Universitat Oberta de Catalunya

Desarrollo de una aplicación web. **Futbol Pro Manager**.

Bernardo Fernández López
Grado de Ingeniería Informática

Consultor: Gregorio Robles Martínez

06/2021

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons
autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	Futbol Pro Manager
Nombre del autor:	Bernardo Fernández López
Nombre del consultor:	Gregorio Robles Martínez
Fecha de entrega (mm/aaaa):	06/2021
Área del Trabajo Final:	Desarrollo web
Titulación:	*Grado de Ingeniería Informática*

Resumen del Trabajo (máximo 250 palabras):

Futbol Pro Manager surge de la unión de aquellos conocimientos adquiridos a lo largo del grado y de varias de las aficiones personales como son; los videojuegos y el fútbol. Así pues, el objetivo del trabajo será facilitar una nueva herramienta con fines recreativos a aquellos usuarios que disfruten también de estos 'hobbies' y desean complementar el visionado común de partidos de fines de semana con una nueva motivación, o desde otra perspectiva.

Así pues, la aplicación propone a los usuarios gestionar de forma virtual una plantilla de fútbol con jugadores pertenecientes a la 1ª división española. El objetivo es conformar una plantilla competitiva jornada tras jornada con el fin de superar a tus rivales en base al desempeño de cada jugador de la liga en sus partidos oficiales. Es decir, se añade todavía más emoción a esos fines de semana futboleros. Para ello, se facilitará la posibilidad de inscribirse en una comunidad existente donde competir, o bien, crear una propia y privada con tus amigos donde pasarás a elegir la táctica más adecuada en base a los jugadores de los qué dispones, entre otros factores. Además del reparto aleatorio de jugadores que obtendrá el usuario inicialmente, se dispondrá de un mercado de fichajes donde pondrás a prueba tu capacidad de ojeo para hacerte con buenas gangas.

En definitiva, se busca una experiencia que combinará la estrategia y el azar, pero especialmente tratará de añadir algo de ocio y competición a tu día a día.

Abstract (in English, 250 words or less):

Futbol Pro Manager arises from the union of the knowledge acquired throughout the degree and several of the personal hobbies such as; video games and football. Thus, the aim of the work will be to provide a new tool for recreational purposes to those users who also enjoy these hobbies and wish to complement the common viewing of weekend matches with a new motivation, or from another perspective.

Thus, the application proposes to users to virtually manage a football squad with players belonging to the Spanish 1st division. The aim is to build a competitive squad day after day in order to beat your rivals based on the performance of each player in the league in their official matches. In other words, even more excitement will be added to those football weekends. To do this, you will be able to join an existing community where you can compete, or create your own private community with your friends where you will choose the most appropriate tactics based on the players you have available, among other factors. In addition to the random distribution of players that the user will initially get, there will be a transfer market where you will test your scouting skills to get good bargains.

In short, we are looking for an experience that will combine strategy and randomness, but will especially try to add some leisure and competition to your day-to-day life.

Palabras clave (entre 4 y 8):

Aplicación, desarrollo, web, futbol, videojuegos, fantasy, manager, estrategia.

Índice

1. Introducción 1
1.1 Contexto y justificación del Trabajo 1
1.2 Objetivos del Trabajo 1
1.3 Enfoque y método seguido 3
1.4 Planificación del Trabajo 4
1.5 Breve sumario de productos obtenidos 5
1.6 Breve descripción de los otros capítulos de la memoria 6
2. Tecnologías utilizadas 7
3. Selección del proyecto 11
4. Evaluación de riesgos 12
5. Primer acercamiento al campus 13
6. Análisis de requisitos 14
7. Definición de la base de datos 15
7.1 Tabla Comunidad. 16
7.2 Tabla jugador. 17
7.3 Tabla usuario. 17
7.4 Tabla alineaciones. 18
7.5 Tabla com_jug_usu. 19
7.6 Tabla club. 20
7.7 Tabla mercado. 20
7.8 Tabla pujas. 21
7.9 Tabla saldoguardado. 21
7.10 Relaciones entre las tablas. 21
8. Diseño de la interfaz 24
9. Implementación 35
9.1 Página de inicio. 35
9.2 Página alineación. 41
9.3 Página clasificación. 48
9.4 Página normas. 50
9.5 Página mercado. 54
9.6 Página estadísticas. 61
9.7 Página contacto. 65
9.8 Página panel de administración. 67
10. Despliegue de la aplicación 71
11. Conclusiones 79
12. Glosario 81
13. Bibliografía 82

