.

2. Convertir bases de datos operacionales.
3. Entrenamiento de usuario y mantenedores.
4. Actividades de ajuste como la corrección de errores, la mejora de rendimiento y la usabilidad.

2.5. Gestión del proceso

2.5.1. Plan de puesta en marcha

A continuación se muestran los conocimientos básicos tanto de lenguajes como de frameworks y herramientas que se utilizarán para el desarrollo de este proyecto.

- Conocimientos en HTML.
- Conocimientos del lenguaje JavaScript.
- Conocimiento de CSS para el diseño y el estilo de la aplicación web.
- Conocimientos del framework JQuery.
- Conocimientos del framework Materialize.
- Conocimientos básicos de AJAX.
- Conocimientos básicos de Glassfish Server.
- Conocimientos y soltura en el gestor de bases de datos Apache Derby.

2.6. Plan de control

A continuación se muestran las actividades de control.

2.6.1. Control de requisitos

Si se produjeran cambios respecto a los requisitos obtenidos inicialmente, se tendría que reevaluar el posible impacto que estos cambios podrían causar sobre el normal desarrollo del proyecto y sobre la planificación realizada en primera instancia, por lo que esto debe estar contemplado en el plan de riesgos. Como ya se ha comentado, en caso de que estos nuevos requisitos se acepten, habría que realizar una remodelación de la planificación. En el caso de este proyecto los requisitos se han mantenido iguales desde el principio.

2.6.2. Control del calendario

Para el control del calendario se utilizará una herramienta que permita gestionar las actividades ya realizadas y las faltan por realizar, para así tener un control sobre el tiempo empleado y ver si se está cumpliendo con lo planificado.

2.6.3. Plan de gestión de recursos

En este proyecto solo hay dos tipos de recursos, humanos y otros.

- Recursos humanos: una persona, que realizará el proyecto, por lo que tendrá los roles de analista, desarrollador y tester sobre la aplicación.
- Otros recursos: una máquina para el desarrollo del proyecto.

2.6.4. Plan de trabajo

A continuación se van a mostrar la descripción de las actividades realizadas.

ID: 01 Comienzo de la fase de inicio
Predecesoras: -
Duración: 1 día
Inicio del grupo de actividades que corresponden a la fase de inicio.
ID: 02 Aprendizaje JavaScript
Predecesoras: 01
Duración: 3 días
Aprendizaje del lenguaje de scripting JavaScript, adquiriendo los conocimientos básicos que nos permitirán el desarrollo del proyecto.
ID: 03 Aprendizaje Materialize
Predecesoras: 01
Duración: 3 días
Aprendizaje del framework Materialize que nos ayudará a la hora de dar formato y estilo a la aplicación viendo sus características principales y sus modos de uso
ID: 04 Aprendizaje JQuery
Predecesoras: 01
Duración: 3 días
Aprendizaje del framework JQuery viendo sus características principales y sus modos de uso
ID: 05 Aprendizaje AJAX
Predecesoras: 01
Duración: 2 días
Aprendizaje del uso de AJAX para poder realizar carga de datos de forma asíncrona en nuestro proyecto.
ID: 06 Definición de las actividades
Predecesoras: 02,03,04,05
Duración: 2 días
Definición de cada una de las actividades que se deben llevar a cabo junto a sus especificaciones para el correcto desarrollo del proyecto. También se estimara la duración de cada una de las actividades.

ID: 07 Planificación de los riesgos
Predecesoras: 06
Duración: 1 día
Estudio y descripción de los posibles riesgos que pueden darse durante el desarrollo de este proyecto. Definir también el plan de actuación para cada riesgo.
ID: 08 Calendarización
Predecesoras: 07
Duración: 1 día
En función de las actividades definidas y del tiempo estimado para el desarrollo de cada una de ellas realizamos un diagrama de Gantt correspondiente a la calendarización de las actividades.
ID: 09 Puesta en marcha control de versiones
Predecesoras: 08
Duración: 1 día
Puesta en marcha del control de versiones para el desarrollo del proyecto.
ID: 10 Obtención de requisitos
Predecesoras: 09
Duración: 3 días
Obtención de los requisitos que tendrá el proyecto junto a sus descripciones a partir de la información que tenemos sobre el proyecto.
ID: 11 Definición de los casos de uso
Predecesoras: 10
Duración: 1 día
A partir de los requisitos y los datos obtenidos del proyecto definir los casos de uso del sistema.
ID: 12 Definición del modelo de dominio
Predecesoras: 11
Duración: 2 días
Tras los datos obtenidos del sistema elaborar el modelo de dominio utilizando lenguaje UML.
ID: 013 Realización de los diagramas de secuencia
Predecesoras: 12
Duración: 3 días
Realización de los diagramas de secuencia de algunos escenarios de la aplicación.

ID: 14 Modelo de despliegue	
Predecesoras:	13
Duración:	1 día
Realización del modelo de despliegue en lenguaje de anotación UML.	
ID: 15 Diseñar estructura de la base de datos	
Predecesoras:	14
Duración:	1 día
Definir el esquema de la Base de Datos teniendo en cuenta los requisitos de información obtenidos.	
ID: 16 Diseño esquema entidad-relación	
Predecesoras:	15
Duración:	1 día
Definir el esquema relacional basándonos en el diseño de la base de datos anteriormente mencionado. Utilizaremos notación UML.	
ID: 17 Script Base de Datos	
Predecesoras:	16
Duración:	2 días
Definir el esquema de la Base de Datos teniendo en cuenta los requisitos de información obtenidos.	
ID: 18 Diseño de la interfaz de usuario	
Predecesoras:	17
Duración:	2 días
Realizar bocetos con las distintas vistas que tendrá la aplicación.	
ID: 19 Diseño de la arquitectura del sistema	
Predecesoras:	18
Duración:	4 días
Diseño de la arquitectura del sistema.	
ID: 20 Desarrollo del ingreso y del registro	
Predecesoras:	19
Duración:	1 día
Desarrollo de las pantallas de ingreso y de registro para la aplicación.	
ID: 21 Desarrollo de la inserción y borrado de entrenadores	
Predecesoras:	20
Duración:	1 día
Desarrollo de las pantallas de adición y la eliminación de un entrenador a un club por parte del administrador.	

ID: 22	Desarrollo de registro y modificaciones de equipo
Predecesoras:	21
Duración:	6 días
Desarrollo de las pantallas para añadir y modificar equipos a un club por parte del administrador o por parte de un entrenador sin club.	

ID: 23	Desarrollo de inserción y modificación de una jornada
Predecesoras:	22
Duración:	3 días
Desarrollo de las pantallas para crear y modificar una jornada con sus partidos correspondientes.	

ID: 24	Desarrollo de descarga, modificación y visualización de resultados
Predecesoras:	23
Duración:	8 días
Desarrollo de las pantallas para la modificación de resultados y la visualización de estos por parte tanto del administrador como de los entrenadores de un club y la posibilidad de descargar los resultados por parte del administrador.	

ID: 25	Desarrollo de creación y modificación de un entrenamiento
Predecesoras:	24
Duración:	8 días
Desarrollo de las pantallas para crear y modificar un entrenamiento por parte de un entrenador.	

ID: 26	Desarrollo de creación de un nuevo ejercicio
Predecesoras:	25
Duración:	1 días
Desarrollo de la pantalla para añadir un nuevo ejercicio por parte de un entrenador.	

ID: 27	Desarrollo de registro y modificación de la asistencia
Predecesoras:	26
Duración:	2 días
Desarrollo de las pantallas de registro y modificación de la asistencia por parte de un entrenador.	

2.6.5. Calendarización inicial

En las siguientes páginas se mostrará la calendarización del proyecto realizada con un diagrama de Gantt.

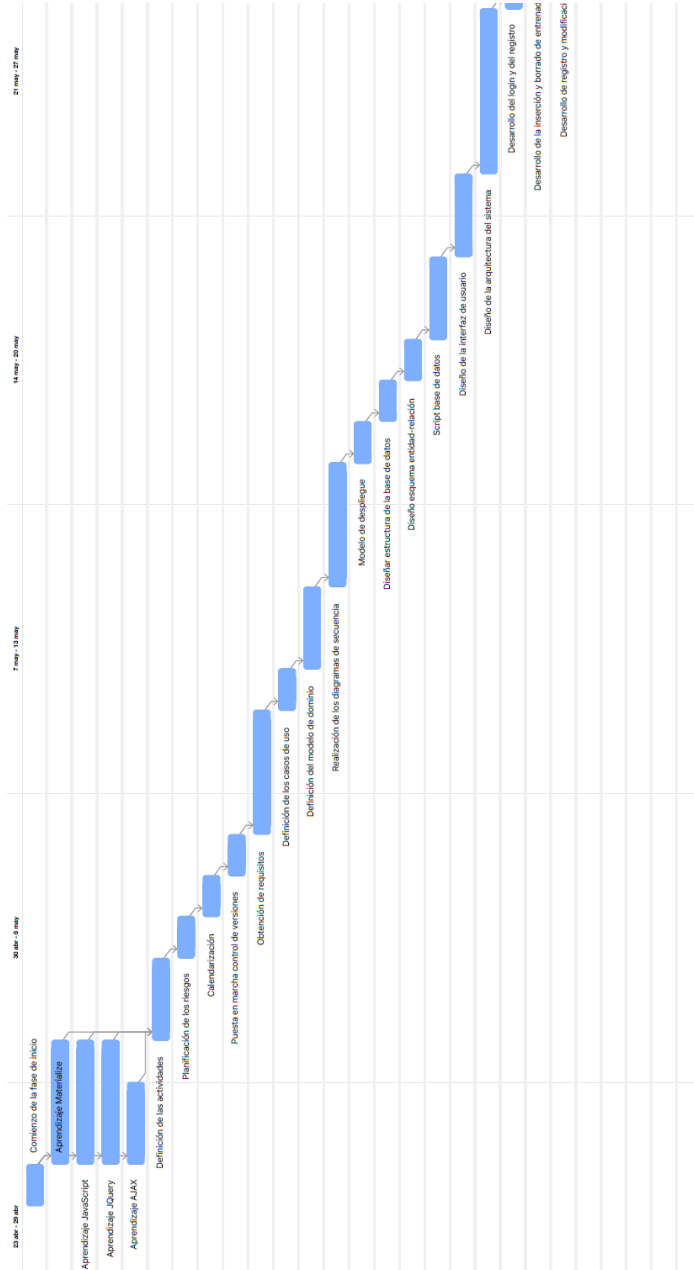


Figura 2.2: Diagrama de Gantt con la calendarización planificada.

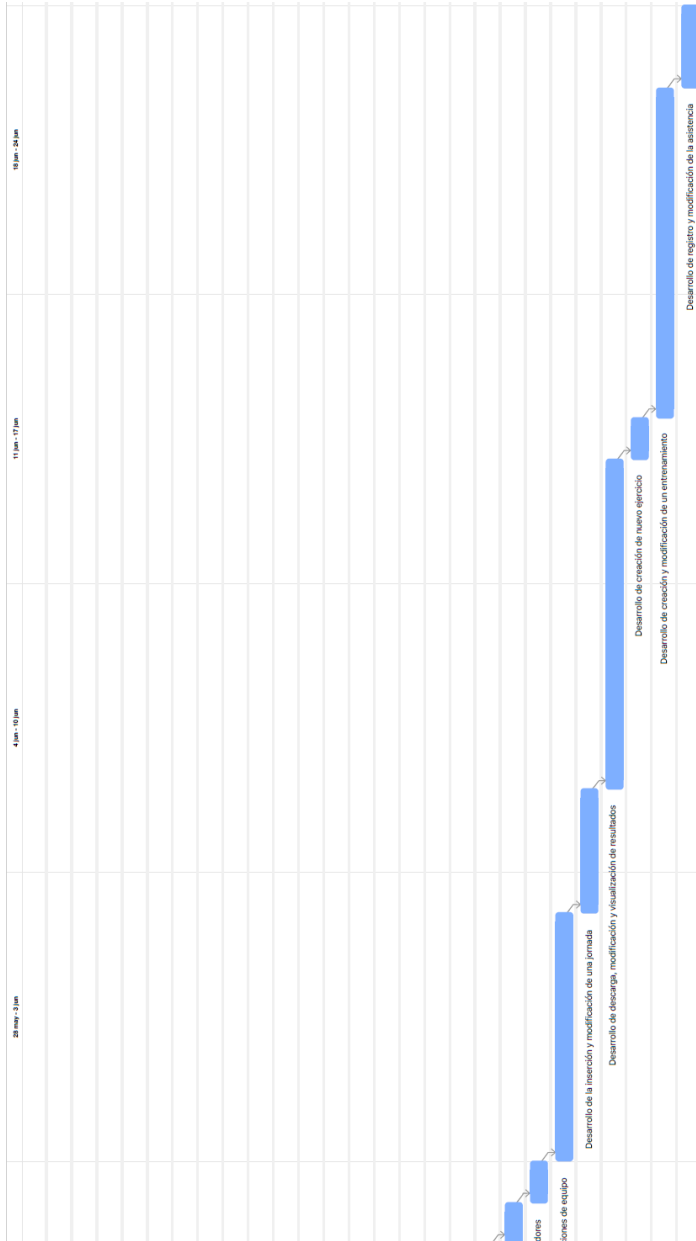


Figura 2.3: Diagrama de Gantt con la calendarización planificada.

2.6.6. Presupuesto

El proyecto está planificado para desarrollarse en torno a 300 horas, lo que equivaldría entorno al trabajo de 2 meses a jornada completa. Teniendo en cuenta que el salario medio nacional en el sector de la información según el informe de infoempleo de Adecco de 2016 [6]

es de 25536€ calculamos el siguiente presupuesto.

Concepto	Cantidad	Precio por unidad	Total
Trabajo humano en horas	300	13,3€	3990€
Astah Profesional	4	9,99€	39,96€
Total			4029,96€

Cuadro 2.1: Cálculo presupuesto del proyecto

Para el uso de astah profesional se ha decidido utilizar en este caso la suscripción mensual, de esta forma quedan explicadas las 4 unidades que aparecen en la tabla anterior.

2.6.7. Plan de gestión de riesgos

La gestión de los riesgos es un aspecto muy importante en el desarrollo de un proyecto y no tener en cuenta los riesgos puede suponer el fracaso del mismo.

A continuación se exponen tanto los riesgos detectados, con una descripción de los mismos, su impacto y probabilidad, como la matriz de exposición mostrada en la tabla 2.2 Con ambos datos se obtiene la exposición al riesgo.

Cuadro 2.2: Matriz Impacto/Probabilidad

Impacto\ Probabilidad	Muy Alto	Alto	Medio	Bajo	Muy Bajo
Catastrófico	Alto	Alto	Moderado	Moderado	Bajo
Crítico	Alto	Alto	Moderado	Bajo	Ninguno
Marginal	Moderado	Moderado	Bajo	Ninguno	Ninguno
Despreciable	Moderado	Bajo	Bajo	Ninguno	Ninguno

A continuación se expondrán los riesgos detectados, con su descripción, su impacto y su probabilidad y el plan de reducción y contingencia

R01 - Pérdida de datos

Pérdida de documentos o de código con la consecuente pérdida del tiempo invertido.

Impacto Catastrófico

Probabilidad 5%

Exposición Baja

Plan de Protección Utilizar control de versiones para mantener la evolución del código y de los documentos en un lugar seguro.

Plan de Contingencia En caso de pérdida de datos habrá que hacer una evaluación del impacto que esto generaría y hacer una replanificación en función de esto.

R02 - Enfermedad

Se produce una pausa en el progreso del proyecto por la imposibilidad de seguir trabajando debido a una enfermedad.

Impacto Marginal

Probabilidad 5 %

Exposición Ninguna

Plan de Protección N/A.

Plan de Contingencia Realizar una replanificación del proyecto según sea necesario en función del tiempo que se haya perdido por la enfermedad.

R03 - Cambios en requisitos

Durante la evolución del proyecto se ha encontrado un nuevo requisito que hay que incorporar a los ya descritos con anterioridad.

Impacto Crítico

Probabilidad 5 %

Exposición Baja

Plan de Protección Efectuar un análisis de requisitos exhaustivo.

Plan de Contingencia Cumplir con lo estipulado en el plan de control.

R04 - Fallo de Calendarización

La planificación temporal demuestra ser insuficiente para cumplir los plazos previstos.

Impacto Crítico

Probabilidad 20 %

Exposición Alta

Plan de Protección Tratar de ser realista en la planificación y cumplir con el trabajo previsto en cada momento.

Plan de Contingencia Cumplir con lo estipulado en el plan de control.

R05 - Diseño erróneo

El diseño que ha sido elaborado demuestra no ser el correcto.

Impacto Crítico

Probabilidad 10%

Exposición Moderada

Plan de Protección Tratar de ajustarse a lo obtenido en la fase de análisis y tener especial cuidado con cualquier duda que se presente en fase de diseño.

Plan de Contingencia Corregir el error y realizar una planificación si así es necesario.

R06 - Tiempo de aprendizaje de las tecnologías

El aprendizaje del funcionamiento de las tecnologías de desarrollo pueden acarrear un mayor tiempo que el esperado al realizar la planificación.

Impacto Crítico

Probabilidad 5%

Exposición Baja

Plan de Protección N/A

Plan de Contingencia Replanificar en función del retraso producido.

R07 - Fallos en implementación

El software desarrollado presenta bugs.

Impacto Crítico

Probabilidad 20%

Exposición Alta

Plan de Protección Utilizar buenas prácticas durante el desarrollo, manteniendo un código fácil de leer y ordenado hará que seamos menos propensos a los errores.

Plan de Contingencia Corregir los bugs y replanificar en caso necesario.

2.7. Planes de procesos de soporte

En esta sección se detallan los planes de soporte para el proyecto.

2.7.1. Plan de gestión de configuraciones

Se dispondrá de un sistema de control de versiones online para la documentación y el código del proyecto. Se utilizará un repositorio privado en Bitbucket que utiliza tecnología GIT.

Capítulo 3

Tecnologías utilizadas

En esta sección se detallarán las tecnologías utilizadas para el desarrollo del proyecto, justificando su uso.

3.1. Estado del arte

En la actualidad, no hay ninguna aplicación que permita llevar a cabo las tareas para las que se ha pensado este proyecto, aunque si que existen algunas aplicaciones que comparten alguna funcionalidad con la aplicación del proyecto. Estas principalmente lo que hacen es servir como biblioteca de ejercicios, estando organizados por características y muchas veces acompañadas por imágenes o vídeo para ilustrar el concepto. estas aplicaciones solo muestran esta funcionalidad sin permitir agrupar un par de ejercicios para realizar un entrenamiento. Otras aplicaciones lo que hacen es ofrecerte un entrenamiento ya predefinido, estas aplicaciones suelen estar más enfocadas a un entrenamiento individual del usuario de la aplicación.

Todas estas aplicaciones ofrecen alguna característica similar con el proyecto pero están mas enfocada a solo ofrecer información mientras que este proyecto esta más enfocado en ser un apoyo y complementar la actividad de entrenador.

Unos ejemplos de estas aplicaciones pueden ser:

- Entrenamiento de baloncesto (app para Android).
- Basketball training (app para Android).
- www.ejerciciosbaloncesto.com (web con ejercicios).

Las dos aplicaciones para móviles se eligieron como ejemplos ya que ambas eran las primeras en aparecer en Google Play. Las características principales de estas dos aplicaciones son que nos ofrecen entrenamientos ya predefinidos para mejorar como jugadores. Una de las grandes diferencias que hay con nuestro proyecto es que en estas apps los entrenamientos te los dan ya hechos con unos ejercicios específicos mientras que en nuestra aplicación somos nosotros quienes generamos los entrenamientos gracias a los ejercicios que hay en la base de datos. Esto nos permite generar una gran combinación de entrenamientos distintos mientras que en las aplicaciones mencionadas anteriormente el número de entrenamientos es limitado.

www.ejerciciosbaloncesto.com por otro lado es una página web que al igual que nuestra aplicación ofrece un gran número de ejercicios de baloncesto. La ventaja de nuestro proyecto

sobre esta web reside en poder generar nuestros propios entrenamientos a partir de los ejercicios que hay en la base de datos.

3.2. Tecnologías web por parte del cliente

3.2.1. Navegador web

Un navegador web es un tipo de software que permite la visualización de documentos y sitios en hipertexto, comúnmente agrupados bajo al denominación de Web o Internet.

Un navegador opera a través de Internet, interpretando archivos y sitios web que contienen información y contenido en hipertexto de todas partes del mundo. Estos documentos a parte de presentar información pueden contener referencias o hipervínculos a otros documentos. Los navegadores han ido evolucionando a lo largo del tiempo permitiendo no solo mostrar documentos estáticos, sino permitiendo además interactuar con la aplicación y obtener respuestas por parte del servidor. Gracias a estas nuevas opciones que ofrecen los navegadores juntado a que cada vez hay más dispositivos a través de los cuales se pueden utilizar navegadores web han generado una nueva forma de desarrollar aplicaciones. Aunque las pantallas donde se utilizan los navegadores pueden ser muy distintas dependiendo del dispositivos desde el que se está trabajando, gracias al diseño responsive una aplicación web se puede adaptar a cualquier pantalla.

Alguna de las ventajas que se obtiene del desarrollo de aplicaciones web son los siguientes:

- **Compatibilidad multiplataforma.** Las aplicaciones web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables.
- **Inmediatez de acceso.** Las aplicaciones basadas en web no necesitan ser descargadas, instaladas y configuradas.

A la hora de desarrollar aplicaciones web hay que tener en cuenta la gran cantidad de navegadores que existen en la actualidad para intentar hacer a nuestra aplicación compatible con el mayor número de navegadores posibles, aunque como podemos observar en la imagen 3.1 casi un 56 % del mercado esta abarcada por Google Chrome.

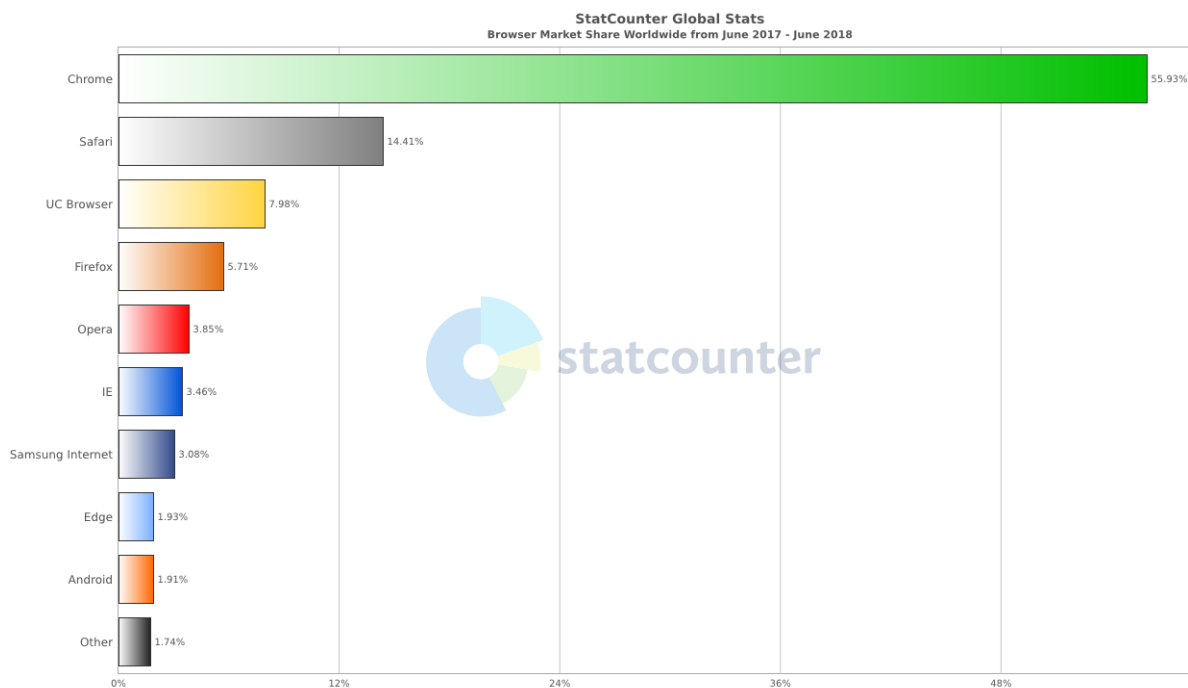


Figura 3.1: Porcentaje uso navegadores según StatCounter, Global Stats. Imagen tomada de su web [12]

3.2.2. HTML

HTML (HyperText Markup Language, Lenguaje de Marcado de Hipertexto) es el lenguaje en el que se construyen las páginas web. La primera descripción de HTML disponible públicamente fue un documento llamado HTML Tags (Etiquetas HTML), publicado por primera vez en Internet por Tim Berners-Lee, un científico del CERN, en 1991. Su propósito era definir un lenguaje que permitiese crear vínculos electrónicos entre distintos documentos, de tal modo que desde un documento se pudiese referir a otro y navegar fácilmente hasta él. [10]

HTML es un lenguaje basado en etiquetas. Estas al principio no eran muy numerosas pero para satisfacer las necesidades de los desarrolladores se han ido creando nuevas etiquetas aumentando la complejidad del lenguaje.

Para que no haya incompatibilidades por las nuevas etiquetas creadas la organización W3C creó un estándar con las características que tiene que tener el lenguaje HTML.

```

1 <body>
2   <h2>Navegadores mas usados</h2>
3   <ul>
4     <li>Google Chrome</li>
5     <li>Safari</li>
6     <li>UC Browser</li>
7     <li>Firefox</li>
8   </ul>
9 </body>

```

Figura 3.2: Ejemplo de código HTML

3.2.3. CSS

CSS (Cascading Style Sheets) es un lenguaje que permite modificar la presentación del contenido de un documento HTML. Esto podría llevarse a cabo también empleando etiquetas de formato que nos permitan especificar el tipo de fuente, color, tamaño, etc. del texto de distintas secciones del documento. [10]

Las hojas de estilo nos permiten especificar cómo se va a presentar el contenido de las distintas etiquetas HTML en un documento independiente del propio documento HTML. De este modo, podemos cambiar la presentación del documento HTML sin necesidad de modificar este documento. Una de las mayores ventajas de CSS es que con una sola hoja de estilo podemos mantener el estilo de toda la aplicación.

Una regla CSS consiste en un selector, que define el elemento o elementos que se van a ver afectados y una lista de declaraciones de estilo para ese selector, formado por una clave y un valor. La clave es el nombre de la propiedad que se quiere definir y el valor es lo que se le quiere asignar a esa propiedad.

El selector puede ser simplemente un elemento HTML, o tomar formas más complicadas como por ejemplo en que contexto está el elemento el cual queremos dar un diseño específico.

```

1 ul li {
2   font-size : 10pt;
3   background-color : 666666;
4 }

```

Figura 3.3: Ejemplo de código CSS

3.2.4. Materialize

Materialize [2] es un moderno framework de front-end responsive basado en Material Design utilizando CSS y JavaScript. “Material Design [5] es un lenguaje visual diseñado por Google que sintetiza los principios clásicos de un buen diseño con la innovación de la tecnología y la ciencia”. Al estar basado en Material Design de Google utiliza elementos atractivos, coherentes y funcionales para hacer la experiencia de usuario lo mas agradable posible. Como hemos explicado con anterioridad al ser responsive permite que la aplicación web se acople a los distintos tamaños de las pantallas de los dispositivos que pueden usarla.

Las principales características de Materialize son:

- Tiene muchos componentes y clases predefinidas, para hacer uso de ellas más fácilmente, aumentando el ritmo de desarrollo
- El framework de Materialize permite entregar mayor información al usuario al usar los principios y componentes de Material Design con sus componentes, transiciones y animaciones ya incorporados.
- Materialize CSS proporciona documentación profunda, y muchos ejemplos de código que le ayudarán a los usuarios nuevos a poder empezar con este framework.
- Integra Parallax, Pushpin y scrollspy para hacer un diseño más atractivo. Parallax es un efecto que sucede al hacer scroll. El fondo parece moverse a una velocidad distinta que el contenido dando sensación de profundidad. Pushpin es un efecto que sucede al hacer scroll. Al hacer scroll y llegar a cierta parte un elemento se queda anclado, generando por ejemplo la sensación de una cabecera. Scrollspy es un efecto que al igual que los anteriores se genera al hacer scroll. Este efecto nos permite mantener en pantalla elementos a los que queramos dar importancia aunque se haga scroll.

```
MyWebsite/  
|--css/  
| |--materialize.css  
|  
|--fonts/  
| |--roboto/  
|  
|--js/  
| |--materialize.js  
|  
|--index.html
```

Figura 3.4: Estructura básica Materialize

3.2.5. JavaScript

JavaScript es un lenguaje de scripting que se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM). Tradicionalmente se utilizaba en páginas web HTML para realizar operaciones solamente en el lado del cliente, sin acceso a funciones del servidor. Actualmente también es utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX (Asynchronous JavaScript And XML), que es una técnica de desarrollo web para crear aplicaciones interactivas. En este proyecto, se usa JavaScript junto el framework de JQuery y Materialize para dar dinamismo a la interfaz y modificar el aspecto visual de la aplicación. También se utilizará la tecnología mencionada anteriormente AJAX para realizar todas las peticiones asíncronas que sean necesarias..

```

1 <script type="text/javascript">
2     function getAction()
3     {
4         var x=document.forms.myForm;
5         alert (x. action );
6     }
7
8     function changeAction(action)
9     {
10        var x=document.forms.myForm;
11        x. action=action;
12        alert (x. action );
13    }
14 </script>

```

Figura 3.5: Ejemplo de código JavaScript

3.2.6. JQuery

JQuery [11] es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. DOM es esencialmente una interfaz de plataforma que proporciona un conjunto estándar de objetos para representar documentos HTML, XHTML y XML. Sus principales características son:

- La selección de elementos DOM, como ya hemos mencionado.
- Uso de AJAX.
- Manipulación de CSS.
- Permite crear animaciones.
- Permite capturar eventos.

```

1 <script>
2 $(" #c1").click( function () {
3     $(" #cont1").hide(1500);
4     });
5 </script>

```

Figura 3.6: Ejemplo de código JQuery

3.3. Tecnologías web por parte del servidor

3.3.1. Java

Java es un lenguaje de programación basado en clases , orientado a objetos y específicamente diseñado para tener la menor cantidad de dependencias de implementación posible. Su objetivo es permitir a los desarrolladores de aplicaciones “escribir una vez, ejecutar en cualquier lugar”. Esto significa que el código compilado de Java se puede ejecutar en todas las plataformas que admiten Java sin la necesidad de recompilarlo. Las aplicaciones Java generalmente se compilan en bytecode que se puede ejecutar en cualquier máquina virtual Java (JVM) independientemente de la arquitectura de la computadora. A partir de 2016, Java es uno de los lenguajes de programación más populares en uso especialmente para aplicaciones web cliente-servidor, con un informe de 9 millones de desarrolladores. Java fue desarrollado originalmente por James Gosling en Sun Microsystems (que luego fue adquirido por Oracle Corporation) y lanzado en 1995 como un componente central de la plataforma Java de Sun Microsystems . El lenguaje deriva gran parte de su sintaxis de C y C ++ , pero tiene menos facilidades de bajo nivel que cualquiera de ellos.

Los principios de Java son:

1. Debe ser “simple, orientado a objetos y familiar”.
2. Debe ser “robusto y seguro”.
3. Debe ser “arquitectura neutral y portátil”.
4. Debe ejecutar con “alto rendimiento”.
5. Debe ser “interpretado, enhebrado y dinámico”.

3.3.2. Apache Derby

Apache Derby (anteriormente distribuido como IBM Cloudscape) es un sistema de administración de bases de datos relacionales (RDBMS) desarrollado por Apache Software Foundation que puede integrarse en programas Java y utilizarse para el procesamiento de transacciones en línea. Apache Derby se desarrolla como un proyecto de código abierto bajo la licencia Apache 2.0 . Oracle distribuye los mismos binarios bajo el nombre Java DB.

El núcleo de la tecnología, el motor de base de datos de Derby, es un motor de base de datos embebido y relacional con todas las funciones, que admite JDBC y SQL como API de programación. Utiliza la sintaxis SQL de IBM DB2.

Capítulo 4

Requisitos y Análisis

4.1. Introducción

Como ya se comentó anteriormente, el objetivo del proyecto es desarrollar una aplicación que nos permita poder administrar un club de baloncesto y ayudar a los entrenadores en su labor diaria con todo lo relacionado con los entrenamientos. Para poder llevar a cabo esto hay que tener muy claro los requisitos, funcionalidades y servicios que va a tener la aplicación, así como sus restricciones. A continuación se estudiarán a fondo estos aspectos citados.

4.2. Entorno

Gracias a todos los años que llevo tanto como de jugador como de entrenador entiendo casi a la perfección cómo funciona un club y las necesidades básicas que puede tener un entrenador a la hora de preparar un entrenamiento.

4.2.1. Conceptos previos

A continuación se expondrán conceptos básicos que ayuden a entender mejor el desarrollo de la aplicación.

- **Club.** Un club está formado por todos los equipos, con sus respectivos jugadores y todos los entrenadores que se encargan de cada uno de los equipos.
- **Equipo.** Para poder competir, un equipo tiene que estar formado por un mínimo de 8 jugadores y un máximo de 15. Un equipo puede tener dos entrenadores, pero en ocasiones un equipo puede tener un tercer entrenador. Para que este tercer entrenador pueda estar en el banquillo durante los partidos se le suele sacar la ficha de delegado del equipo.
- **Jugador.** En cuanto a los datos que pide la Fundación Municipal de Deportes de Valladolid [1] de los jugadores son nombre, apellidos, dirección de residencia y año de nacimiento.
- **Jornada.** Una jornada está formada por cada uno de los partidos que jueguen los equipos del club. De normal cada fin de semana todos los equipos juegan un partido.

- **Entrenamiento.** Un entrenamiento está formado por un conjunto de ejercicios. Cada ejercicio se trabaja durante un tiempo determinado. Cada ejercicio trabaja un concepto principalmente, pero muchas veces un mismo ejercicio trabaja más de un concepto. No todos los ejercicios se pueden utilizar en todas las categorías ya que pueden necesitarse conceptos avanzados para los jugadores.
- **Asistencia.** Aunque lo normal es que la asistencia de un jugador a un entrenamiento o a un partido se tendría que reducir a *ha asistido* o *no ha asistido* muchas veces los entrenadores se encuentran con que el jugador por distintos motivos puede *llegar tarde* o tener que *irse pronto* y esto hay que tenerlo en cuenta.

4.3. Funcionalidad requerida para la aplicación

Los requisitos que se desarrollarán a continuación nos ayudará a entender las necesidades exactas que tienen los usuarios.

4.3.1. Requisitos funcionales

Los requisitos funcionales (RF) describen el funcionamiento del sistema y como debe comportarse ante cada situación.

- **RF-01** El Sistema permitirá al usuario identificarse.
- **RF-02** El sistema permitirá al usuario elegir el club y/o el equipo con el que trabajar.
- **RF-03** El sistema permitirá al usuario de tipo administrador crear un equipo para su club.
- **RF-04** El sistema permitirá al usuario de tipo entrenador sin club crear sus propios equipos.
- **RF-05** El sistema permitirá al usuario de tipo administrador y entrenador sin club crear jugadores para un equipo.
- **RF-06** El sistema permitirá al usuario de tipo administrador y entrenador sin club modificar un equipo.
- **RF-07** El sistema permitirá al usuario de tipo administrador añadir un entrenador a su club.
- **RF-08** El sistema permitirá al usuario de tipo administrador eliminar un entrenador de su club.
- **RF-09** El sistema permitirá al usuario de tipo administrador crear una jornada con los partidos correspondientes.
- **RF-10** El sistema permitirá al usuario de tipo administrador modificar una jornada y los partido correspondientes.
- **RF-11** El sistema permitirá al usuario de tipo administrador descargar los resultados de una jornada.

- **RF-12** El sistema permitirá al usuario de tipo entrenador crear un nuevo entrenamiento.
- **RF-13** El sistema permitirá al usuario de tipo entrenador modificar un entrenamiento.
- **RF-14** El sistema permitirá al usuario de tipo entrenador añadir un comentario para un ejercicio de un entrenamiento.
- **RF-15** El sistema permitirá al usuario de tipo entrenador apuntar la asistencia de cada jugador para una fecha.
- **RF-16** El sistema permitirá al usuario de tipo entrenador modificar la asistencia de cada jugador para una fecha.
- **RF-17** El sistema permitirá al usuario de tipo entrenador añadir un nuevo ejercicio..
- **RF-18** El sistema permitirá al usuario de tipo entrenador con club modificar el resultado de un partido de una jornada.
- **RF-19** El sistema permitirá al usuario de tipo entrenador ver los resultados de los partidos de una jornada de su club.
- **RF-20** El sistema permitirá al usuario de tipo entrenador descargar la asistencia de sus jugadores a un archivo.

4.3.2. Requisitos no funcionales

Un requisito no funcional (RNF), es un requisito que especifica como debe funcionar el Sistema. Muchas veces estos requisitos son más importantes que los propios requisitos funcionales. Existen tres tipos de requisitos no funcionales. Los requisitos del producto, que especifican el comportamiento del Sistema. Los requisitos de organización que especifican políticas y procedimientos de la empresa que utilizará el Sistema. Los requisitos externos que son los que ofrecen factores externos al Sistema.

A continuación expondremos los requisitos no funcionales de este Sistema.

- **RNF-01** La identificación se hará de forma segura.
- **RNF-02** La descargar de datos se realizará mediante un archivo .xls.

4.3.3. Requisitos de información

A continuación se explicarán los requisitos de información recopilados.

- **RNF-01** El Sistema deberá almacenar la información de cada usuario:
 - Correo.
 - Contraseña.
 - Nombre.
 - Apellidos.
 - Tipo Usuario.
- **RNF-02** El Sistema deberá almacenar la información de cada equipo:

- Nombre.
 - Categoría.
 - Sexo.
 - Entrenador1.
 - Entrenador2.
 - Delegado.
- **RNF-03** El Sistema deberá almacenar la información de los jugadores:
- Nombre.
 - Apellidos.
 - Dirección.
 - Fecha de nacimiento.
- **RNF-04** El Sistema deberá almacenar la información de los partidos:
- Equipo A.
 - Equipo B.
 - Resultado equipo A.
 - Resultado equipo B.
 - Fecha.
 - Hora.
 - Lugar.
- **RNF-05** El Sistema deberá almacenar la información de los ejercicios:
- Nombre
 - Descripción
 - Categoría
 - Concepto1
 - Concepto2
 - Concepto3
 - Tipo
- **RNF-06** El Sistema deberá almacenar la información de los entrenamientos:
- Ejercicios
 - Comentario
 - Fecha

4.4. Roles

Como se ha indicado en los requisitos funcionales hay tres tipos de usuarios, el administrador, el entrenador que pertenece a un club que usa la aplicación y el entrenador que pertenece a un club que no utiliza la aplicación. Un mismo entrenador puede pertenecer a más de un club y a más de un equipo. El administrador tendrá acceso a las funciones básicas para poder administrar al club como puede ser crear equipos, asignar entrenadores, añadir las jornadas... El entrenador que pertenece a un club podrá realizar las acciones propias de la administración de un equipo tales como crear un entrenamiento, poner la asistencia a los jugadores, subir los resultados de las jornadas, etc. El entrenador sin club podrá realizar las mismas acciones que el entrenador asociado a un club pero con pequeñas diferencias como que podrá crear equipos pero no podrá subir resultados.

4.5. Casos de uso

Un caso de uso es una secuencia de interacciones que se desarrollarán entre un Sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio Sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. Los diagramas de casos de uso se utilizan para ilustrar los requisitos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

A continuación se mostrará el diagrama de casos de uso para este proyecto.

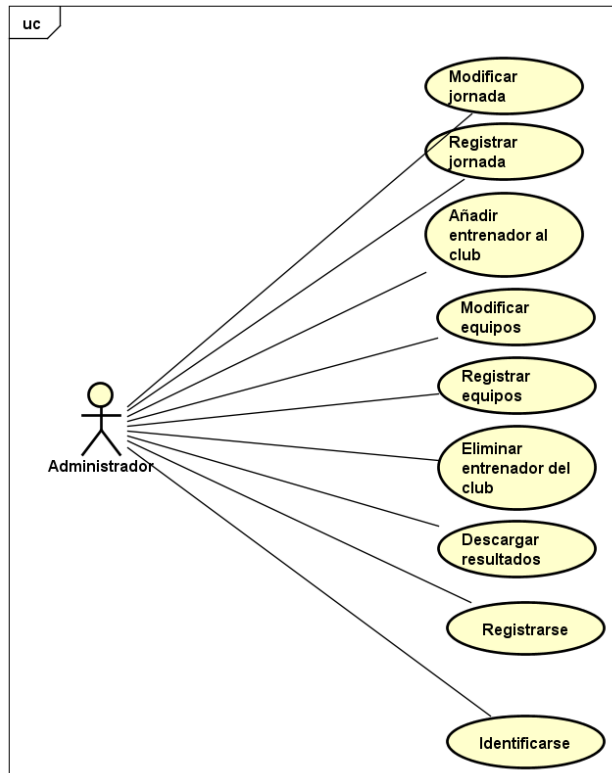


Figura 4.1: Diagrama de Casos de Uso del administrador

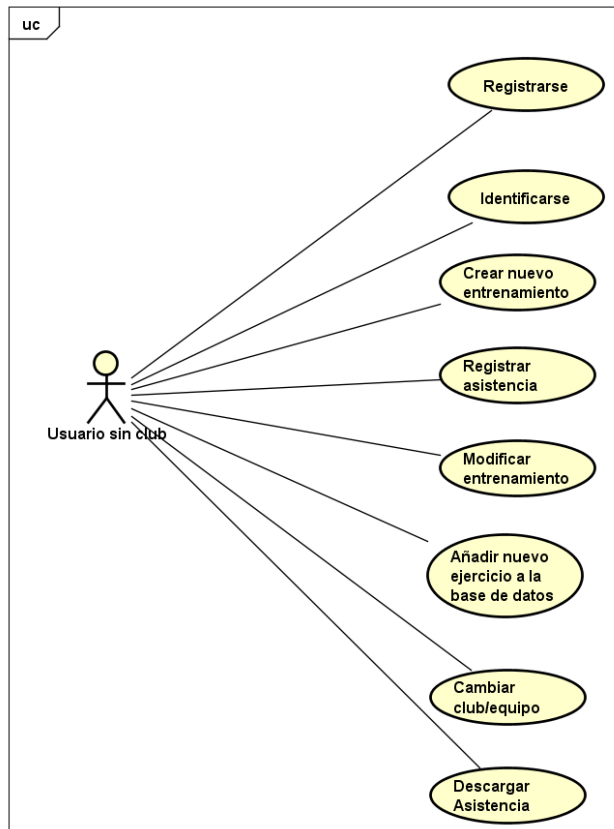


Figura 4.2: Diagrama de casos de uso de un entrenador sin club

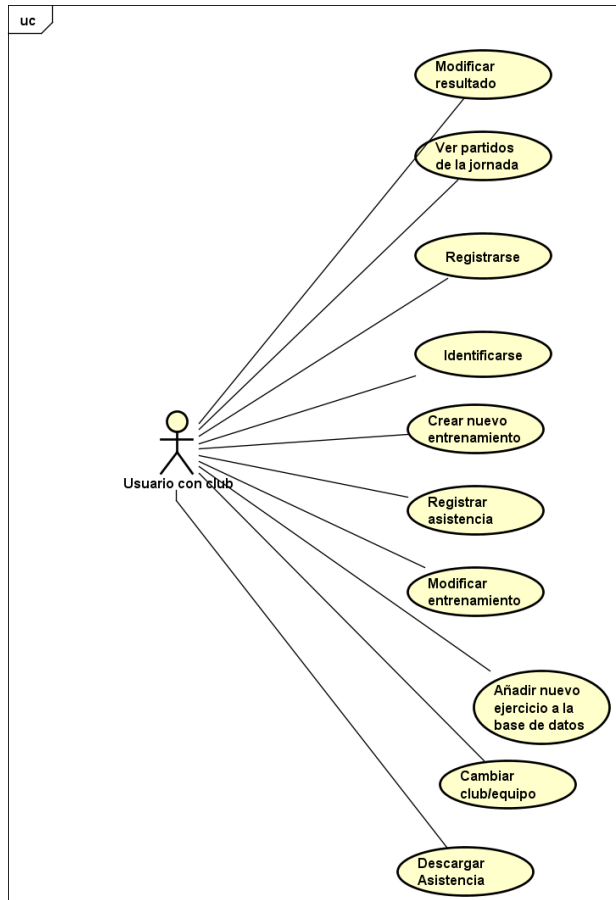


Figura 4.3: Diagrama de casos de uso de un entrenador con club

Cada caso de uso normalmente está formado por un escenario que denominamos como secuencia normal, en la que se termina con éxito y uno o varios escenarios donde no se termina con éxito.

A continuación se hará una descripción de cada uno de los casos de uso.

CU-01	Registrarse	
Descripción	El sistema deberá permitir al actor registrarse.	
Precondición		
Secuencia normal	Paso	Acción
	1	El actor selecciona registrarse.
	2	El Sistema muestra los datos a rellenar Correo, Contraseña, Repetir contraseña, Nombre, Apellidos y Tipo de usuario.

	3	El actor introduce los datos solicitados anteriormente.
	4	El Sistema comprueba los datos introducidos, si ha introducido el tipo de usuario administrador se le pide que introduzca el nombre del club, si ha introducido tipo de usuario entrenador el sistema registra el usuario finaliza el caso de uso.
	5	El actor introduce el nombre del club.
	6	El Sistema comprueba los datos introducidos y registra el usuario y el club.
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.
	5a	El Sistema comprueba que existe un club registrado con ese nombre y el caso de uso continua en el paso 2.

Cuadro 4.1: Caso de uso : Registrarse

CU-02	Ingresar.	
Descripción	El sistema deberá permitir al actor ingresar.	
Precondición	El actor deberá de estar registrado en el Sistema	
Secuencia normal	Paso	Acción
	1	El actor selecciona ingresar.
	2	El Sistema muestra los datos a rellenar Correo y Contraseña.
	3	El actor introduce los datos solicitados anteriormente.
	4	El Sistema comprueba los datos introducidos, si el actor es de tipo administrador muestra la pantalla principal, si es entrenador el Sistema pide que seleccione el club y/o equipo con el que trabajar.
	5	El actor introduce el club y/o equipo con el que trabajar.
	6	El Sistema comprueba los datos introducidos e ingresa al usuario.
Excepciones	Paso	Acción

	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.
	4b	Si el correo o la contraseña introducidas no son correctas se muestra un mensaje y el caso de uso continua en el paso 2.
	6a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 4.