Lista de figuras

Ilustración 1. Diagrama de Gantt con la planificación del proyecto. 5
Ilustración 2. Modelo entidad-relación de la base de datos Futbol Pro Manager. 16
Ilustración 3. Logo de la aplicación FPM diseñado con Photoshop. 24
Ilustración 4. Banner de la aplicación FPM diseñado con Photoshop. 24
Ilustración 5. Boceto página de inicio. 25
Ilustración 6. Boceto página de alineación. 26
Ilustración 7. Boceto página de clasificación. 27
Ilustración 8. Boceto página de normas. 28
Ilustración 9. Boceto página de mercado. 29
Ilustración 10. Boceto de la sección para consultar las ofertas realizadas y recibidas. 30
Ilustración 11. Boceto de la sección para poner a la venta jugadores. 31
Ilustración 12. Boceto para la página de estadísticas. 32
Ilustración 13. Boceto para la página de contacto. 33
Ilustración 14. Boceto para la página de administración. 34
Ilustración 15. Vista de la página de inicio de la aplicación Futbol Pro Manager. 35
Ilustración 16. Barra para las acciones relativas a la identificación del usuario. 35
Ilustración 17. Barra de identificación muestra error en la información introducida. 36
Ilustración 18. Barra de identificación muestra datos del usuario conectado. 36
Ilustración 19. Panel de creación de cuenta de usuario. 36
Ilustración 20. Fallo de registro por contraseñas. 37
Ilustración 21. Fallo de registro por campos vacíos. 37
Ilustración 22. Opciones relativas a las comunidades. 37
Ilustración 23. Configuración de la comunidad. 38
Ilustración 24. Pantalla de error de configuración. 38
Ilustración 25. Pantalla selección de comunidad. 38
Ilustración 26. Solicitud de clave para acceder a una comunidad. 38
Ilustración 27. Página de alineación y gestión de la plantilla. 41
Ilustración 28. Panel informativo del usuario. 41
Ilustración 29. Vista del panel lateral izquierdo con la plantilla del usuario. 42
Ilustración 30. Vista de la alineación elegida por un usuario. 43
Ilustración 31. Ejemplo de desplegable para elegir defensa. 43
Ilustración 32. Plantilla con destacado de jugadores alineados en fondo azul y negrita. 44
Ilustración 33. Panel de selección de táctica. 45
Ilustración 34. Estado de la alineación en el momento que se cambia de táctica. 45
Ilustración 35. Botones de guardado y reinicio de la alineación. 46
Ilustración 36. Página de clasificación. 48
Ilustración 37. Tabla con la clasificación general en dicha comunidad. 49
Ilustración 38. Tabla con la clasificación de la última jornada en una comunidad con tres miembros. 49
Ilustración 39. Widget externo con los resultados de la última jornada. 49
Ilustración 40. Página de normas del juego. 50
Ilustración 41. Sección normas "cómo jugar". 51
Ilustración 42. Sección normas "Normas Generales". 51
Ilustración 43. Sección normas "Puntuaciones". 52
Ilustración 44. Sección normas “mercado”. 52
Ilustración 45. Sección normas “Alineación”. 53
Ilustración 46. Vista de la página de mercado para un usuario en una comunidad. 54
Ilustración 47. Vista de los datos relevantes de los jugadores activos en el mercado de fichajes. 55