Cuadro 4.2: Caso de uso : Ingresar

CU-03	Registrar equipos	
Descripción	El sistema deberá permitir al actor Administrador o Entrenador sin club registrar un nuevo equipo.	
Precondición	El actor deberá de haber ingresado en el Sistema	
Secuencia normal	Paso	Acción
	1	El actor selecciona añadir un equipo.
	2	El Sistema muestra los datos a rellenar Nombre, Categoría, Sexo, Entrenador1, Entrenador2, Delegado.
	3	El actor introduce los datos solicitados anteriormente.
	4	El Sistema comprueba los datos introducidos y pide que rellene los datos de los jugadores que componen el equipo Nombre, Apellidos, Dirección Fecha de nacimiento.
	5	El actor introduce los datos solicitados.
	6	El Sistema comprueba los datos introducidos y crea el equipo y los jugadores.
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.

	4b	Si se pone el mismo entrenador como entrenador1 y entrenador2 o delegado el sistema muestra un mensaje de error y el caso de uso continua en el paso 2.
	6a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 4.
	6b	El actor cancela la operación y el caso de uso queda sin efecto.

Cuadro 4.3: Caso de uso : Registrar equipo

CU-04	Modificar equipos	
Descripción	El sistema deberá permitir al actor Administrador o Entrenador sin club modificar un equipo.	
Precondición	El actor deberá de haber ingresado en el Sistema y deberá de haber un equipo registrado.	
Secuencia normal	Paso	Acción
	1	El actor selecciona modificar un equipo.
	2	El Sistema muestra los datos del equipo seleccionado.
	3	El actor modifica los datos que desea cambiar.
	4	El Sistema comprueba los datos introducidos y muestra los datos de los jugadores para modificarlos
	5	El actor modifica los datos deseados.
	6	El Sistema comprueba los datos introducidos y modifica el equipo y los jugadores.
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.
	4b	Si se pone el mismo entrenador como entrenador1 y entrenador2 o delegado el sistema muestra un mensaje de error y el caso de uso continua en el paso 2.
	6a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 4.

	6b	El actor cancela la operación y el caso de uso queda sin efecto.
--	----	------------------------------------------------------------------

Cuadro 4.4: Caso de uso : Modificar equipo

CU-05	Añadir entrenador al club	
Descripción	El sistema deberá permitir al actor Administrador añadir un entrenador al club.	
Precondición	El actor deberá de haber ingresado en el Sistema y deberá de haber un entrenador registrado en el Sistema.	
Secuencia normal	Paso	Acción
	1	El actor selecciona añadir un entrenador.
	2	El Sistema muestra los datos a introducir del entrenador Correo.
	3	El actor introduce los datos pedidos por el Sistema.
	4	El Sistema comprueba los datos introducidos y añade al entrenador al club
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.
	4b	Si se pone el correo de un entrenador el sistema muestra un mensaje de error y el caso de uso continua en el paso 2.

Cuadro 4.5: Caso de uso : Añadir entrenador

CU-06	Eliminar entrenador del club	
Descripción	El sistema deberá permitir al actor Administrador eliminar un entrenador del club.	
Precondición	El actor deberá de haber ingresado en el Sistema y deberá de haber un entrenador registrado como entrenador de ese club en el Sistema.	
Secuencia normal	Paso	Acción
	1	El actor selecciona eliminar un entrenador.
	2	El Sistema muestra los entrenadores de ese club

	3	El actor elige el entrenador a eliminar.
	4	El Sistema comprueba los datos introducidos y elimina al entrenador del club
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.

Cuadro 4.6: Caso de uso : Eliminar entrenador

CU-07	Registrar jornada	
Descripción	El sistema deberá permitir al actor Administrador añadir una jornada.	
Precondición	El actor deberá de haber ingresado en el Sistema y deberá de haber un equipo de ese club registrado en el Sistema.	
Secuencia normal	Paso	Acción
	1	El actor selecciona registrar una jornada.
	2	El Sistema muestra los datos a introducir de la jornada Primer día de la jornada, Equipo1, Resultado equipo1, Equipo2, Resultado equipo2, Fecha partido, Hora partido, Lugar partido(por cada partido de la jornada).
	3	El actor introduce los datos pedidos por el Sistema.
	4	El Sistema comprueba los datos introducidos y crea la jornada
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.

Cuadro 4.7: Caso de uso : Registrar jornada

CU-08	Modificar jornada
--------------	-------------------

Descripción	El sistema deberá permitir al actor Administrador modificar una jornada.	
Precondición	El actor deberá de haber ingresado en el Sistema y deberá de haber una jornada de ese club registrada en el Sistema.	
Secuencia normal	Paso	Acción
	1	El actor selecciona registrar una jornada.
	2	El Sistema muestra las fechas de las jornadas registradas.
	3	El actor selecciona la fecha de la jornada que desea modificar.
	4	El Sistema comprueba los datos introducidos y muestra los datos de la jornada seleccionada.
	5	El El actor modifica los datos que desea de la jornada.
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.
	5a	El actor cancela la operación y el caso de uso queda sin efecto.
	6a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 4.

Cuadro 4.8: Caso de uso : Modificar jornada

CU-09	Descargar resultados	
Descripción	El sistema deberá permitir al actor Administrador descargar los resultados de una jornada.	
Precondición	El actor deberá de haber ingresado en el Sistema y deberá de haber una jornada de ese club registrada en el Sistema.	
Secuencia normal	Paso	Acción
	1	El actor selecciona ver resultados de una jornada.

	2	El Sistema muestra las fechas de las jornadas registradas.
	3	El actor selecciona la fecha de la jornada que desea ver.
	4	El Sistema comprueba los datos introducidos y muestra los datos de la jornada seleccionada.
	5	El El actor selecciona la opción de descargar resultados.
	6	El Sistema descarga los resultados de la jornada.
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.
	5a	El actor cancela la operación y el caso de uso queda sin efecto.

Cuadro 4.9: Caso de uso : Descargar resultados

CU-10	Crear un entrenamiento	
Descripción	El sistema deberá permitir al actor Entrenador sin club y Entrenador con club crear un nuevo entrenamiento.	
Precondición	El actor deberá de haber ingresado en el Sistema.	
Secuencia normal	Paso	Acción
	1	El actor selecciona crear un entrenamiento.
	2	El Sistema muestra un selector de fechas para el entrenamiento.
	3	El actor selecciona la fecha para el entrenamiento.
	4	El Sistema comprueba los datos introducidos y muestra los datos a introducir para el ejercicio.
	5	El El actor introduce los datos del ejercicio.
	6	El Sistema comprueba los datos introducidos y muestra los ejercicios con esas características.
	7	El actor selecciona uno de los ejercicios mostrados.
8	El Sistema muestra la ficha del ejercicio seleccionado.	

	9	Si el usuario decide añadirlo a lista de ejercicios del entrenamiento introduce el tiempo que se va a trabajar el ejercicio y el caso de uso continua en el paso 10, si no, el caso de uso continua en el paso 4.
	10	El Sistema comprueba los datos introducidos y muestra la ejercicios del entrenamiento.
	11	Si el usuario decide finalizar el entrenamiento el caso de uso finaliza, si no, el caso de uso continua en el paso 4.
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.
	5a	El actor cancela la operación y el caso de uso queda sin efecto.
	7a	El actor cancela la operación y el caso de uso queda sin efecto.
	9a	El actor cancela la operación y el caso de uso queda sin efecto.
	9b	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 8.

Cuadro 4.10: Caso de uso : Crear entrenamiento

CU-11	Modificar un entrenamiento	
Descripción	El sistema deberá permitir al actor Entrenador sin club y Entrenador con club modificar un entrenamiento.	
Precondición	El actor deberá de haber ingresado en el Sistema y debe de tener creado al menos un entrenamiento.	
Secuencia normal	Paso	Acción
	1	El actor selecciona ver entrenamientos.
	2	El Sistema muestra las fechas de los entrenamientos creados.
	3	El actor selecciona la fecha del entrenamiento a modificar.
	4	El Sistema comprueba los datos introducidos y muestra la lista de todos los ejercicios que componen el entrenamiento.

	5	Si el actor selecciona un ejercicio a modificar el caso de uso continua en el paso 6, si no, se termina el caso de uso.
	6	El Sistema comprueba los datos introducidos y muestra la ficha del ejercicio seleccionado.
	7	El actor modifica el ejercicio.
	8	El Sistema comprueba los datos introducidos y el caso de uso continua en el paso 4.
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.
	5a	El actor cancela la operación y el caso de uso queda sin efecto.
	7a	El actor cancela la operación y el caso de uso queda sin efecto.

Cuadro 4.11: Caso de uso :Modificar un entrenamiento

CU-12	Añadir un nuevo ejercicio a la base de datos	
Descripción	El sistema deberá permitir al actor Entrenador sin club y Entrenador con club añadir un nuevo ejercicio a la base de datos.	
Precondición	El actor deberá de haber ingresado en el Sistema.	
Secuencia normal	Paso	Acción
	1	El actor selecciona añadir ejercicio.
	2	El Sistema muestra los datos a introducir del ejercicio Nombre, Descripción, Categoría, Concepto1, Concepto2, Concepto3, Tipo de ejercicio.
	3	El actor introduce los datos necesarios.
	4	El Sistema comprueba los datos introducidos y añade el ejercicio a la base de datos.
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.

Cuadro 4.12: Caso de uso :Añadir nuevo ejercicio a la base de datos

CU-13	Registrar Asistencia	
Descripción	El sistema deberá permitir al actor Entrenador sin club y Entrenador con club registrar la asistencia de los jugadores.	
Precondición	El actor deberá de haber ingresado en el Sistema y debe de ser entrenador de un equipo.	
Secuencia normal	Paso	Acción
	1	El actor selecciona añadir asistencia.
	2	El Sistema muestra a todos los jugadores del equipo y un selector para cada uno con una de las opciones de asistencia.
	3	El actor selecciona para cada jugador la asistencia.
	4	El Sistema comprueba los datos introducidos y añade la asistencia.
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.

Cuadro 4.13: Caso de uso :Registrar asistencia

CU-14	Modificar asistencia	
Descripción	El sistema deberá permitir al actor Entrenador sin club y Entrenador con club modificar una asistencia.	
Precondición	El actor deberá de haber ingresado en el Sistema y debe de tener guardada una asistencia.	
Secuencia normal	Paso	Acción
	1	El actor selecciona modificar asistencia.
	2	El Sistema muestra las fechas de las asistencias guardadas con anterioridad.
	3	El actor selecciona la fecha de la asistencia a modificar.

	4	El Sistema comprueba los datos introducidos y muestra a todos los jugadores del equipo y un selector para cada uno con una de las opciones de asistencia.
	5	El actor modifica la asistencia.
	6	El Sistema comprueba los datos introducidos y modifica la asistencia.
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.
	5a	El actor cancela la operación y el caso de uso queda sin efecto.
	6a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 4.

Cuadro 4.14: Caso de uso :Modificar asistencia

CU-15	Modificar resultado	
Descripción	El sistema deberá permitir al actor Entrenador sin club y Entrenador con club modificar el resultado de una jornada.	
Precondición	El actor deberá de haber ingresado en el Sistema y el administrador debe haber creado una jornada.	
Secuencia normal	Paso	Acción
	1	El actor selecciona modificar resultado.
	2	El Sistema muestra las fechas de las jornadas existentes de ese club.
	3	El actor selecciona la fecha de la jornada a modificar.
	4	El Sistema comprueba los datos introducidos y muestra los partidos de esa jornada.
	5	El actor modifica el resultado de su partido.
	6	El Sistema comprueba los datos introducidos y modifica la jornada.
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.

	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.
	5a	El actor cancela la operación y el caso de uso queda sin efecto.
	6a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 4.

Cuadro 4.15: Caso de uso :Modificar un resultado

CU-16	Ver resultados	
Descripción	El sistema deberá permitir al actor Entrenador sin club y Entrenador con club ver los resultados de los partidos de una jornada.	
Precondición	El actor deberá de haber ingresado en el Sistema y el administrador debe haber creado una jornada.	
Secuencia normal	Paso	Acción
	1	El actor selecciona ver resultados.
	2	El Sistema muestra las fechas de las jornadas existentes de ese club.
	3	El actor selecciona la fecha de la jornada a ver.
	4	El Sistema comprueba los datos introducidos y muestra los partidos de esa jornada.
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.

Cuadro 4.16: Caso de uso :Ver resultado

CU-17	Cambiar club/equipo	
Descripción	El sistema deberá permitir al actor Entrenador sin club y Entrenador con club cambiar de club y/o equipo con el que trabajar.	
Precondición	El actor deberá de haber ingresado en el Sistema.	
	Paso	Acción

Secuencia normal

	1	El actor selecciona cambiar club/equipo.
	2	El Sistema muestra los clubes y los equipos a los que entrena el actor.
	3	El actor selecciona el club y/o equipo con el que quiere trabajar.
	4	El Sistema comprueba los datos introducidos y cambia el club y/o equipo con el que trabaja el actor.
Excepciones	Paso	Acción
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.

Cuadro 4.17: Caso de uso :Cambiar club/equipo

CU-18	Descargar asistencia	
Descripción	El sistema deberá permitir al actor Entrenador, tanto con club como sin él, descargar los datos de la asistencia de sus jugadores para una fecha.	
Precondición	El actor deberá de haber ingresado en el Sistema y deberá de haber una asistencia registrada.	
Secuencia normal	Paso	Acción
	1	El actor selecciona ver asistencia.
	2	El Sistema muestra las fechas de las asistencias registradas.
	3	El actor selecciona la fecha de la asistencia que desea ver.
	4	El Sistema comprueba los datos introducidos y muestra los datos de la asistencia seleccionada.
	5	El El actor selecciona la opción de descargar asistencia.
	6	El Sistema descarga los datos de la asistencia.
Excepciones	Paso	Acción
	3a	El actor cancela la operación y el caso de uso queda sin efecto.
	4a	Si alguno de los campos requeridos no es introducido por el actor, el Sistema muestra un mensaje de error y el caso de uso continua en el paso 2.
	5a	El actor cancela la operación y el caso de uso queda sin efecto.

Cuadro 4.18: Caso de uso : Descargar asistencia

4.6. Matriz de trazabilidad de requisitos/Casos de uso

RF/CU	CU1	CU2	CU3	CU4	CU5	CU6	CU7	CU8	CU9	CU10	CU11	CU12	CU13	CU14	CU15	CU16	CU17	CU18
RF1	X	X																
RF2																	X	
RF3			X															
RF4			X															
RF5			X															
RF6				X														
RF7					X													
RF8						X												
RF9							X											
RF10								X										
RF11									X									
RF12										X								
RF13											X							
RF14											X							
RF15												X						
RF16													X					
RF17											X							
RF18														X				
RF19															X			
RF20																		X

Figura 4.4: Matriz trazabilidad

4.7. Modelo de dominio

En este apartado mostraremos el diagrama de modelo de dominio de la aplicación a desarrollar.

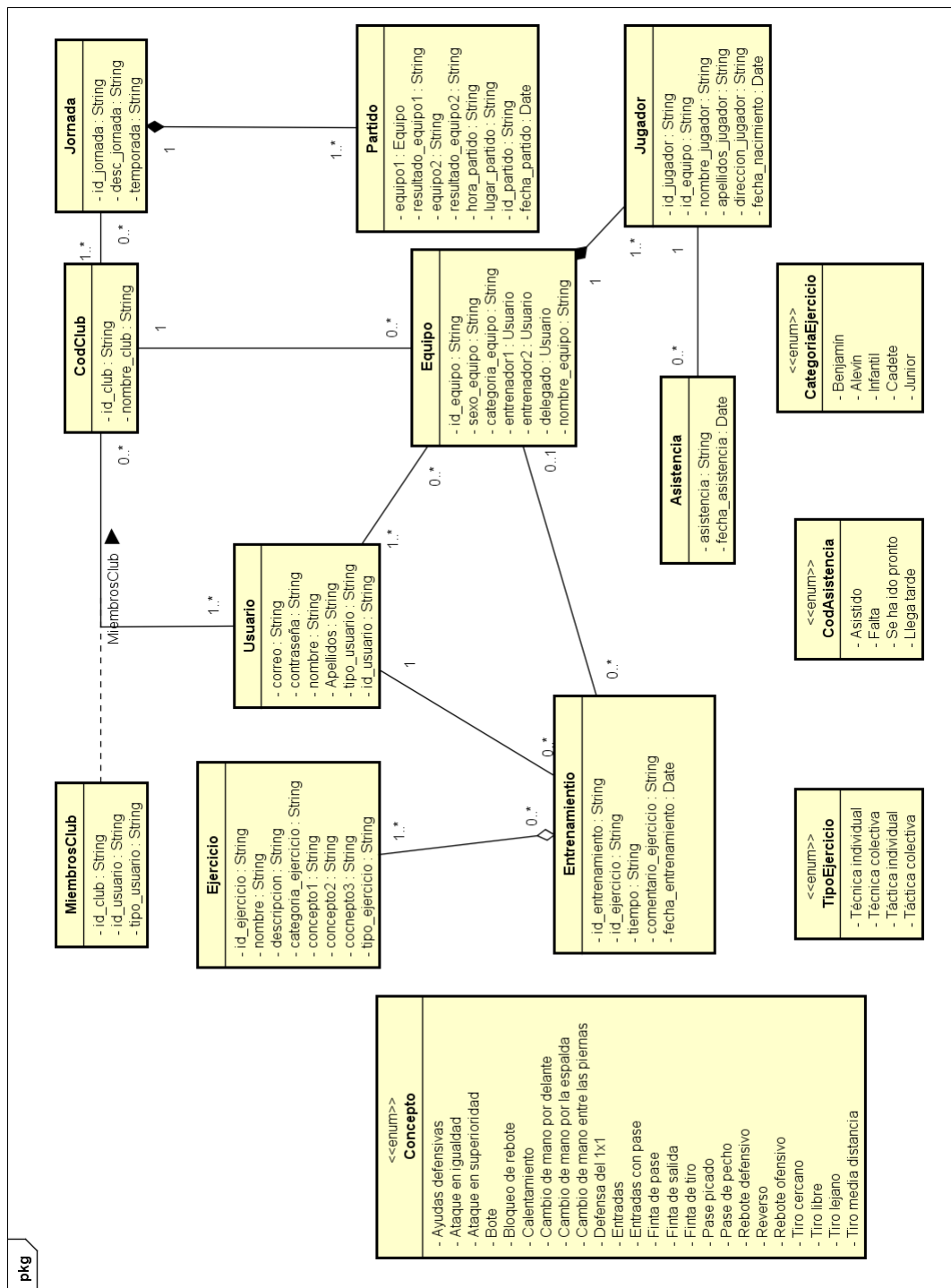


Figura 4.5: Modelo de dominio

Basándonos en los requisitos funcionales, caso de uso y en nuestros conocimientos sobre el mundo de los clubs de baloncesto hemos obtenido el modelo de dominio mostrado en la figura 4.5 y ahora explicaremos cada una de las clases que en el aparecen.

- **Usuario.** Representa al usuario de la aplicación. Como ya hemos mencionado anteriormente hay tres tipos de usuarios, los administradores, los entrenadores que tiene un club asignado y los entrenadores que no tienen ningún club asignado.
- **CodClub.** Representa a cada uno de los clubes que utilizan esta aplicación.
- **MiembroClub.** Representa la asociación entre un usuario y el/los clubs a los que pertenece.
- **Jornada.** Representa a las jornadas que hay a lo largo del año. Esta compuesta por partidos.
- **Partido.** Representa a cada uno de los partidos que se celebran en cada jornada.
- **Equipo.** Representa a los equipos que pertenecen a un club. Un equipo esta compuesto por jugadores.
- **Jugador.** Representa a los jugadores que componen cada uno de los equipos.
- **Asistencia.** Representa a la asistencia de cada uno de los jugadores a un entrenamiento o a cualquier fecha que el entrenador quiera tener en cuenta.
- **CodAsistencia.** Representa al código de valor que puede tomar cada asistencia.
- **Entrenamiento.** Representa a los entrenamientos que puede generar los usuarios de tipo entrenador. Están compuesta por un conjunto de ejercicios.
- **Ejercicio.** Representa a los ejercicios que están en la base de datos y que se pueden utilizar para componer un entrenamiento.
- **CodCategoriaEjercicio** Representa el código de valor que puede tener la categoría de cada ejercicio.
- **CodConcepto** Representa el código de valor que pueden tener los conceptos de cada ejercicio.
- **CodTipoEjercicio** Representa el código de valor que puede tener el tipo de cada ejercicio.

4.8. Máquinas de estado

A continuación se mostrará la máquina de estado de la clase usuario que es la única clase que puede tener algo de complejidad a la hora de entender sus estados.

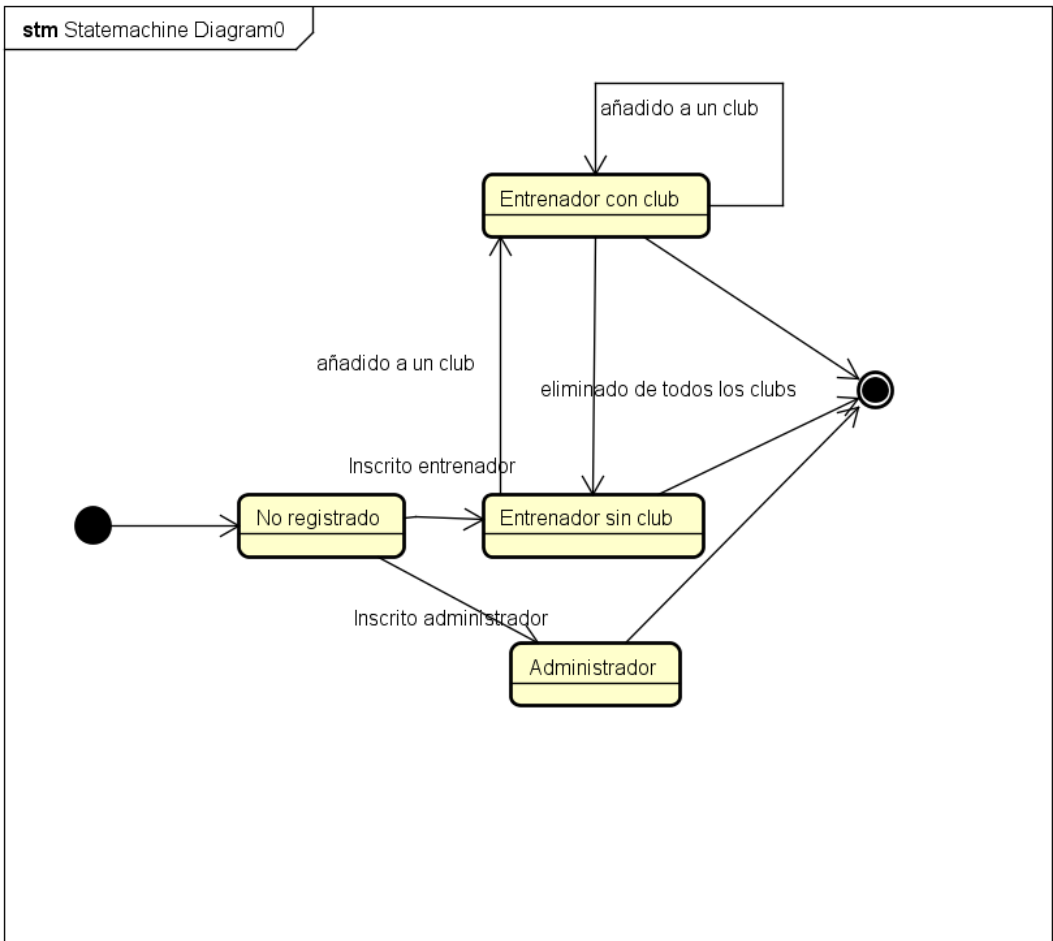


Figura 4.6: Máquina de estado de un usuario

Como podemos observar en la figura 4.6 al empezar la aplicación no hay ningún usuario registrado. Al registrarse se puede registrar como un entrenador o como un administrador. Al inscribirse como entrenador este empieza siendo un entrenador sin club. Cuando un administrador da de alta para su club a un entrenador este pasa a ser un entrenador con club. Aunque mas administradores den de alta a ese entrenador en su club este no cambiara de estado. Cuando todos los clubs den de baja a este entrenador de su club este volverá a ser un entrenador sin club.

4.9. Diagramas de secuencia

A continuación expondremos algunos diagramas de secuencia. Estos diagramas de secuencia están realizados en la etapa de análisis con el proyecto aún por desarrollar.

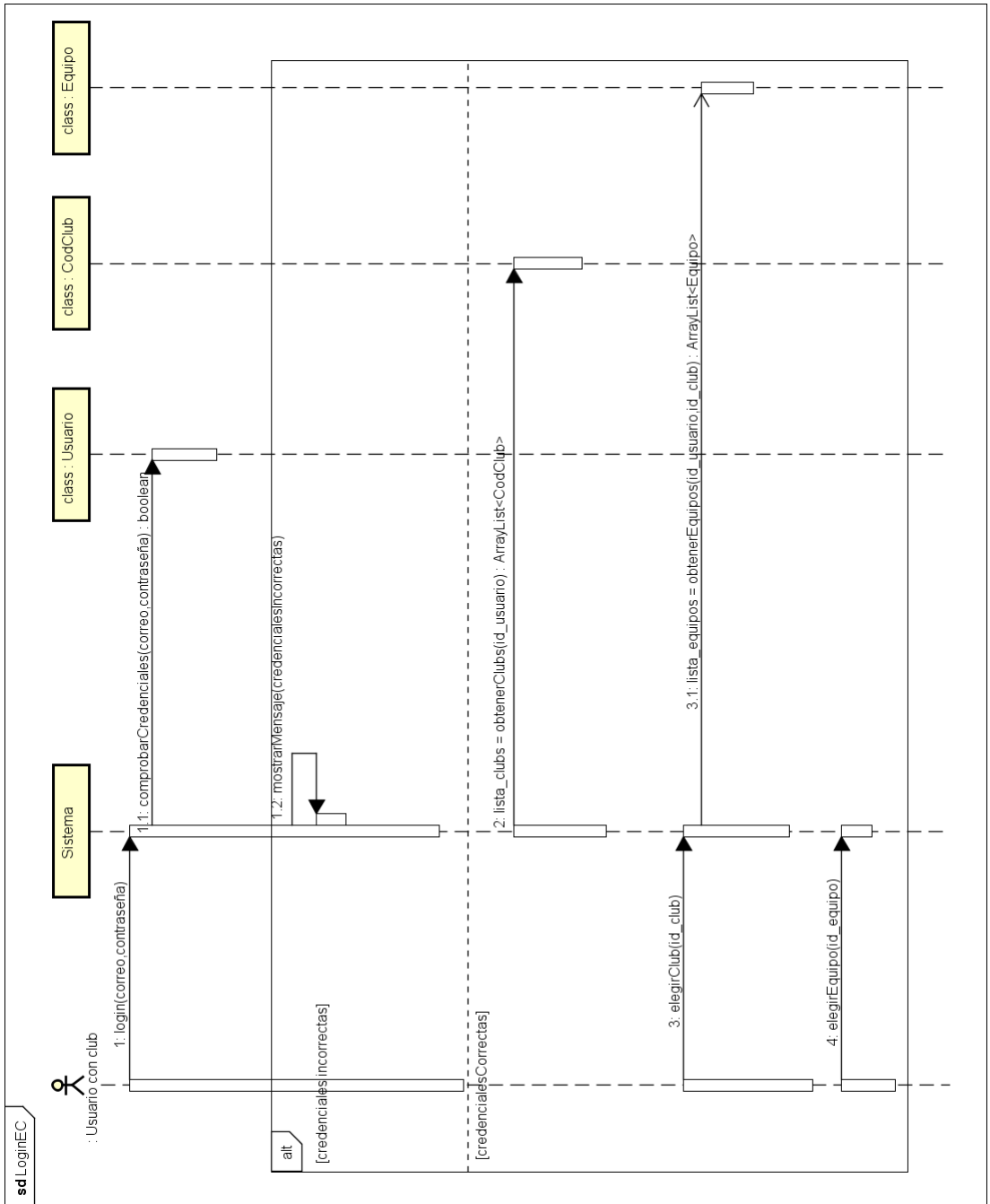


Figura 4.7: Diagrama de secuencia del ingreso de un entrenador con club

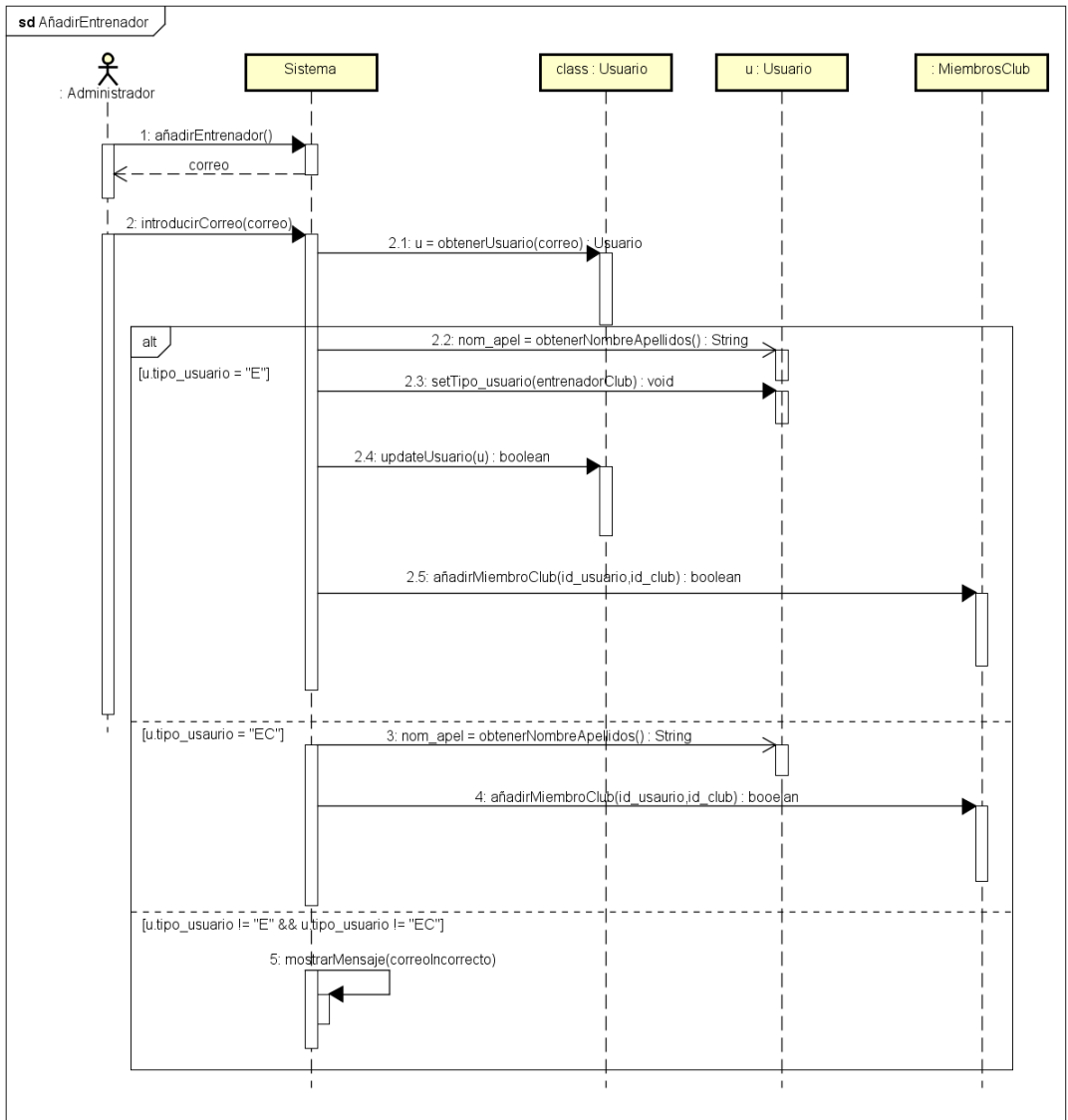


Figura 4.8: Diagrama de secuencia de añadir un entrenador

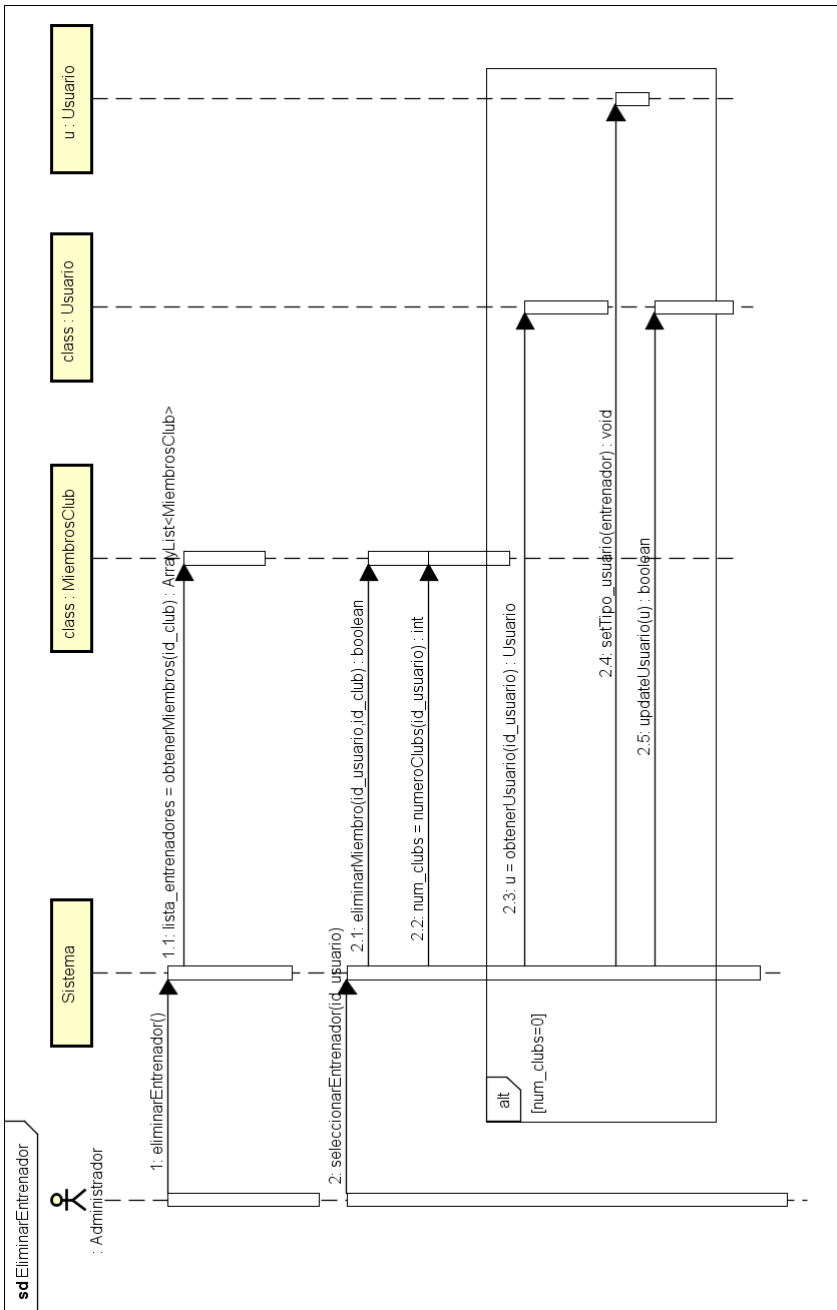


Figura 4.9: Diagrama de secuencia de eliminar un entrenador

El primer diagrama de secuencia corresponde al ingreso en la aplicación de un entrenador que tiene un club asignado que podemos ver en la descripción de caso de uso Ingresar, en la

tabla 4.2. En este diagrama de secuencia podemos observar como el usuario solicita ingresar. El sistema comprueba las credenciales introducidas. Si son incorrectas muestra un mensaje de error, si son correctas carga los clubs a los que pertenece el entrenador. Este selecciona el club con el que quiere trabajar y el sistema recarga los equipos de ese club a los que entrena y el usuario selecciona el equipo con el que quiere trabajar.

El segundo diagrama se corresponde al proceso de añadir un nuevo entrenador al club que podemos ver en la descripción de caso de uso Añadir entrenador al club, en la tabla 4.5. En este diagrama vemos como el administrador solicitar añadir un entrenador al club y el sistema le pide que introduzca el correo con el que esta registrado el entrenador. El sistema cagará el nombre y los apellidos del usuario al que corresponde el correo. Si este usuario no pertenecía a ningún club se le cambiara el tipo de usuario y se le añadirá al club. Si el usuario ya pertenecía a un club se le añadirá al nuevo club. Si es un administrador o el correo no pertenece a nadie registrado se mostrará un mensaje de error.

En el ultimo diagrama se corresponde al proceso de eliminar un entrenador de un club que podemos ver en la descripción de caso de uso Eliminar entrenador de club, en la tabla 4.6. En este diagrama podemos ver que el usuario pide eliminar un usuario y el sistema le muestra un listado de todos los entrenadores del club. El usuario selecciona el entrenador a borrar y el Sistema se encarga de borrarlo. Después revisa si sigue perteneciendo a algún club, si no es así cambia su tipo de usuario a entrenador sin club.

Capítulo 5

Diseño

5.1. Introducción

En este capítulo explicaremos el diseño de la aplicación cumpliendo con los requisitos mencionados en capítulos anteriores.

5.2. Arquitectura

“La arquitectura de software [3], consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software. Una arquitectura de software se selecciona y diseña con base a los requisitos y restricciones obtenidos en la fase de análisis”. Unas arquitecturas son más recomendables de implementar con ciertas tecnologías mientras que otras tecnologías no son aptas para determinadas arquitecturas. La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos.

En este proyecto lo que hemos utilizado ha sido una arquitectura siguiendo el patrón modelo vista controlador (MVC).

5.2.1. Patrón Modelo Vista Controlador

El patrón modelo vista controlador es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

A continuación se mostrará en un diagrama una pequeña representación de un patrón modelo vista controlador.

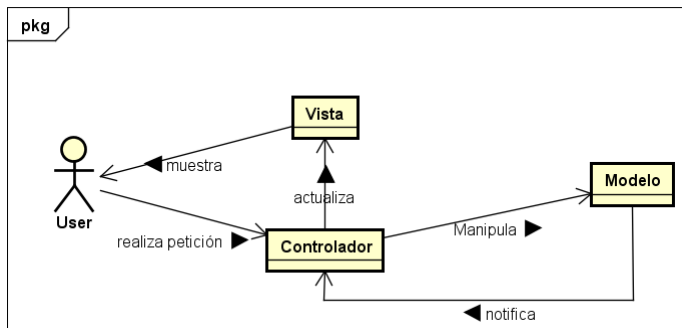


Figura 5.1: Diagrama de patrón modelo vista controlador.

Los componentes del MVC se podrían definir de la siguiente manera:

- **El Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones. Envía a la “vista” aquella parte de la información que en cada momento se le solicita para que sea mostrada. Las peticiones de acceso o manipulación de información llegan al modelo a través del controlador.

- **El Controlador:** Responde a eventos e invoca peticiones al “modelo” cuando se hace alguna solicitud sobre la información. También puede enviar comandos a su “vista” asociada si se solicita un cambio en la forma en que se presenta el “modelo”, por tanto se podría decir que el “controlador” hace de intermediario entre la “vista” y el “modelo”.

- **La Vista:** Presenta el modelo en un formato adecuado para interactuar, por tanto requiere de dicho modelo la información que debe representar como salida.

Viendo nuestro proyecto el patrón modelo vista controlador estaría formado por los siguientes paquetes

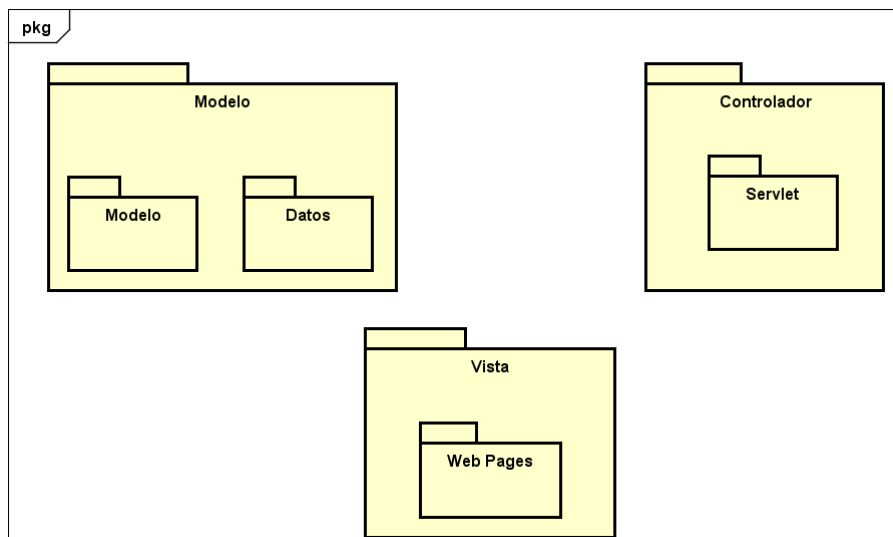


Figura 5.2: Composición del modelo la vista y el controlador del proyecto.

En este caso toda nuestra aplicación se encuentra en el lado del servidor. Por el lado del cliente tendríamos el navegador web que nos permite hacer peticiones a la aplicación.

De esta forma nuestro modelo estaría formado por las clases que hay tanto en el paquete modelo como en el paquete datos.

En el paquete modelo las clases que hay son las que representan la información con las que vamos a trabajar. Por otro lado en el paquete datos se encuentran las clases que se encargan tanto de consultar como de actualizar los datos que hay en la base de datos del proyecto.

El controlador estaría formado por los servlets, que son unas clases java utilizadas para ampliar las capacidades de un servidor. Aunque los servlets pueden responder a cualquier tipo de solicitudes, estos son utilizados comúnmente para extender las aplicaciones alojadas por servidores web, de tal manera que pueden ser vistos como applets de Java que se ejecutan en servidores en vez de navegadores web.

Por ultimo la vista estaría formada por el conjunto de paginas .jsp junto a las funciones JavaScript y los archivos css que dan estilo a dichas páginas.

5.3. Diseño de Base de Datos Relacionales

A continuación se muestra el diagrama de bases de datos relacional.

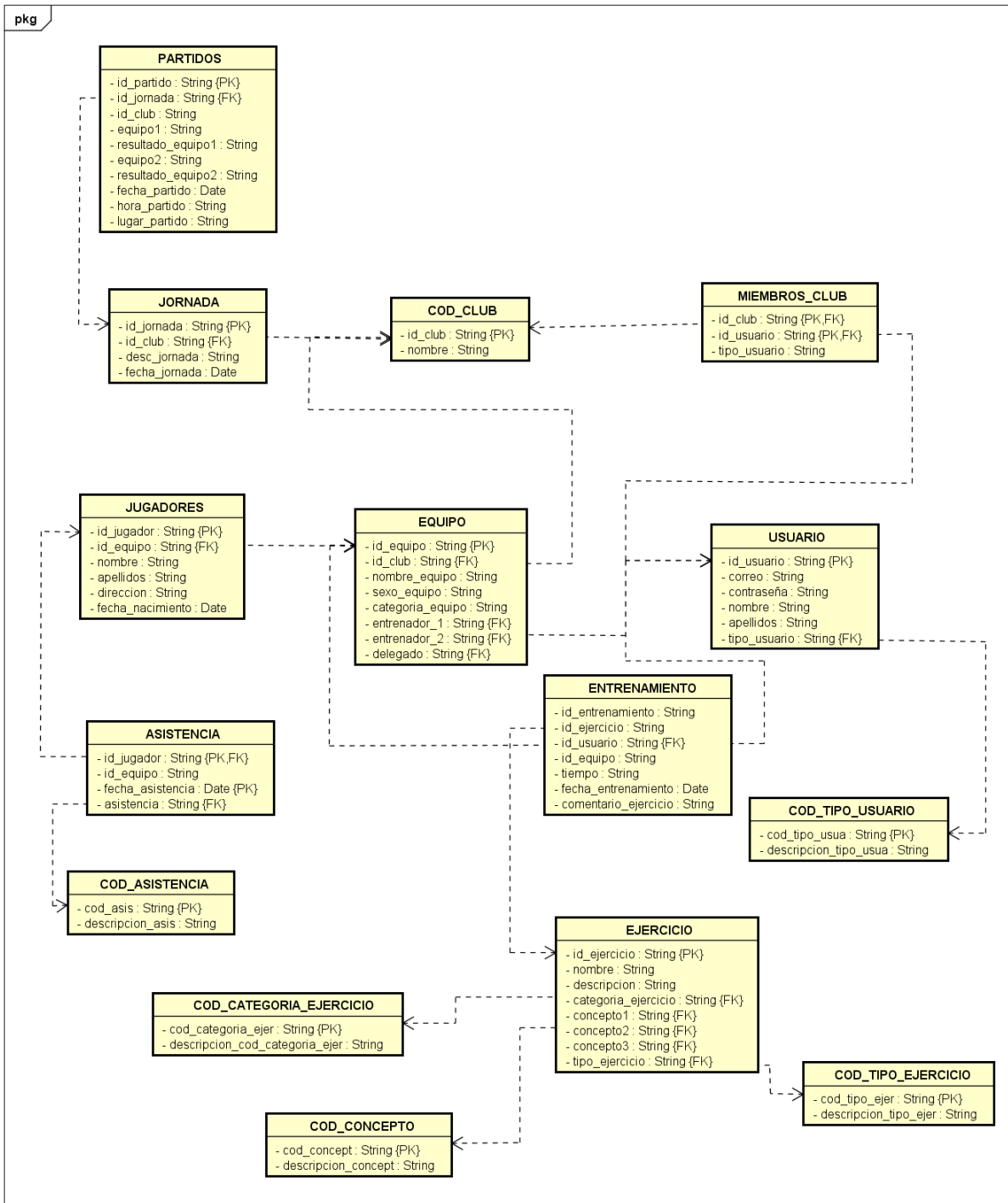


Figura 5.3: Diagrama de bases de datos relacional.

5.3.1. Tipos de consultas

A continuación se listarán las consultas que la base de datos deberá responder para poder obtener la funcionalidad deseada.

- Clubs a los que pertenece un entrenador
- Equipos dentro del club a los que entrena un entrenador
- Equipos de un club
- Entrenadores de un club.
- Jugadores de un equipo.
- Partidos de una jornada.
- Asistencia de los jugadores para una fecha.
- Entrenamientos de un equipo.
- Ejercicios que cumplen con las características introducidas.