Ilustración 48. Vista de la botonera donde se ubica la función 'Ofrecer Jugador'. 56
Ilustración 49. Vista de un usuario concreto para poner a la venta jugadores. 56
Ilustración 50. Vista de la botonera donde se ubica la función 'Quitar jugador del mercado'. 57
Ilustración 51. Vista del listado de jugadores puestos a la venta por un usuario. 57
Ilustración 52. Vista de la botonera donde se ubica la función 'Ver Ofertas'. 57
Ilustración 53. Ejemplo de vista del panel de negociaciones. 58
Ilustración 54. Vista de la botonera donde se ubica la función 'Pujar'. 59
Ilustración 55. Mensaje informativo en rojo al insertar una cifra inferior al precio mínimo. 59
Ilustración 56. Vista desde la cual se puede retirar un jugador del mercado de fichajes. 60
Ilustración 57. Página para consultar datos destacables de todos los jugadores. 61
Ilustración 58. Vista de los porteros con más puntos. 62
Ilustración 59. Vista de todos los porteros con orden descendente según la media. 63
Ilustración 60. Página para contactar con el administrador del sitio web. 65
Ilustración 61. Formulario de contacto. 65
Ilustración 62. Breve fragmento del uso de la función 'mail' de php. 66
Ilustración 63. Vista del panel de administración/desarrolladores. 67
Ilustración 64. Paneles desplegables para asignar puntuaciones a jugadores. 69
Ilustración 65. Logo Hostinet. 72
Ilustración 66. Vista de los productos contratados en hostinet. Hosting + dominio. 72
Ilustración 67. Información del dominio relativa a las dns. 73
Ilustración 68. Mensaje informativo representativo de conexión no segura. 73
Ilustración 69. Mensaje de conexión segura obtenido tras instalar el certificado ssl. 73
Ilustración 70. Favicon de la aplicación en una pestaña de google chrome. 74
Ilustración 71. Nombre de la base de datos y usuario con todos los privilegios. 74
Ilustración 72. Base de datos cargada en hostinet. 74
Ilustración 73. Configuración de sitio y del servidor en dreamweaver. 75
Ilustración 74. Modificación de los datos de conexión. 75
Ilustración 75. Widget en local. 76
Ilustración 76. Widget en el servidor web. 76
Ilustración 77. Formato de correo tras envío desde el formulario de contacto en la aplicación. 77
Ilustración 78. Listado de tareas cron que aplican automatismo a la aplicación. 77
Ilustración 79. Configuración que fuerza la redirección hacia HTTPS en todas las secciones de la web. 77

1. Introducción

1.1 Contexto y justificación del Trabajo

El punto de partida del trabajo encuentra sus raíces a la hora de lograr la satisfacción de aquellos grupos de personas aficionadas al deporte (entre los que se incluye quien redacta la memoria), más específicamente el fútbol y que también le gustan los videojuegos. Y es que ambas aficiones tienen detrás un séquito bastante abultado. De hecho, es fácil percatarse que el deporte más seguido es el fútbol[1] a través de las diferentes vías (televisión, dispositivos móviles, radio analógica, etc.). Por su parte, los videojuegos tienen un origen más reciente, sin embargo, han adquirido una presencia importante, llegando a alcanzar los 15 millones de jugadores según los datos extraídos de la Asociación Española de Videojuegos, con datos de 2019[2]. Así pues, es fácil advertir la relevancia de ambos sectores a nivel nacional y su repercusión inevitable en la sociedad.

La propuesta que se realiza mediante este trabajo final de grado no es exclusiva, es decir, ya se pueden encontrar en el mercado distintas facilidades que unen estos dos conceptos de la mano de páginas como https://www.comunio.es/, o bien, https://www.laligafantasymarca.com/. Evidentemente, estas aplicaciones se han trabajado sin restricciones temporales y probablemente con un grupo de personas estructurado, lo que ha permitido definir objetivos más trabajados y, en definitiva, un alcance más profundo. Así pues, queda patente que el objetivo de este proyecto no es competir dentro del sector, sino ser capaz de demostrar las capacidades adquiridas en el grado, pero plasmándolas en un entorno amigable para el desarrollador al unificar dos de sus pasiones. ¿Quién sabe si en un futuro se puede llegar a profundizar lo suficiente en la idea?

1.2 Objetivos del Trabajo

La realización de una aplicación comporta una definición previa de objetivos y subobjetivos que permiten llevarla a cabo. A continuación, se mencionan en forma de listado:

1. **Selección de proyecto.**
   - Identificación de las diferentes áreas que propone la UOC.
   - Análisis en base a mis capacidades.
   - Posibilidades de aplicación dentro del área más afín.

2. **Plan de trabajo (PEC 1).**
   - Establecer contacto con el foro y profesor asociado a la asignatura.
   - Planteamiento de la idea.
   - Negociación con el profesor asociado.
   - Producción del entregable PEC1.

- Espacio para modificaciones/cambios post-feedback.
- Producción definitiva del entregable PEC1.

3. **Análisis de los requisitos (PEC 2).**
   - Determinar el alcance estimado.
   - Confección de un entorno de trabajo.
   - Selección de las herramientas para producirlo.