5.3.2. Script de la base de datos

A continuación explicaremos principalmente las tablas de la base de datos que necesitan tener datos por defecto insertados para el correcto funcionamiento siguiendo el modelo relacional que podemos ver en la figura 5.3.

- **COD_ASISTENCIA:** En esta tabla estarán por defecto los valores que puede tomar el valor asistencia a la hora de que un entrenador apunte la asistencia de sus jugadores.
- **CATEGORIA_EJERCICIO:** En esta tabla estarán registrados las diferentes categorías a las que están destinadas los ejercicios.
- **COD_CONCEPTO:** En esta tabla están recogidos los diferentes conceptos que puede trabajar un ejercicio. En caso de que se quiera añadir otro concepto nuevo habría que insertarlo en esta tabla.
- **COD_TIPO_EJERCICIO:** En esta tabla se recogen los diferentes tipos a los que puede pertenecer un ejercicio.
- **EJERCICIO:** Aunque cualquier usuario de tipo entrenador puede añadir nuevos ejercicios a la base de datos ya vienen por defecto añadidos algunos ejercicios de uso común para facilitar las cosas a los entrenadores.

5.4. Diagrama de despliegue

Un diagrama de despliegue se utiliza para modelar la disposición física de los artefactos software en nodos. Muestra la arquitectura del sistema como el despliegue de los artefactos de software a los objetivos de despliegue.

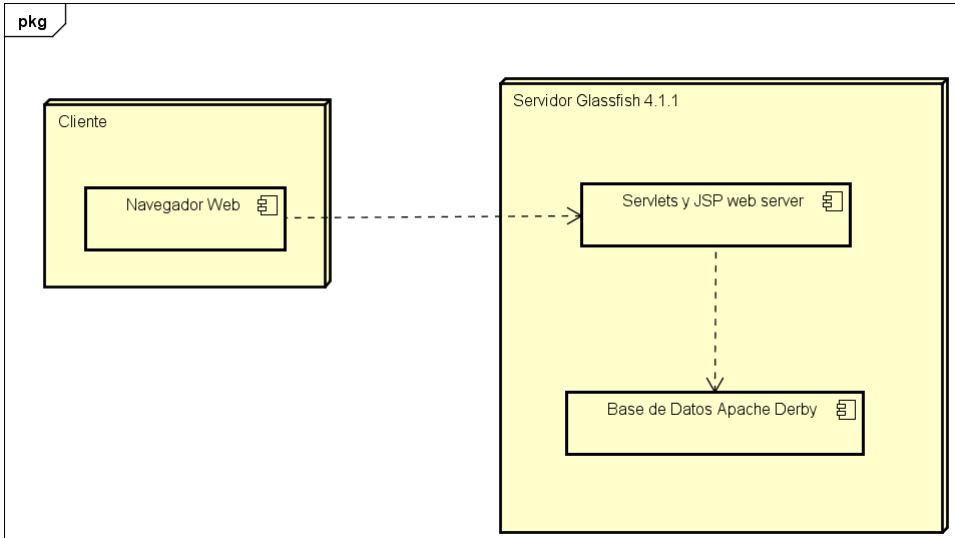


Figura 5.4: Diagrama de despliegue.

Capítulo 6

Implementación

En este capítulo explicaremos cómo, siguiendo la arquitectura explicada en el capítulo anterior, se ha desarrollado el proyecto.

Como ya se explicó con anterioridad los lenguajes y herramientas utilizadas para el desarrollo de la aplicación son:

- Java como lenguaje de servidor.
- HTML para la generación de las distintas vistas.
- JavaScript y JQuery para la validación de formularios, llamadas AJAX, etc.
- Materialize para dar estilo a las paginas HTML creadas.

Como ya se mencionó en el capítulo anterior el desarrollo del proyecto iba a seguir el patrón modelo vista controlador, a continuación se explicarán un poco más en profundidad las clases que nos permiten seguir este patrón.

6.1. Clases controlador

En nuestro proyecto las clases que harán de controlador serán los servlets. Los servlets son objetos que corren dentro del contexto de un servidor de aplicaciones y extienden su funcionalidad. El uso más común de los servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

Los servlets se desarrollaron como un avance respecto a la tecnología tradicional de CGI (Common Gateway Interface). Un servlet Java es un programa que, al igual que su similar en CGI, corre en un servidor web. El tipo de tareas que uno y otro pueden realizar son similares. Los servlets también procesan peticiones HTTP, ejecutan acciones (consultas a bases de datos, o conexiones a un host remoto, etc.) y generan respuestas en forma dinámica. A diferencia de los programas CGI, los servlets se ejecutan dentro de un contenedor web. Éste es en la práctica una máquina virtual Java que entrega una implementación de la API de Java para servlets. De esta manera, un servlet es una instancia de un `HTTPServlet`, administrada por el contenedor web para responder ante peticiones HTTP, mediante los métodos `doPost()` o `doGet()`.

```

1  import java.io.IOException;
2  import javax.servlet.RequestDispatcher;
3  import javax.servlet.ServletException;
4  import javax.servlet.http.HttpServlet;
5  import javax.servlet.http.HttpServletRequest;
6  import javax.servlet.http.HttpServletResponse;
7  import javax.servlet.http.HttpSession;
8
9  public class LogoutServlet extends HttpServlet {
10
11     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
12         throws ServletException, IOException {
13         response.setContentType("text/html;charset=UTF-8");
14         HttpSession sesion = request.getSession();
15         sesion.removeAttribute("usuario");
16         sesion.removeAttribute("club");
17         sesion.invalidate();
18         String url = "/index.jsp";
19         RequestDispatcher dispatcher = getServletContext().getRequestDispatcher(url);
20         dispatcher.forward(request, response);
21     }
22
23     @Override
24     protected void doGet(HttpServletRequest request, HttpServletResponse response)
25         throws ServletException, IOException {
26         processRequest(request, response);
27     }
28
29     @Override
30     protected void doPost(HttpServletRequest request, HttpServletResponse response)
31         throws ServletException, IOException {
32         processRequest(request, response);
33     }
34 }

```

Figura 6.1: Ejemplo de un servlet

6.2. Vista

Como ya hemos explicado para el desarrollo de las vistas usaremos páginas HTML. Además de páginas HTML usaremos la tecnología JSP que nos permite incluir código Java en páginas HTML. El contenedor JSP es el encargado de tomar la página, sustituir el código Java que contiene por el resultado de su ejecución, y enviarla al cliente. De esta forma podemos desarrollar fácilmente páginas con una parte fija y con otra variable.

Como también hemos explicado a estas páginas desarrolladas en HTML y JSP las vamos a dar estilo gracias a Materialize que es un framework que usando CSS y JavaScript nos permite de una forma muy fácil y sencilla basar nuestras interfaces en Material Design que es un estándar de diseño creado por Google.

Por último para nuestras llamadas asíncronas utilizaremos AJAX (Asynchronous Javascript

and XML), que nos permite obtener información del servidor sin recargar la página completa. Con AJAX es posible realizar una conexión a un servidor desde dentro de una página web usando un programa Javascript. Dicho servidor enviará una respuesta, esta respuesta se almacenará en una variable del programa Javascript y, una vez almacenada en la variable, podremos hacer con ella lo que deseemos.

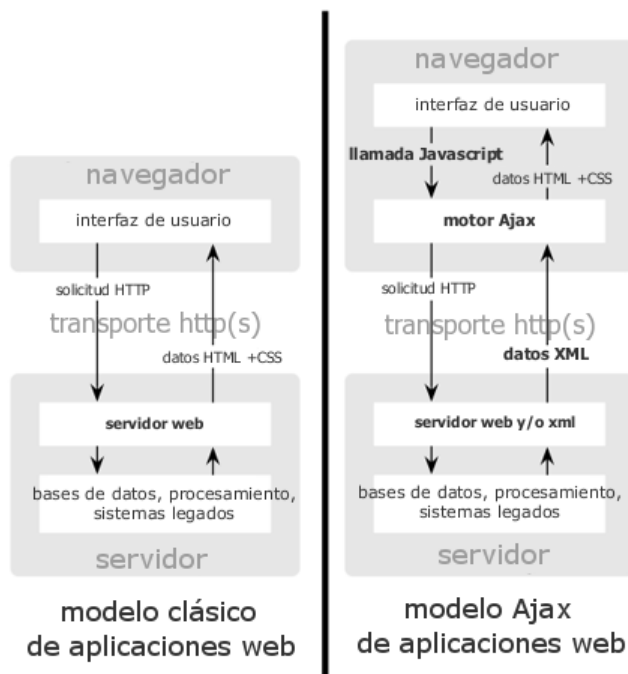


Figura 6.2: Comparación de Modelos de Aplicación Web clásico y basado en Ajax. Tomada de [7]

```

1 <%@page contentType="text/html" pageEncoding="UTF-8" %>
2 <!DOCTYPE html>
3 <html>
4 <head>
5   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" >
6   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1.0,
7     user-scalable=no" >
8   <meta http-equiv="X-UA-Compatible" content="IE=edge" >
9   <meta name="msapplication-tap-highlight" content="no" >
10  <title>Bappsket</title>
11  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" >
12  <link href="css/materialize.min.css" type="text/css" rel="stylesheet" media="screen,projection" >
13  <link href="css/style.css" type="text/css" rel="stylesheet" media="screen,projection" >
14  <link href="css/page-center.css" type="text/css" rel="stylesheet" media="screen,projection" >
15
16 </head>
17
18 <body class="blue darken-4" >
19   <div id="login-page" class="row" >
20     <div class="col s12 z-depth-4 card-panel" >
21       <form action="principal" method="post" >
22         <div class="row" >
23           <div class="input-field col s12 center" >
24             <p class="center login-form-text" >Mensaje:</p>
25           </div>
26         <div class="row" >
27           <div class="input-field col s12 center" >
28             <p class="center login-form-text" ><%=request.getParameter("mensaje") %></p>
29           </div>
30         </div>
31       </div>
32
33       <div class="row margin" >
34         <button class="btn waves-effect waves-light col s12 orange darken-4" type="submit"
35           id="ok" name="ok" >Aceptar
36         </button>
37       </div>
38       <div class="row margin" >
39         <br>
40         <br>
41       </div>
42     </form>
43   </div>
44 </div>
45 <script type="text/javascript" src="js/jquery.js" ></script>
46 <script type="text/javascript" src="js/materialize.min.js" ></script>
47 <script type="text/javascript" src="js/validarFormulario.js" ></script>
48 </body>
49 </html>

```

Figura 6.3: Ejemplo de una página HTML con Materialize

6.3. Clases Modelo

Las clases modelo son la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio).

En nuestro caso hemos separado las clases de modelo en dos paquetes, en el paquete modelo y en el paquete datos.

- Paquete modelo. En este paquete se encuentran las clases que representan la información con la que tiene que trabajar el sistema. Representan la lógica de negocio de la aplicación.
- Paquete datos. En este paquete se encuentran las clases encargadas de gestionar los accesos a la información que hay en la base de datos y que se representará en las clases del paquete modelo.

Capítulo 7

Pruebas

Una vez desarrollada la aplicación hay que realizar pruebas sobre ésta para poder asegurar un grado de fiabilidad de las tareas que realiza. Se van a realizar pruebas de caja negra que son aquellas que se utilizan para el estudio de un elemento desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. En este caso van a ser pruebas de sistema que son las encargadas de probar todo el sistema software desde el punto de vista de requisitos de la aplicación. Estas pruebas estarán dirigidas por casos de uso.

A continuación se mostrarán algunas de las pruebas realizadas.

CP-01	Añadir un entrenador al club
Descripción	Añadir un entrenador que no pertenecía al club.
Entrada	correo:entrenador1@gmail.com
Salida esperada	El entrenador empieza a formar parte del club.
Salida obtenida	Salida esperada.

Cuadro 7.1: CP-01: Añadir entrenador

CP-02	Eliminar un entrenador
Descripción	Eliminar un entrenador perteneciente al club.
Entrada	Alberto Gómez Fernández
Salida esperada	El entrenador deja formar parte del club.
Salida obtenida	Salida esperada.

Cuadro 7.2: CP-02: Eliminar entrenador

CP-03	Añadir un entrenador al club que ya pertenece al club
Descripción	Intentamos añadir un entrenador que ya pertenece al club.
Entrada	correo:entrenador1@gmail.com
Salida esperada	Mensaje de que ese entrenador ya forma parte del club.
Salida obtenida	Salida esperada.

Cuadro 7.3: CP-013: Añadir entrenador ya existente

CP-04	Crear un equipo con un mismo entrenador como 1º y como 2º
Descripción	Crear un equipo donde la misma persona es el 1er y el 2º entrenador.
Entrada	1er entrenador: Alberto Gómez Fernández, 2º entrenador: Alberto Gómex Fernández
Salida esperada	Mensaje de que una misma persona no puede ocupar dos cargos en un equipo.
Salida obtenida	Salida esperada.

Cuadro 7.4: CP-04: Crear equipo con mismo 1er y 2º entrenador

CP-05	Crear equipo
Descripción	Crear un equipo para el club.
Entrada	Nombre:Benjamin A, Categoría: Benjamín 1er año, Sexo : Masculino, 1er entrenador: Alberto Gómez Fernández. Jugador1 Nombre: Alberto Apellidos: Ramón Alonso Dirección: Avd.Prado Boyal 3 Fecha: DEC 12, 2008 Jugador2 Nombre: Sergio Apellidos: Pérez Casado Dirección: Calle Irún 3 Fecha: FEB 12, 2008 Jugador3 Nombre: Rafael Apellidos: Hernández López Dirección:Calle Pensamiento2 Fecha: MAY 14, 2008 Jugador4 Nombre: Jose Apellidos: Izaguirre de la Fuente Dirección: Avd. Madrid 12 Fecha: JUN 17, 2008 Jugador5 Nombre: Javier Apellidos: Izaguirre de la Fuente Dirección: Avd. Madrid 12 Fecha: JUN 17, 2008 Jugador6 Nombre: Manuel Apellidos: Pérez Vaca Dirección: Calle Gerona 8 Fecha: MAY 19, 2008 Jugador7 Nombre: Lucas Apellidos: Bueno Manjón Dirección:Calle López de Vega Fecha: APR 15, 2008 Jugador8 Nombre: Pablo Apellidos: Carrión Mejías Dirección: Calle Caamaño 34 Fecha: FEB 1, 2008
Salida esperada	Equipo añadido al club.
Salida obtenida	Error al parsear "DEC 12. 2008".

Cuadro 7.5: CP-05: Crear equipo

El error que podemos observar en la tabla 7.5 se producía al intentar parsear ciertas fechas. Este error venía dado por el idioma en el que el datapicker de Materialize ponía la fechas. Basto con usar Locale.ENGLISH en el formateo de las fechas para solucionar el error. Una vez solucionado este error se procede a volver a realizar la prueba de crear un equipo.

CP-06	Crear equipo
Descripción	Crear un equipo para el club.
Entrada	Nombre:Benjamin A, Categoría: Benjamín 1er año, Sexo : Masculino, 1er entrenador: Alberto Gómez Fernández. Jugador1 Nombre: Alberto Apellidos: Ramón Alonso Dirección: Avd.Prado Boyal 3 Fecha: DEC 12, 2008 Jugador2 Nombre: Sergio Apellidos: Pérez Casado Dirección: Calle Irún 3 Fecha: FEB 12, 2008 Jugador3 Nombre: Rafael Apellidos: Hernández López Dirección:Calle Pensamiento2 Fecha: MAY 14, 2008 Jugador4 Nombre: Jose Apellidos: Izaguirre de la Fuente Dirección: Avd. Madrid 12 Fecha: JUN 17, 2008 Jugador5 Nombre: Javier Apellidos: Izaguirre de la Fuente Dirección: Avd. Madrid 12 Fecha: JUN 17, 2008 Jugador6 Nombre: Manuel Apellidos: Pérez Vaca Dirección: Calle Gerona 8 Fecha: MAY 19, 2008 Jugador7 Nombre: Lucas Apellidos: Bueno Manjón Dirección:Calle López de Vega Fecha: APR 15, 2008 Jugador8 Nombre: Pablo Apellidos: Carrión Mejías Dirección: Calle Caamaño 34 Fecha: FEB 1, 2008
Salida esperada	Equipo añadido al club.
Salida obtenida	Salida esperada.

Cuadro 7.6: CP-06: Crear equipo

CP-07	Modificar Equipo
Descripción	Modificar un equipo del club.
Entrada	Nombre:Benjamin A, Categoría: Benjamín 1er año, Sexo : Masculino, 1er entrenador: Alberto Gómez Fernández, 2º entrenador: Ramón García López. Jugador1 Nombre: Alberto Apellidos: Ramón Alonso Dirección: Avd.Prado Boyal 3 Fecha: DEC 13, 2008 Jugador2 Nombre: Sergio Apellidos: Pérez Casado Dirección: Calle Irún 3 Fecha: FEB 12, 2008 Jugador3 Nombre: Rafael Apellidos: Hernández López Dirección:Calle Pensamiento2 Fecha: MAY 14, 2008 Jugador4 Nombre: Jose Apellidos: Izaguirre de la Fuente Dirección: Avd. Madrid 12 Fecha: JUN 17, 2008 Jugador5 Nombre: Javier Apellidos: Izaguirre de la Fuente Dirección: Avd. Madrid 12 Fecha: JUN 17, 2008 Jugador6 Nombre: Manuel Apellidos: Pérez Vaca Dirección: Calle Gerona 8 Fecha: MAY 19, 2008 Jugador7 Nombre: Lucas Apellidos: Bueno Manjón Dirección:Calle López de Vega Fecha: APR 15, 2008 Jugador8 Nombre: Pablo Apellidos: Carrión Mejías Dirección: Calle Caamaño 34 Fecha: FEB 1, 2008
Salida esperada	Equipo modificado correctamente.
Salida obtenida	Salida esperada.

Cuadro 7.7: CP-07: Modificar equipo

CP-08	Crear una jornada
Descripción	Crear una jornada nueva.
Entrada	Fecha inicio jornada: JUL 09, 2018 Equipo1: Benjamin A San Agustín Equipo2: Benjamín San Isidro B Fecha: JUL 09, 2018 Hora: 20:30 Lugar Polideportivo San Agustín.
Salida esperada	Jornada añadida.
Salida obtenida	Salida esperada.

Cuadro 7.8: CP-08: Crear jornada

Como precondition a esta prueba no debe de haber ningún equipo registrado en el club en el que se intenta dar de alta una jornada.

CP-09	Crear una jornada sin tener equipos registrados
Descripción	Crear una jornada nueva pero sin tener ningún equipo dado de alta.
Entrada	Fecha inicio jornada: JUL 09, 2018 .
Salida esperada	Mensaje de error por no tener equipos creados.
Salida obtenida	Salida esperada.

Cuadro 7.9: CP-09: Crear jornada

CP-10	Modificar una jornada
Descripción	Modificar una jornada creada previamente.
Entrada	Fecha inicio jornada: JUL 09, 2018 Equipo1: Benjamin A San Agustín Equipo2: Benjamín San Isidro B Fecha: JUL 09, 2018 Hora: 20:45 Lugar Polideportivo San Agustín.
Salida esperada	Jornada modificada.
Salida obtenida	Salida esperada.

Cuadro 7.10: CP-10: Modificar una jornada

CP-11	Descarga de los resultados de una jornada
Descripción	Obtener un EXCEL con los resultados de una jornada.
Entrada	Fecha inicio jornada: JUL 09, 2018
Salida esperada	Un EXCEL con los resultados.
Salida obtenida	Salida esperada.

Cuadro 7.11: CP-11: Modificar una jornada

CP-12	Añadir asistencia
Descripción	Añadir la asistencia de los jugadores para una fecha.
Entrada	Fecha asistencia: JUL 09, 2018, Asistencia jugador1: Asistido, jugador2: Asistido, jugador3: Asistido, jugador4: Asistido, jugador5: Asistido, jugador6: Asistido, jugador7: Asistido, jugador8: Asistido
Salida esperada	Asistencia registrada correctamente.
Salida obtenida	Salida esperada.

Cuadro 7.12: CP-12: Añadir asistencia

CP-13	Modificar asistencia
Descripción	Modificar la asistencia de los jugadores para una fecha ya usada.
Entrada	Fecha asistencia: JUL 09, 2018, Asistencia jugador1: Llega tarde, jugador2: Asistido, jugador3: Asistido, jugador4: Asistido, jugador5: Asistido, jugador6: Asistido, jugador7: Asistido, jugador8: Asistido
Salida esperada	Asistencia registrada correctamente.
Salida obtenida	Salida esperada.

Cuadro 7.13: CP-13: Modificar asistencia

CP-14	Crear un entrenamiento
Descripción	Crear un entrenamiento nuevo para una fecha.
Entrada	Fecha entrenamiento: JUL 09, 2018, ejercicio bote con las dos manos, tiempo 20 min, ejercicio zig zag, tiempo 20 min, ejercicio rueda de entradas, tiempo 20 min.
Salida esperada	Crea un entrenamiento para la fecha elegida.
Salida obtenida	Salida esperada.

Cuadro 7.14: CP-14: Crear entrenamiento

CP-15	Modificar un entrenamiento
Descripción	Modificar un entrenamiento creado con anterioridad.
Entrada	Fecha entrenamiento: JUL 09, 2018, ejercicio bote con las dos manos, tiempo 20 min, ejercicio zig zag, tiempo 20 min, ejercicio rueda de tiro, tiempo 20 min.
Salida esperada	Modifica el entrenamiento creado.
Salida obtenida	Salida esperada.

Cuadro 7.15: CP-15: Modificar entrenamiento

CP-16	Añadir un comentario
Descripción	Se añade un comentario a un ejercicio de un entrenamiento.
Entrada	Fecha entrenamiento: JUL 09, 2018, ejercicio bote con las dos manos, tiempo 20 min, ejercicio zig zag, tiempo 20 min, ejercicio rueda de entradas, tiempo 20 min, comentario 10 minutos por el lado derecho y 10 minutos por el lado izquierdo..
Salida esperada	Añade un comentario al ejercicio de el entrenamiento seleccionado.
Salida obtenida	Salida esperada.

Cuadro 7.16: CP-16: Añadir un comentario

CP-17	Crear un entrenamiento en un día en el que ya hay un entrenamiento.
Descripción	Crear un entrenamiento nuevo para una fecha en la que ya existe un entrenamiento.
Entrada	Fecha entrenamiento: JUL 09, 2018, ejercicio bote con las dos manos, tiempo 20 min, ejercicio zig zag, tiempo 20 min, ejercicio rueda de entradas, tiempo 20 min.
Salida esperada	Mensaje de error.
Salida obtenida	Salida esperada.

Cuadro 7.17: CP-17: Crear entrenamiento

CP-18	Añadir resultado de partido.
Descripción	Se añade el resultado de un partido de la jornada seleccionada.
Entrada	Fecha jornada: JUL 09, 2018, resultado1 50, resultado2 45.
Salida esperada	Modificación de los datos del partido para mostrar el resultado introducido.
Salida obtenida	Salida esperada.

Cuadro 7.18: CP-18: Crear entrenamiento

CP-19	Ver resultados.
Descripción	Ver los resultados de los partidos de la jornada seleccionada.
Entrada	Fecha jornada: JUL 09, 2018.
Salida esperada	Los resultados de la jornada seleccionada.
Salida obtenida	Salida esperada.

Cuadro 7.19: CP-19: Ver resultados

CP-20	Añadir un nuevo ejercicio.
Descripción	Añadir un nuevo ejercicio a la base de datos de ejercicios.
Entrada	nombre Rueda de entradas, descripción Se divide al equipo en dos y se colocan en la intersección de la línea de medio campo con la de banda mirando cada grupo hacia una canasta. El primero de cada fila hace una entrada en la canasta a la que mira y se coloca en la siguiente fila categoría benjamín, concepto1 Entradas concepto2 “” concepto3 “”, tipo ejercicio técnica individual.
Salida esperada	Se añade un ejercicio nuevo a la base de datos.
Salida obtenida	Salida esperada.

Cuadro 7.20: CP-20: Ver resultados

CP-21	Cambiar el club o el equipo con el que se trabaja.
Descripción	Cambiar el club o el equipo con el que se trabaja.
Entrada	club San José, equipo Junior A.
Salida esperada	Se cambia el club y/o el equipo con el que se trabaja y muestra los datos relacionados con este.
Salida obtenida	Salida esperada.

Cuadro 7.21: CP-21: Cambiar club/equipo trabajo

Capítulo 8

Conclusiones

El desarrollo de este proyecto me ha ayudado a entender lo importante que es una buena planificación en proyectos de media envergadura para poder ajustarse a los tiempos de entrega. Esto es algo que durante la carrera no tenemos mucho en cuenta. Aunque en cada entrega que se nos plantea en la carrera mas o menos nos intentamos planificar, lo solemos hacer muy por encima y muchas veces acabamos no haciendo caso a esa planificación ni dándola ninguna importancia. Los errores en la planificación inicial se pueden deber precisamente a esto a que no tengo mucha experiencia en la planificación de este tipo de proyectos, e incluso en una sobre valoración de mis propias capacidades, haciendo que pensase que podría realizar ciertas acciones mucho más rápido de lo que en verdad soy capaz. Gracias a este proyecto también he podido aprender tecnologías y frameworks con los que no tenía tanta relación y que tenía ganas de aprender como puede ser Materialize. Finalmente el resultado de este proyecto es una aplicación funcional que cumple con los requisitos y objetivos obtenidos en las fases de análisis.

Una de las características a destacar del proyecto es la portabilidad que nos ofrece una aplicación web ya que para simplemente se necesita un navegador con conexión al servidor. Gracias a esto nuestra aplicación puede ser utilizada en diferentes plataformas. En este caso durante el desarrollo de este proyecto se ha estado comprobando sobre diferentes sistemas operativos, para ser precisos OS X y Windows y también sobre diferentes Navegadores, en este caso Safari y Google Chrome.

El desarrollo de este proyecto me ha vuelto a acercar a fases del desarrollo a las que ya no estaba tan acostumbrado como puede ser la de análisis o el desarrollo de diagramas, ya que en los proyectos en los que trabajo actualmente ya llevan bastante tiempo en desarrollo.

En el aspecto personal puedo decir que el resultado es muy satisfactorio ya que aunque han sido unos meses difíciles por la compaginación de este proyecto con el trabajo han merecido la pena. Estas dificultades me han ayudado a aprender a planificar mi tiempo mucho mejor y ha ser mucho más productivo.

8.1. Trabajo futuro

Una de las primeras cosas que hay que hacer en un futuro es poder mostrar la aplicación a usuarios potenciales para recibir retroalimentación por su parte. Esto no se ha podido

realizar durante el desarrollo del proyecto ya que la mayoría de mis compañeros son estudiantes y la época de desarrollo de este proyecto ha coincidido con épocas de exámenes. A parte de esto se sopesan las siguientes mejoras:

- Realizar una aplicación móvil ya que esto ayudaría aun más en la parte de la administración de equipos por parte de los entrenadores.
- Permitir la subida y la visualización de la planificación del equipo en formato PDF.
- Generar estadísticas sobre los ejercicios entrenados a lo largo del año. Estas estadísticas se podrían exportar en un fichero EXCEL al igual que la asistencia o los resultados de una jornada. Estas estadísticas podrían ser un resumen con cada uno de los ejercicios de la base de datos y las veces que se ha trabajado cada uno a lo largo del año o un resumen de los conceptos que un jugador ha trabajado menos que el resto de sus compañeros por faltar a los entrenamientos.

Bibliografía

- [1] Fundación municipal de deportes de valladolid. <http://www.fmdva.org/>.
- [2] Alex Mark Kevin Louie Alvin Wang, Alan Chang. Framework materialize. <https://materializecss.com>.
- [3] Anonimo. Arquitectura en software. <https://arquitecturatecnologica455.wordpress.com/2016/08/15/arquitectura-en-software/>.
- [4] Apache. Apache derby. <https://db.apache.org/derby/index.html>, 2018.
- [5] Google. Material design. <https://material.io/>.
- [6] The Adecco Group. Oferta y demanda de empleo en españa 2016. <https://www.infoempleo.com/informe-infoempleo-adecco/>, 2016.
- [7] Digital Learning. ¿qué es ajax? <https://www.digitallearning.es/blog/que-es-ajax/>, mar 2012.
- [8] Polytechnique Montréal. Unified process for education. http://www.upedu.org/process/ovu_proc.htm, 2014.
- [9] Oracle. Glassfish server. <https://javaee.github.io/glassfish/>, 2017.
- [10] Abraham Otero. *Tutorial básico de Java EE*, 2010.
- [11] John Resig. Jquery api documentation. <https://api.jquery.com>.
- [12] StatCounter Global Stats. Browser market share worldwide. <http://gs.statcounter.com/browser-market-share#monthly-201706-201806-bar>, jul 2018.

Apéndice A

Contenido del CD

El CD contiene en su interior los siguientes ficheros y documentos:

- **Manual de usuario.** Contiene las instrucciones de como funciona la aplicación.
- **Manual de mantenimiento.** Contiene las instrucciones para el mantenimiento de la aplicación
- **Memoria.** Memoria del TFG en formato PDF.
- **Código fuente.** Contiene los ficheros para el correcto funcionamiento de la aplicación
- **Script de la base de datos.** Script para crear las tablas de la base de datos.

Apéndice B

Manual de mantenimiento

B.1. Introducción

En este apartado de la memoria se explicarán los pasos a seguir para poder utilizar la aplicación.

B.2. Requisitos

B.2.1. Servidor web

Para el funcionamiento de esta aplicación es necesario un servidor web, durante el desarrollo de el proyecto se ha utilizado un servido GlassFish [9] 4.1.1.

Ya que nuestra aplicación de momento solo se puede usar en local crearemos un servidor GlassFish en nuestro entorno de desarrollo favorito. Para el desarrollo de este proyecto se usó NetBeans y a continuación se explicará como crear un servidor GlasFish en este entorno.

Para la instalación del servidor lo primero que tenemos que hacer es descargarnos el propio servidor de su página <https://javaee.github.io/glassfish/>.

Después descomprimiremos el .zip que se nos ha descargado y lo almacenaremos en la carpeta que prefiramos. Acto seguido desde el propio NetBeans iremos a la pestaña Services y haremos click derecho en el apartado Servers seleccionando la opción Add Server. Esto nos abrirá una pestaña donde seleccionamos la opción GlassFish Server. En la siguiente pantalla en el apartado Installation Location: pondremos la ruta a la carpeta donde hemos guardado los archivos de GlassFish descargados. En la siguiente pantalla dejaremos las opciones de dominio tal y como vienen y daremos a finish para terminar.

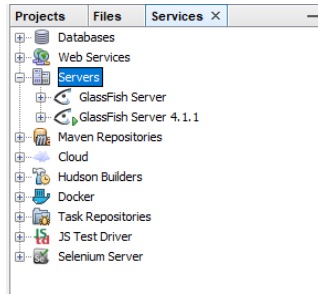


Figura B.1: Añadimos un nuevo server.

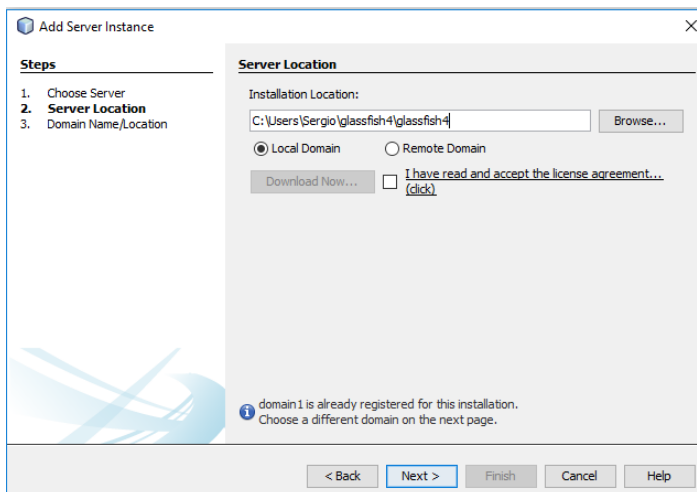


Figura B.2: Seleccionamos la carpeta donde tenemos los archivos de GlassFish.

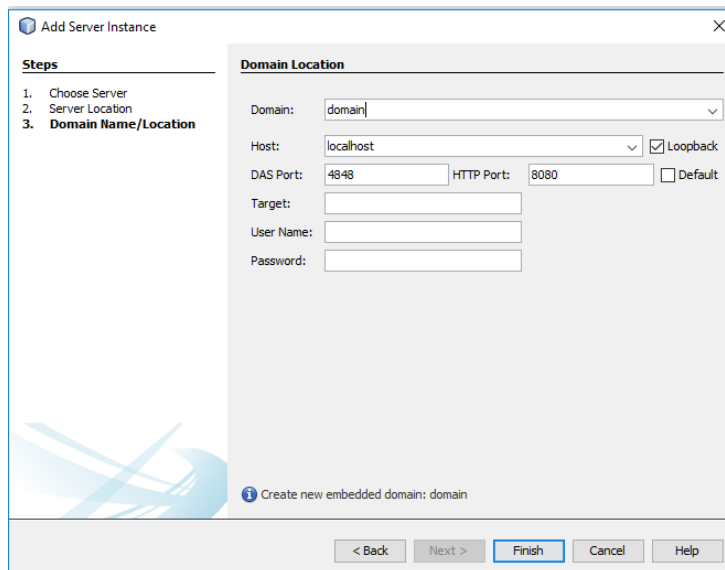


Figura B.3: Dejamos la pantalla de dominio tal cual..

B.2.2. Base de datos

Para la base de datos como ya hemos explicado con anterioridad hemos usado una base de datos Apache Derby [4].

Para la instalación de Apache Derby lo primero que tenemos que hacer es descargarnos las librerías de Apache Derby de su página

https://db.apache.org/derby/derby_downloads.html. Al igual que con los archivos de GlassFish descomprimos el .zip que nos descargamos y lo almacenamos en la carpeta que deseemos. Después de esto vamos de nuevo a la pestaña Services de NetBeans y desplegamos la opción de Databases y sobre Java DB hacemos click derecho y elegimos la opción de Create Database... Se nos desplegará un formulario a rellenar donde tendremos que poner los siguientes valores:

- **Database Name:** BAPPSKET

- **User Name:** bappsket

- **Password:** 1234

- **Confirm Password:** 1234

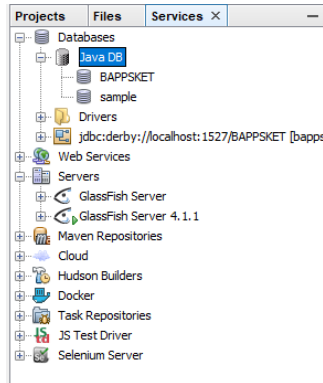


Figura B.4: Añadimos una nueva base de datos.

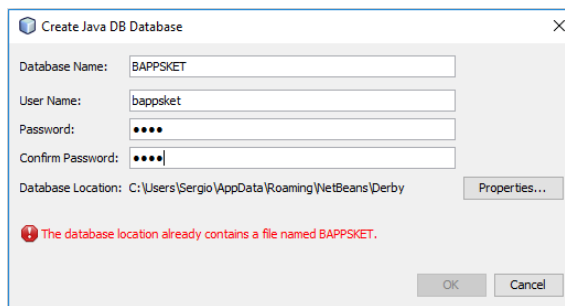


Figura B.5: Rellenamos los datos

Una vez creada la base de datos se nos habrá creado una conexión. Hacemos click derecho sobre esa conexión y seleccionamos la opción de Execute Comand...

Se nos abrirá una página de editor de SQL. En esta página pegaremos el script que encontraremos en el CD y lo ejecutaremos dando al primer botón que nos sale al lado del nombre de la conexión.

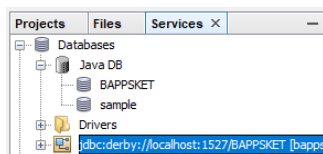
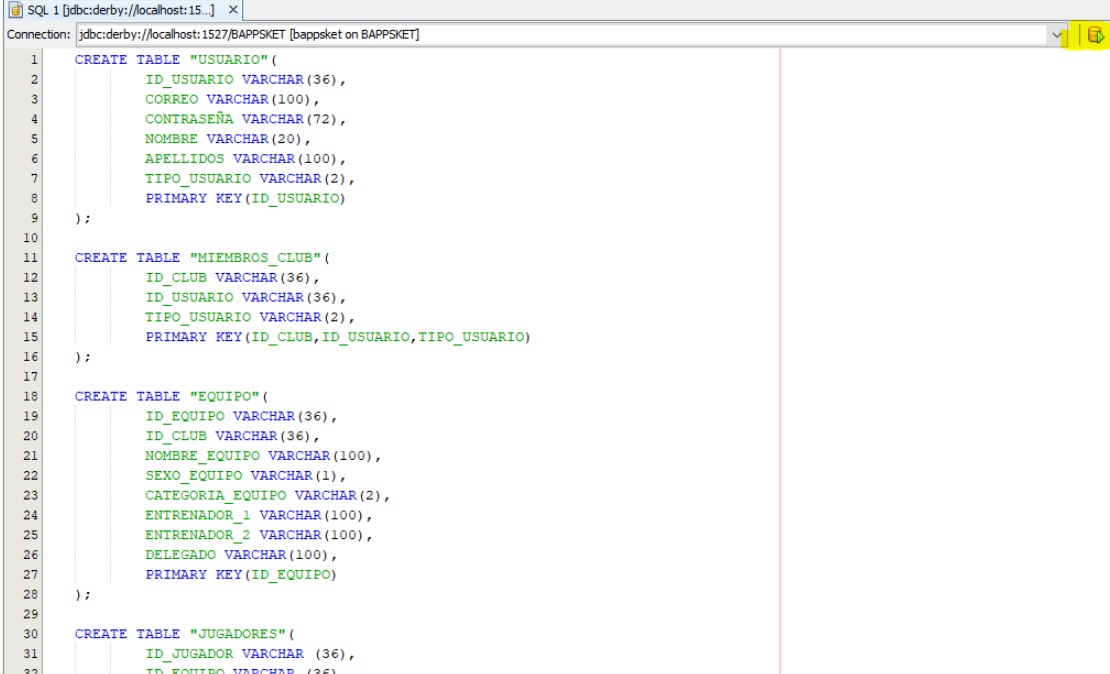


Figura B.6: Abrimos la página para editar SQL



The screenshot shows a window titled "SQL 1 [jdbc:derby://localhost:1527/BAPPSKET]". The connection string is "jdbc:derby://localhost:1527/BAPPSKET [bappsket on BAPPSKET]". The main area contains the following SQL script:

```
1 CREATE TABLE "USUARIO" (  
2     ID_USUARIO VARCHAR(36),  
3     CORREO VARCHAR(100),  
4     CONTRASEÑA VARCHAR(72),  
5     NOMBRE VARCHAR(20),  
6     APELLIDOS VARCHAR(100),  
7     TIPO_USUARIO VARCHAR(2),  
8     PRIMARY KEY(ID_USUARIO)  
9 );  
10  
11 CREATE TABLE "MIEMBROS_CLUB" (  
12     ID_CLUB VARCHAR(36),  
13     ID_USUARIO VARCHAR(36),  
14     TIPO_USUARIO VARCHAR(2),  
15     PRIMARY KEY(ID_CLUB, ID_USUARIO, TIPO_USUARIO)  
16 );  
17  
18 CREATE TABLE "EQUIPO" (  
19     ID_EQUIPO VARCHAR(36),  
20     ID_CLUB VARCHAR(36),  
21     NOMBRE_EQUIPO VARCHAR(100),  
22     SEXO_EQUIPO VARCHAR(1),  
23     CATEGORIA_EQUIPO VARCHAR(2),  
24     ENTRENADOR_1 VARCHAR(100),  
25     ENTRENADOR_2 VARCHAR(100),  
26     DELEGADO VARCHAR(100),  
27     PRIMARY KEY(ID_EQUIPO)  
28 );  
29  
30 CREATE TABLE "JUGADORES" (  
31     ID_JUGADOR VARCHAR(36),  
32     ID_EQUIPO VARCHAR(36)
```

Figura B.7: Ejecutamos Script

B.2.3. Código

Por último importaremos el proyecto que se encuentra en el CD a nuestro NetBeans. Para ello desde la pantalla de Projects haremos click derecho y seleccionaremos la opción de Open Project... Ahí seleccionaremos el proyecto que se encuentra en el CD y solo nos faltaría ejecutarlo para poder hacer uso de la aplicación.

Apéndice C

Manual de usuario.

En este capítulo explicaremos como llevar a cabo cada una de las tareas que permite realizar la aplicación. Como hay varios roles se seguirá un orden. Primero se explicaran las tareas del Administrador, y después las de los entrenadores, tanto con club como sin club. En las tareas que son propias de cualquier usuario se indicará al inicio de la descripción.

C.1. Menú de inicio, ingreso y registro

El menú de inicio es el mismo para cualquier rol y para poder acceder a la aplicación lo único que tendremos que hacer es introducir el correo y la contraseña con la que nos hemos registrado en la aplicación.

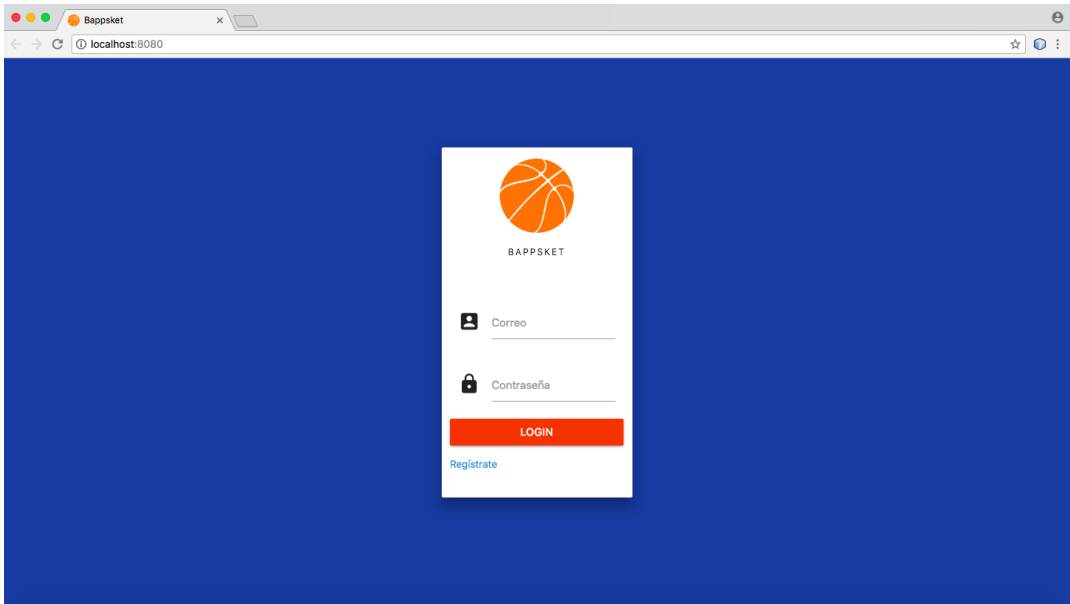


Figura C.1: Pagina de login

Una vez introducidas las credenciales si somos usuarios de tipo administrador se nos mostrará la pagina de inicio propia de los administradores donde podemos ver los equipos de nuestro club y los resultados de la ultima jornada.

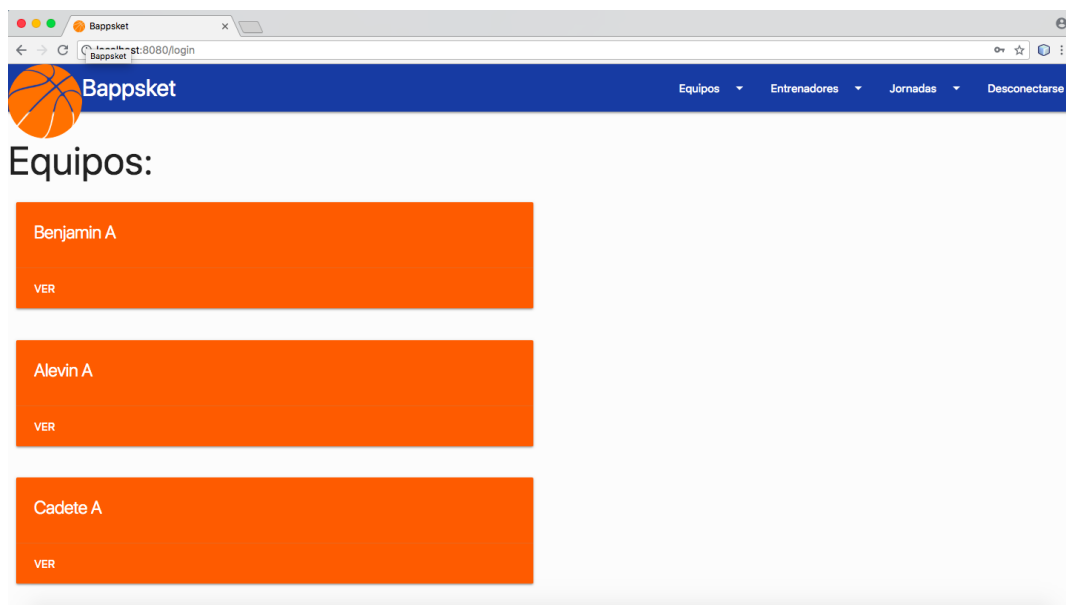


Figura C.2: Pantalla principal de un administrador

Si por el contrario somos entrenadores nos saldrá una pantalla para que elijamos, si es que tenemos, un club y un equipo con el que trabajar. Una vez seleccionado el club y el equipo se nos mostrará la pantalla de inicio propia de un entrenador donde podemos ver los diez últimos entrenamientos que se han hecho y si se pertenece a un club los resultados de la última jornada.

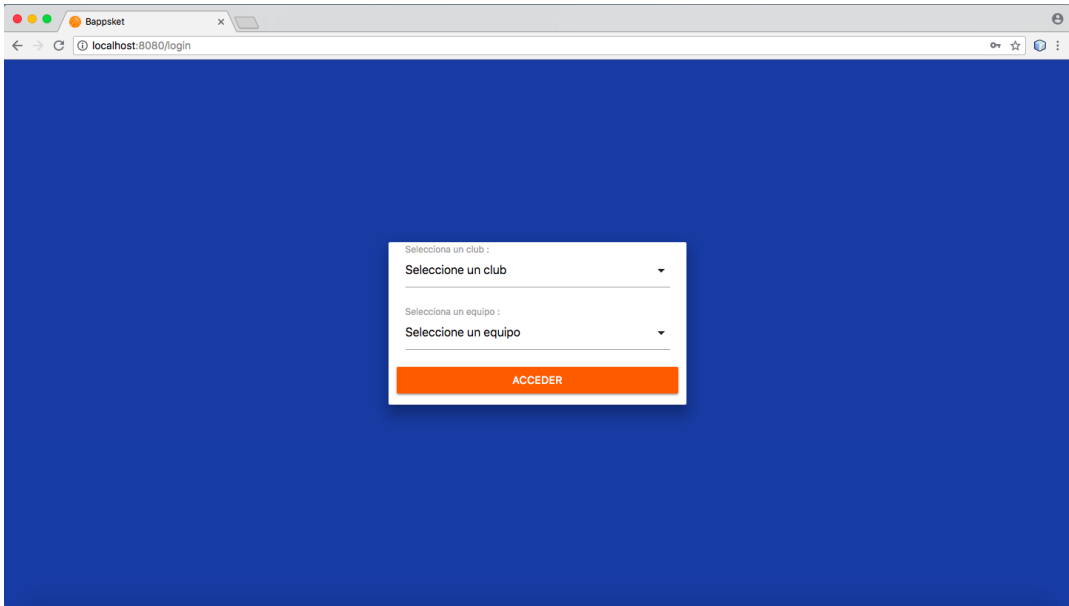


Figura C.3: Pantalla elección club/equipo

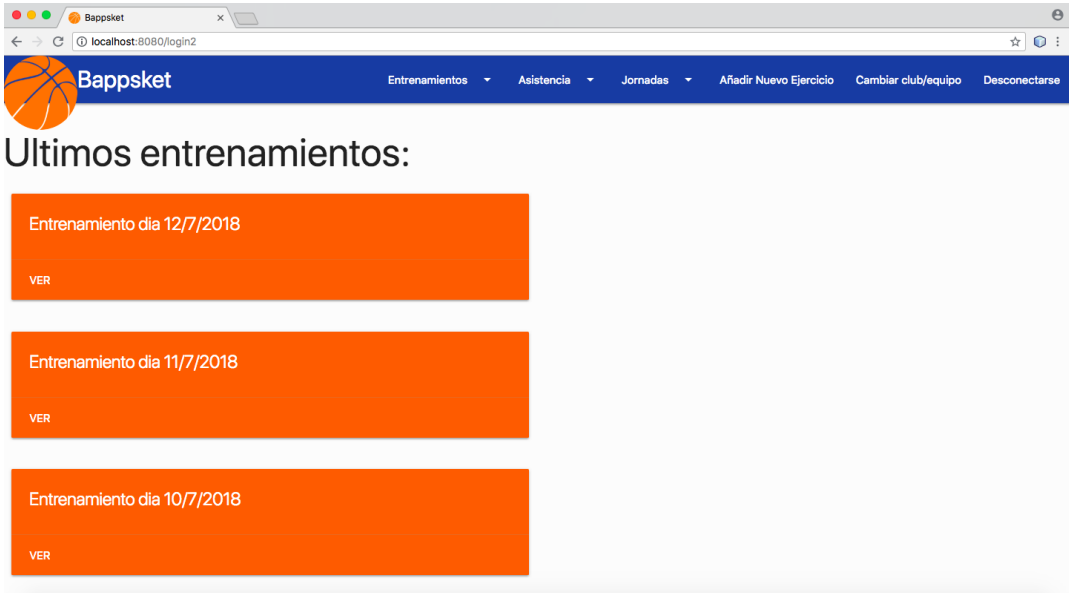


Figura C.4: Pantalla principal Entrenador

Si lo que queremos es registrarnos en la aplicación en la pantalla de ingreso tenemos una

opción que nos llevará a una pantalla para registrarnos.

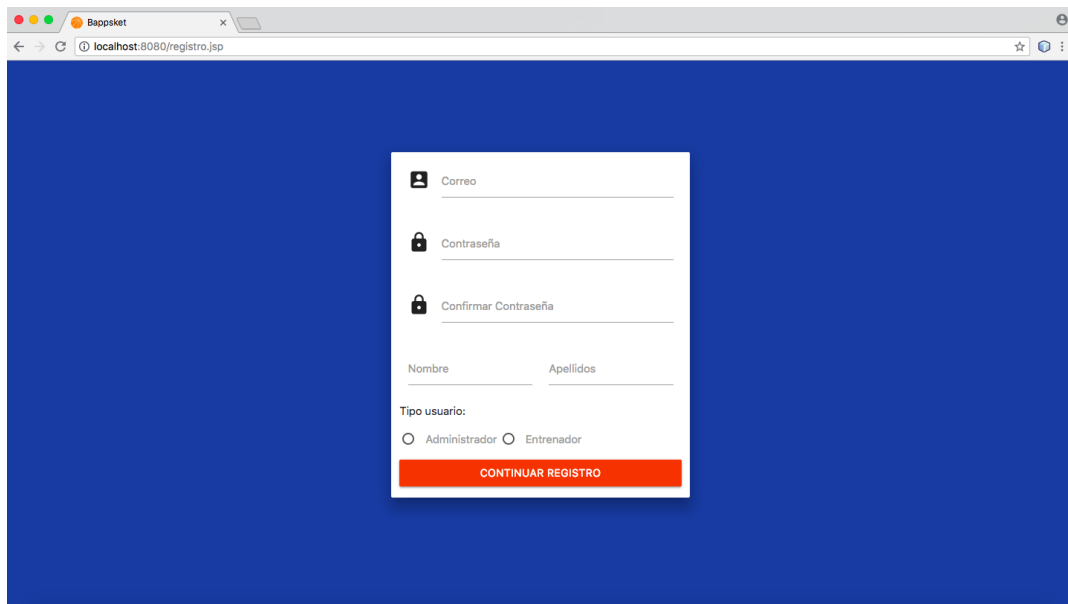


Figura C.5: Pantalla de registro

En esta pantalla tenemos que rellenar todos los datos que se nos piden. Si en tipo de usuario seleccionamos administrador se nos mostrará una pantalla donde seleccionaremos el nombre de nuestro club, si por el contrario elegimos entrenador se nos mostrará la pantalla principal.

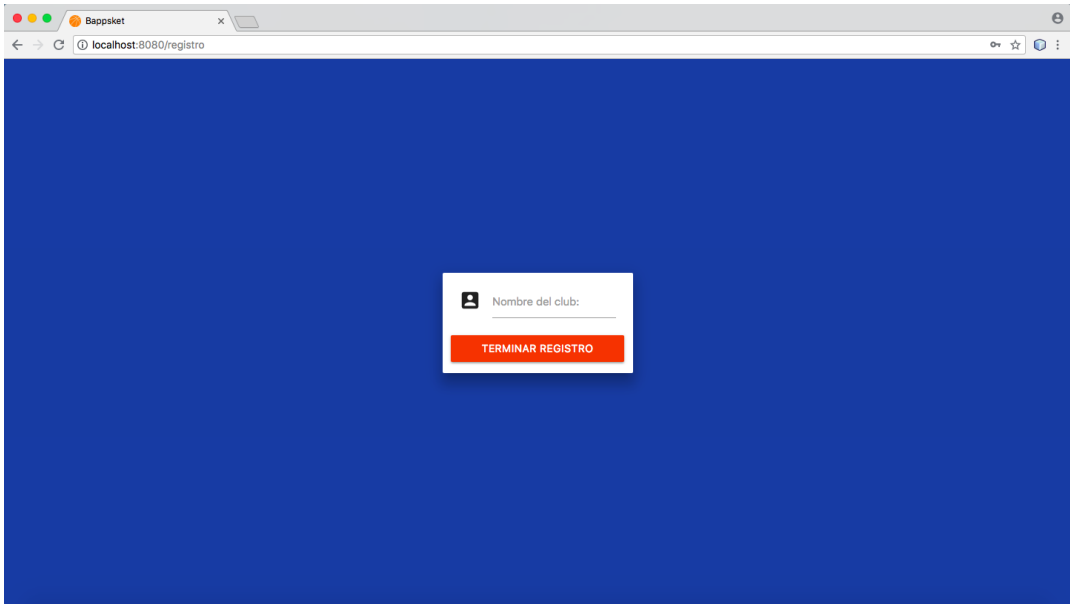
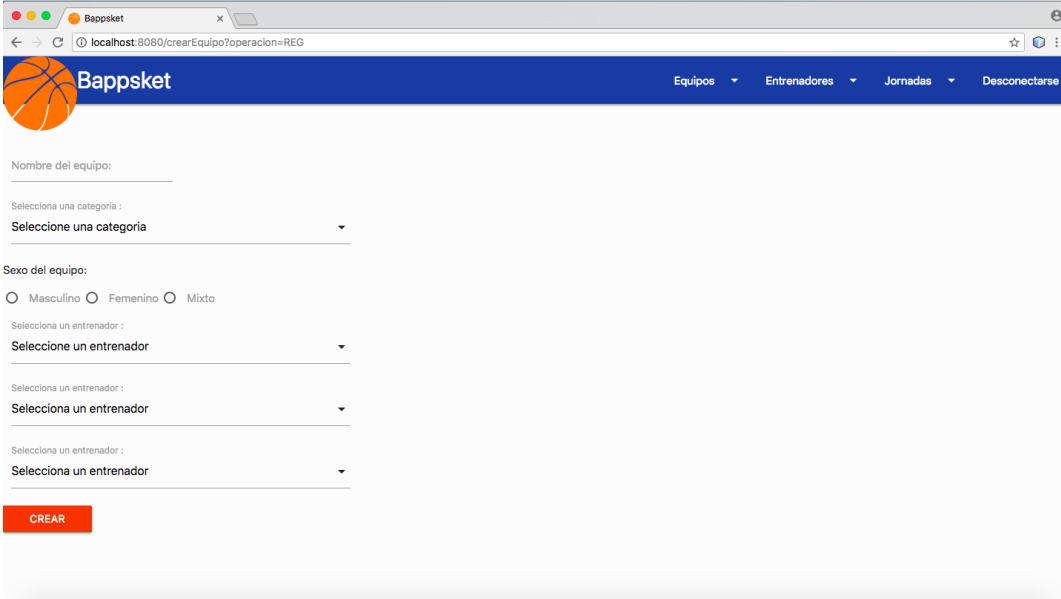


Figura C.6: Pantalla de añadir nombre al club

C.2. Registrar equipo y modificar un equipo

Esta opción es propia tanto de los administradores como de los entrenadores que no pertenecen a ningún club.

Al pinchar sobre la opción de registrar equipo se nos muestra una página con los datos básicos para crear un equipo, como son el nombre, la categoría, el sexo y los entrenadores de dicho equipo.



The screenshot shows a web browser window with the Bappsket logo and navigation menu. The URL is localhost:8080/crearEquipo?operacion=REG. The form includes fields for team name, category selection, gender selection (Masculino, Femenino, Mixto), and three coach selection dropdowns. A red 'CREAR' button is at the bottom.

Nombre del equipo: _____

Selecciona una categoría :
Seleccione una categoría ▼

Sexo del equipo:
 Masculino Femenino Mixto

Selecciona un entrenador :
Seleccione un entrenador ▼

Selecciona un entrenador :
Seleccione un entrenador ▼

Selecciona un entrenador :
Seleccione un entrenador ▼

CREAR

Figura C.7: Pantalla de crear equipo

En la siguiente pantalla se nos pedirá introducir los datos básicos de los jugadores que conforman el equipo.

Nombre	Apellidos	Dirección	Fecha de nacimiento
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Figura C.8: Pantalla de añadir jugadores

La opción de modificar nos muestra las mismas pantallas que las de registrar solo que esta vez con los datos del equipo cargados para poder modificarlos.

The screenshot shows a web browser window with the URL `localhost:8080/crearEquipo`. The page header features the Bappsket logo and navigation links: Equipos, Entrenadores, Jornadas, and Desconectarse. The main content area is a form for modifying a team. The form fields are as follows:

- Nombre del equipo: Benjamin A
- Selección de categoría: Benjamin 1er año
- Sexo del equipo: Masculino, Femenino, Mixto
- Selección de entrenador: Alberto García López
- Selección de entrenador: Selección un entrenador
- Selección de entrenador: Selección un entrenador

A red button labeled 'MODIFICAR' is located at the bottom of the form.

Figura C.9: Pantalla modificar equipo

The screenshot shows a web browser window with the URL `localhost:8080/crearEquipo2`. The page header features the Bappsket logo and navigation links: Equipos, Entrenadores, Jornadas, and Desconectarse. The main content area is titled 'Jugadores' and displays a table of player information.

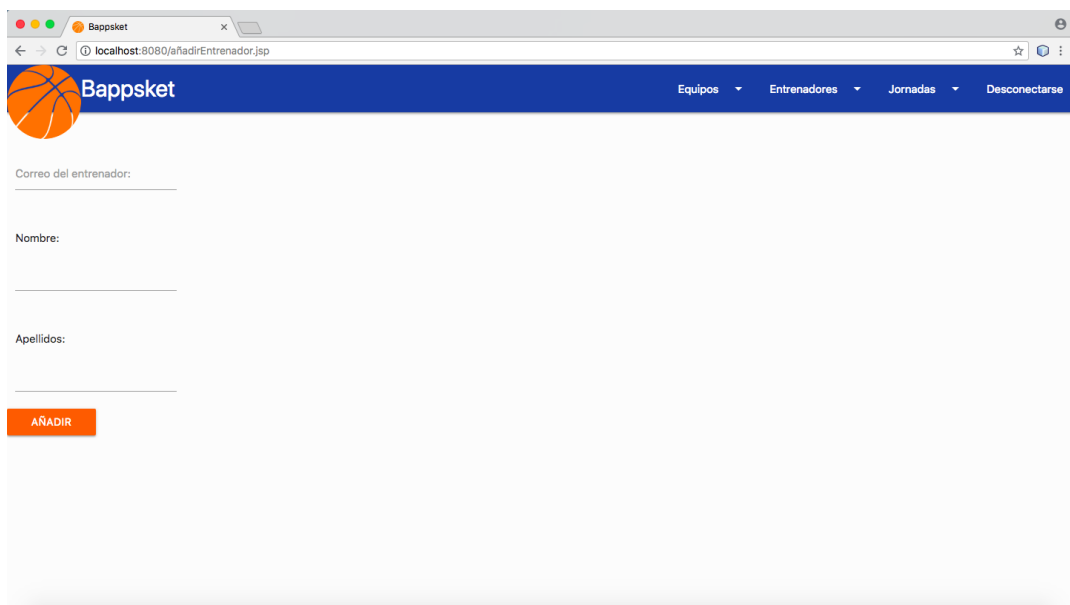
Nombre	Apellidos	Dirección	Fecha de nacimiento
Sergio	Perez Vaca	Avd Gijón	2000-01-01
Jose	Enjuto Herrero	Avd Palencia	2000-01-01
Ramon	Magdaleno Villasol	Avd Prado Boyal	2000-01-01
Alejandro	Roman Collazos	Calle Pensamiento	2000-01-01
Miguel	Fernandez Fernandez	Avd Madrid	2000-01-01

Figura C.10: Pantalla de modificar jugadores

C.3. Añadir y eliminar entrenadores del club.

Esta acción es solo propia de los administradores.

En la pantalla de añadir entrenador al club nos saldrá la opción para introducir el correo con el que el entrenador se registre en la aplicación. Si el entrenador existe se cargarán su nombre y sus apellidos para asegurarnos que es la persona que estamos buscando. Si intentamos dar de alta a un administrador o a un entrenador que ya esta dado de alta en nuestro club nos saltará una error avisándonos de ello.



The screenshot shows a web browser window with the URL `localhost:8080/añadirEntrenador.jsp`. The page header is blue and contains the Bappsket logo (an orange basketball) and the text 'Bappsket'. To the right of the logo are navigation links: 'Equipos', 'Entrenadores', 'Jornadas', and 'Desconectarse'. The main content area is white and contains a form with three input fields: 'Correo del entrenador:', 'Nombre:', and 'Apellidos:'. Below the fields is an orange button labeled 'AÑADIR'.

Figura C.11: Pantalla de añadir un entrenador

En la pantalla de eliminar un entrenador del club tendremos un desplegable donde nos aparecerán el nombre de todos los entrenadores que pertenecen al club. Simplemente tenemos que seleccionar el que queremos para eliminarlo.

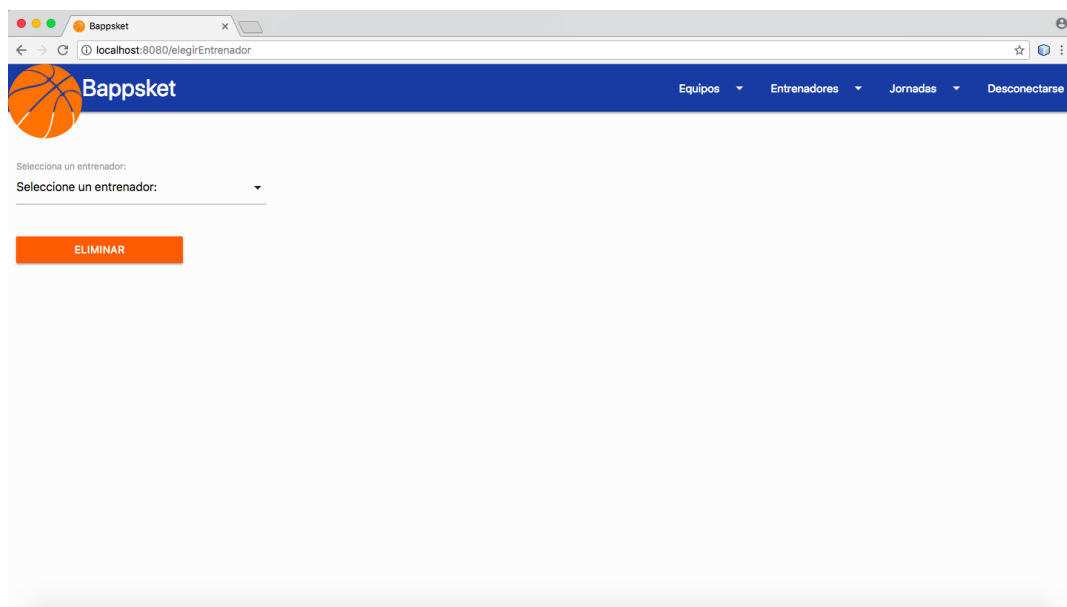


Figura C.12: Pantalla de eliminar un entrenador

C.4. Registrar, modificar y ver una jornada.

Esta acción es solo propia de los administradores, excepto la parte de ver jornada que es común con los entrenadores que tienen club.

En la pantalla de registrar tendremos que seleccionar la fecha en la que empieza nuestra jornada y nos encontraremos con tantos huecos como equipos haya en el club para apuntar los diferentes partidos de la jornada. Si por cualquier casual un equipo no juega esa jornada se puede dejar un hueco en blanco. A la hora de registrar la jornada los resultados no se podrán modificar.

Introduzca el primer día de la jornada

Equipo Local	Resultado	Equipo visitante	Fecha	Hora	Lugar
	-				
	-				
	-				

AÑADIR JORNADA

Figura C.13: Pantalla de crear una jornada

La pantalla de modificación es la misma que la de registro solo que esta vez tendremos la opción de modificar cualquier apartado de la jornada.

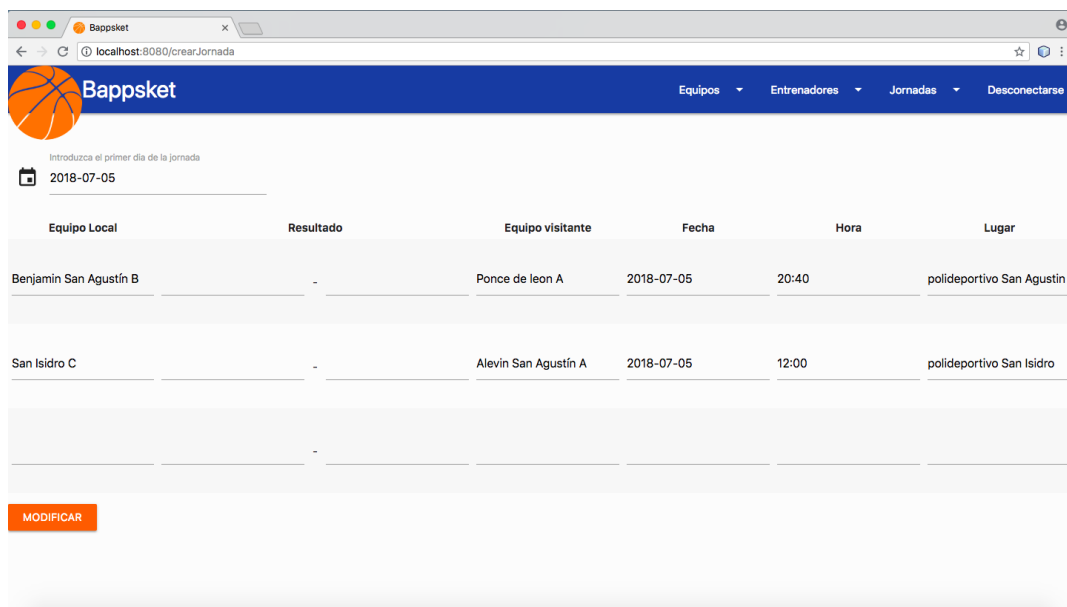
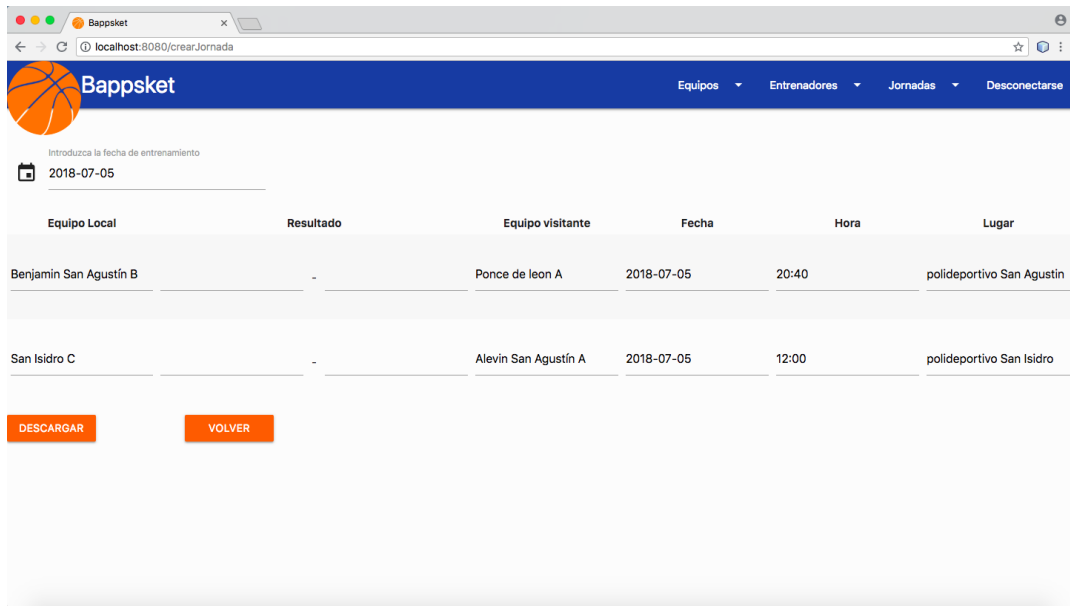


Figura C.14: Pantalla de modificar una jornada

Por último en la pantalla de ver jornada nos aparecerán los partidos de la jornada que seleccionemos y si somos administradores nos dará la opción de descargar los resultados de esa jornada.



The screenshot shows a web browser window with the URL `localhost:8080/crearJornada`. The page header includes the Bappsket logo and navigation links for `Equipos`, `Entrenadores`, `Jornadas`, and `Desconectarse`. Below the header, there is a date selection field set to `2018-07-05`. The main content area displays a table of training sessions:

Equipo Local	Resultado	Equipo visitante	Fecha	Hora	Lugar
Benjamin San Agustín B	-	Ponce de leon A	2018-07-05	20:40	polideportivo San Agustin
San Isidro C	-	Alevin San Agustín A	2018-07-05	12:00	polideportivo San Isidro

At the bottom of the table, there are two orange buttons: `DESCARGAR` and `VOLVER`.

Figura C.15: Pantalla de ver una jornada

C.5. Crear y modificar un entrenamiento

Esta acción es propia de los entrenadores, dando igual si tienen o no club.

En la pantalla de creación de entrenamiento se nos pedirá que introduzcamos una fecha para el entrenamiento. Solo podemos tener un entrenamiento cada día.

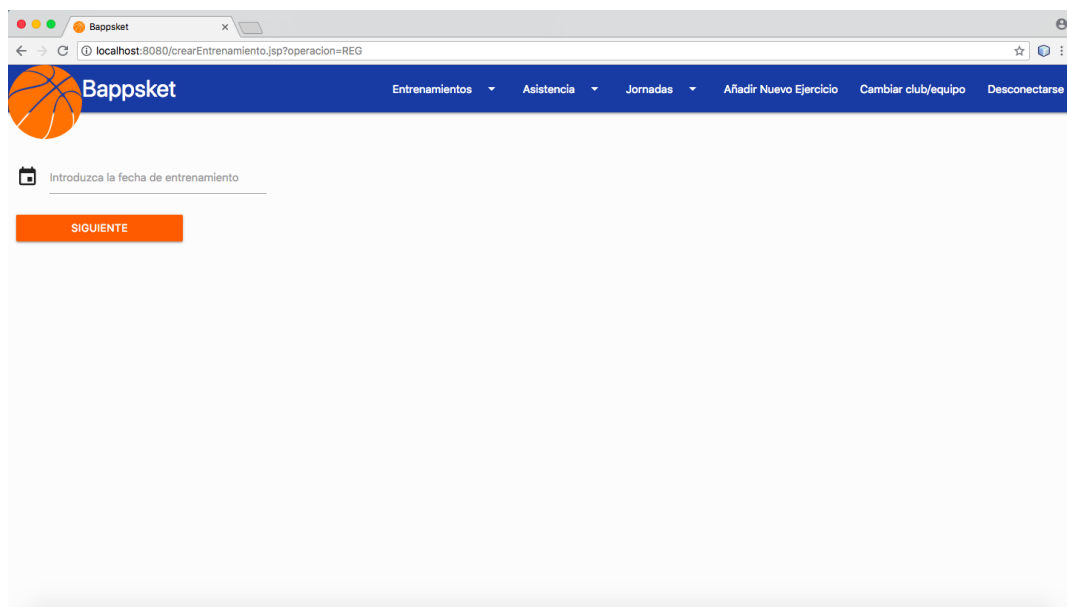


Figura C.16: Pantalla de elegir fecha para entrenamiento

Después de esto nos aparecerá un buscador de ejercicios según el concepto que entrenan, la categoría para la que están preparados y el tipo de ejercicio que sean. Si no seleccionamos ningún atributo y damos a buscar nos saldrán todos los ejercicios que haya.

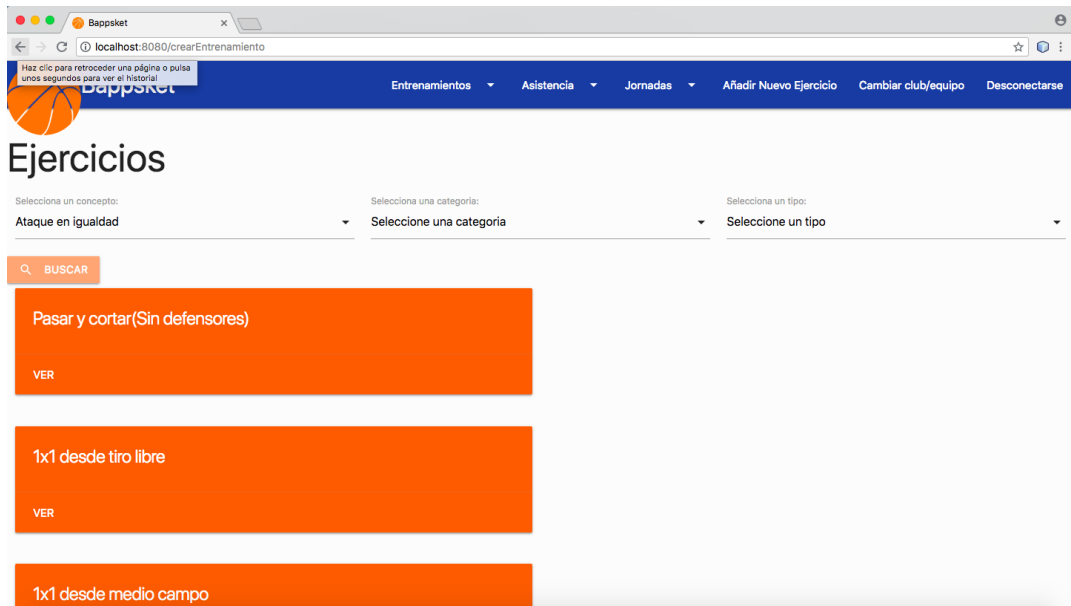


Figura C.17: Pantalla de elegir ejercicio

Una vez que seleccionamos el ejercicio se nos abrirá una pantalla con la descripción del ejercicio. Si queremos añadirlo al entrenamiento debemos introducir el tiempo que deseamos trabajar ese ejercicio.

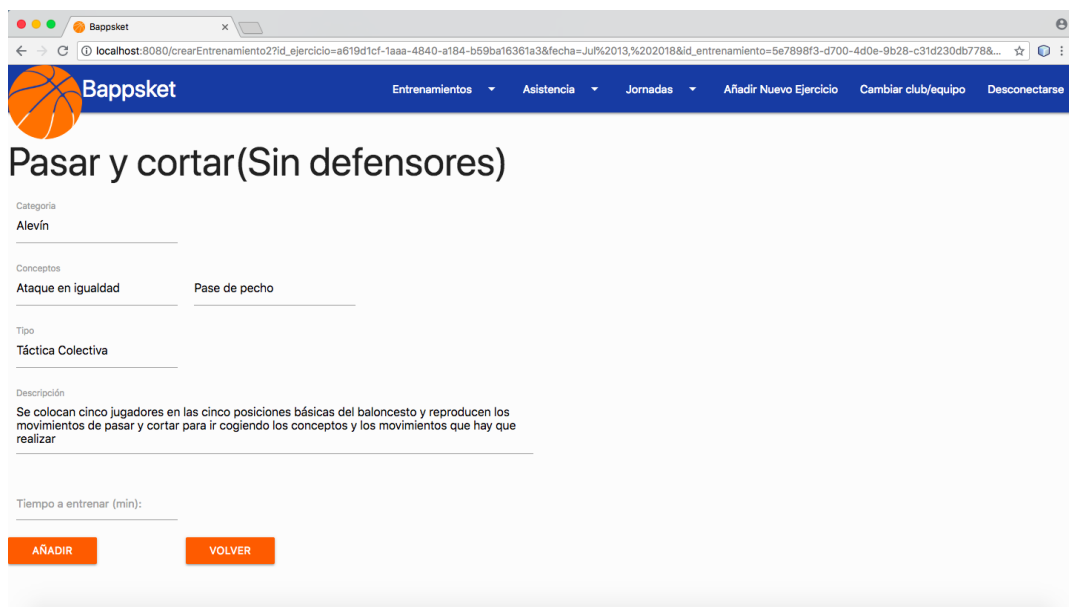


Figura C.18: Pantalla de descripción de ejercicio

Después de eso se nos mostrará una pantalla con todos los ejercicios que hay en el entrenamiento y las opciones de terminar el entrenamiento o de añadir mas ejercicios.

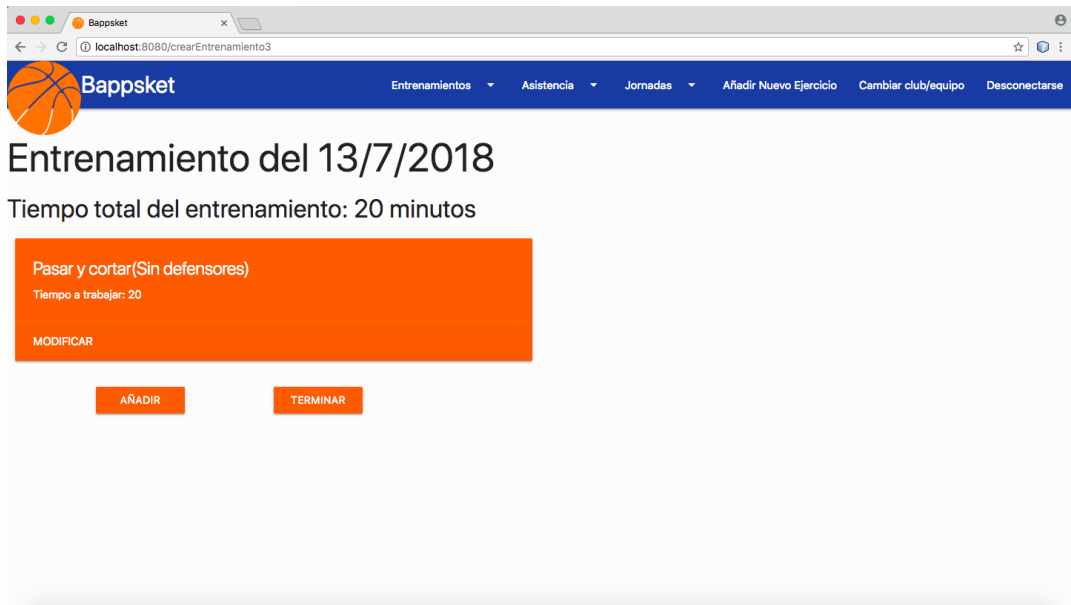


Figura C.19: Pantalla de resumen de entrenamiento

La pantalla de modificar el entrenamiento nos mostrará la lista de ejercicios que componen el entrenamiento, si queremos modificar el entrenamiento solo tendremos que seleccionar el ejercicio que queremos modificar y modificarlo.

C.6. Guardar y modificar la asistencia de los jugadores

Esta opción es propia de cualquier tipo de entrenador.

En la pantalla de guardado de asistencias nos pide que introduzcamos la fecha de la que queremos guardar la asistencia y a continuación nos muestra el nombre de cada jugador con un desplegable donde tenemos que elegir la asistencia de ese jugador.

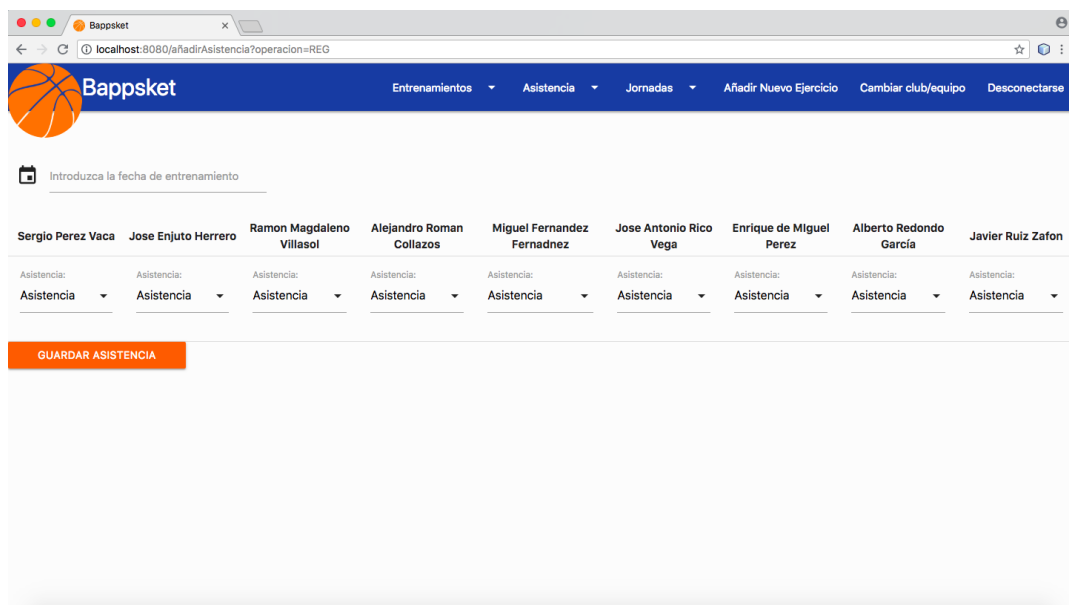


Figura C.20: Pantalla de añadir asistencia

La opción de modificación es igual solo que con los datos cargados de la fecha que seleccionemos.

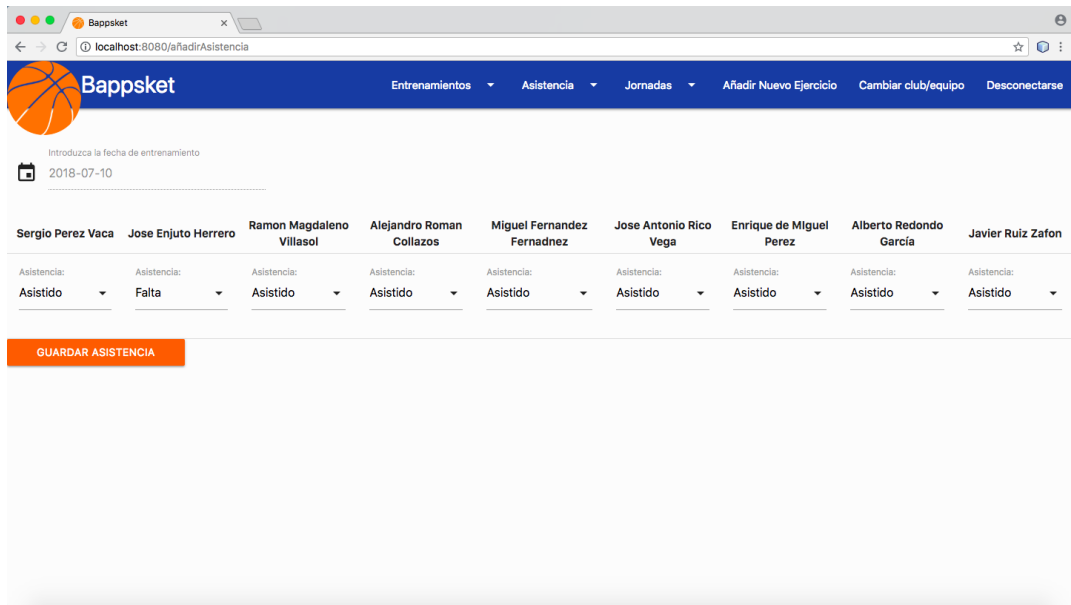


Figura C.21: Pantalla de modificar asistencia

C.7. Añadir resultado

Esta opción es propia solo de los entrenadores que tienen un club.

En la pantalla de añadir resultado se nos cargarán los partidos de la jornada que hayamos seleccionado. Ahí podremos modificar el resultado de nuestro partido. El resultado de los equipos es lo único que puede modificar el entrenador.

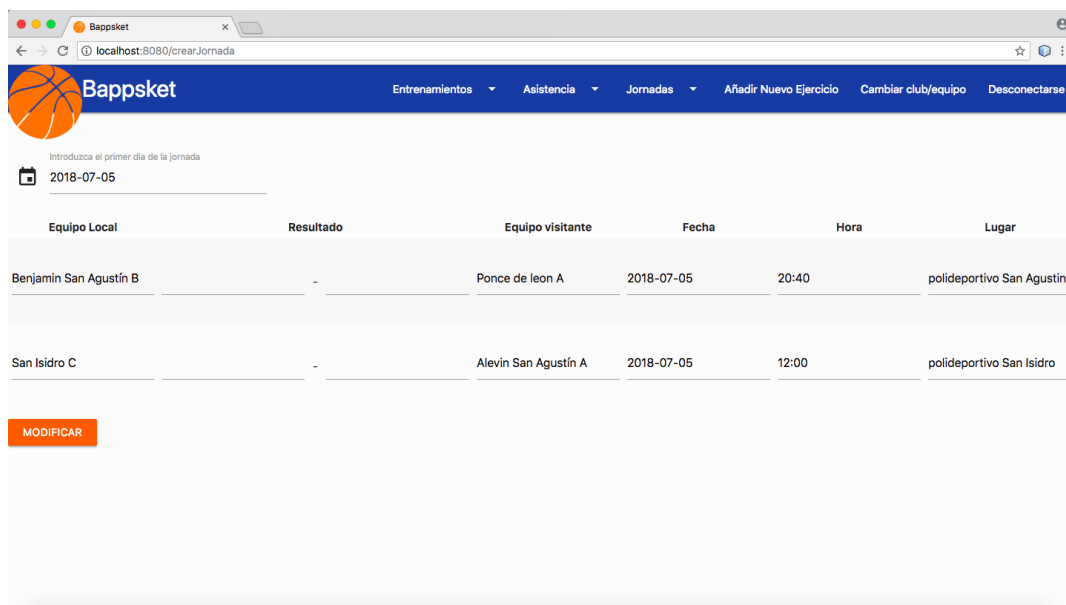
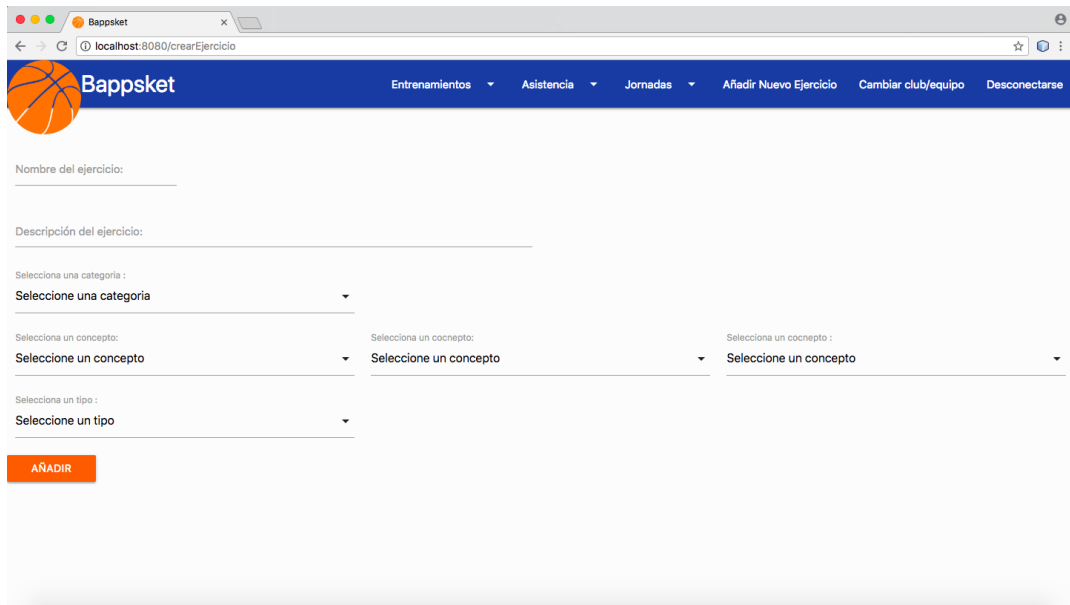


Figura C.22: Pantalla de añadir resultado

C.8. Añadir nuevo ejercicio

Esta opción es propia de cualquier tipo de entrenador.

En la pantalla de añadir un nuevo ejercicio nos aparecer un formulario con los datos a rellenar para poder crear un nuevo ejercicio en la base de datos. Los datos a rellenar son el nombre del ejercicio, la descripción del mismo, los conceptos que entrena (se da la opción de poner tres pero solo es obligatorio poner uno), la categoría para la que está preparado el ejercicio y el tipo de ejercicio que es.



The screenshot shows a web browser window with the URL `localhost:8080/crearEjercicio`. The page header features the Bappsket logo and a navigation menu with items: Entrenamientos, Asistencia, Jornadas, Añadir Nuevo Ejercicio, Cambiar club/equipo, and Desconectarse. The main content area contains a form for creating a new exercise with the following fields:

- Nombre del ejercicio:
- Descripción del ejercicio:
- Selección de categoría: **Seleccione una categoría** (dropdown menu)
- Selección de concepto: **Seleccione un concepto** (dropdown menu)
- Selección de concepto: **Seleccione un concepto** (dropdown menu)
- Selección de tipo: **Seleccione un tipo** (dropdown menu)

At the bottom of the form is an orange button labeled **AÑADIR**.

Figura C.23: Pantalla de añadir un nuevo ejercicio

C.9. Cambiar club/equipo

Esta opción es solo propia de los entrenadores, indistintamente de que tengan club o no.

En esta pantalla se nos abrirá un selector para cambiar entre los clubs y equipos de los que forma parte el entrenador.

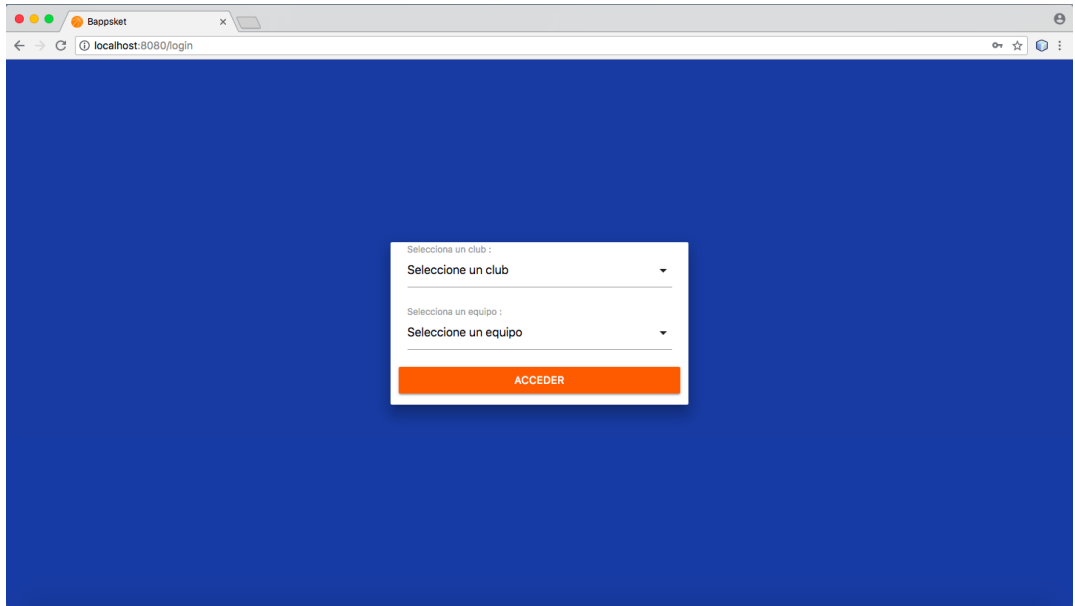


Figura C.24: Pantalla de cambiar club/equipo