4. **Definición de la base de datos (PEC 2).**
   - Analizar el propósito de la misma.
   - Estructuración (crear tablas y relaciones).
   - Recopilación de datos útiles.
   - Disposición de los datos en tablas.
   - Revisión de restricciones y relaciones.

5. **Diseño de la interfaz (PEC 2).**
   - Recopilación de ideas.
   - Bocetaje de la aplicación.
   - Implementación (en constante revisión y paralelamente).

6. **Implementación (PEC 3).**
   - **Página Inicial.**
     - Sistema de login/logout/registro.
     - Sistema de creación de comunidad (pública o privada).
     - Generación equilibrada de plantilla post-registro.
     - Asignación de saldo y plantilla al usuario.
     - Tests.
   - **Página Alineación.**
     - Carga de datos del perfil del usuario y comunidad.
     - Carga de datos de la plantilla.
     - Visualización de alineación activa.
     - Indicador de jugadores alineados la última jornada.
     - Selección de táctica.
     - Selección de jugadores por posición.
     - Sistema de guardado/reinicio de alineación.
     - Tests
   - **Página Clasificación.**
     - Carga de clasificación general.
     - Carga de clasificación en la última jornada disputada.
     - Carga de resultados de los partidos en vivo.
   - **Página Normas.**
     - Planteamiento y redacción de las reglas.
     - Paso de información a la web.

   - **Página Mercado.**
     - Generación aleatoria de jugadores libres.
     - Carga de dicha información en cada comunidad.
     - Función para añadir jugadores al mercado.
     - Función para vender directamente jugadores.
     - Función para quitar jugadores del mercado.

    - Posibilidad de consultar ofertas recibidas y realizadas.
    - Función para realizar ofertas.
    - Validaciones para el campo donde se realizan ofertas.
    - Función para retirar ofertas realizadas.
    - Testeo.
  - **Página Estadísticas.**
    - Cargar mejores jugadores por posición.
    - Carga de otras estadísticas interesantes.
  - **Página Contacto.**
    - Creación de formulario.
    - Validación de campos.
    - Configuración para la recepción por correo.
  - **Página Administración.**
    - Configuración de acceso a perfiles concretos.
    - Función para aceptar las pujas realizadas en todos los mercados
    - Función para actualizar el mercado (automatizable).
    - Función para guardar las alineaciones definitivas de la jornada.
    - Función para reiniciar saldo de usuario.
    - Panel para otorgar puntuaciones a los jugadores.
    - Desarrollo de funcionalidades no contempladas.

7. **Despliegue de la aplicación.**
   - Análisis de mercado.
   - Selección de opción adecuada.
   - Migración de proyecto de local al servidor.
   - Adaptación de funcionalidades que requieren automatización por horas.
   - Testeo y corrección de problemáticas derivadas.

8. **Memorias.**
   - Recopilación de los entregables de las PEC.
   - Recopilación de la información derivada de los objetivos y tareas fijadas.
   - Proyección del desarrollo en la estructura de la memoria.

9. **Presentación.**
   - Selección de herramientas para presentaciones.
   - Selección de ideas clave del proyecto.
   - Descripción de las ideas adecuada al tiempo de presentación.
   - Simulación de la presentación.
   - Correcciones y entrega.
   - Respuestas al tribunal.

1.3 Enfoque y método seguido

La experiencia personal en el ámbito laboral ha permitido desarrollar el proyecto bajo una mecánica cuyo núcleo radica en la metodología Scrum, dentro de las metodologías ‘agile’. Sin embargo, este adquiere

una dimensión diferente si se tiene en cuenta que los mayores beneficios de esta metodología se pueden lograr mediante la conformación de un grupo de trabajo que se retroalimente y en esta ocasión se trabaja de forma individual.

Pese a todo, el hecho de desarrollar una idea que no era especialmente innovadora, puesto que es posible localizar en el mercado ideas similares, ha permitido definir claramente los objetivos a alcanzar (mencionados en el apartado anterior). En este punto, es sencillo darse cuenta del paralelismo que surge entre estos objetivos/subobjetivos y los denominados 'sprint' de Scrum[3]. A excepción de unos pocos, la mayoría de los objetivos se pueden desarrollar de forma independiente (no hay un alto grado de interdependencia entre estos), permitiendo la posibilidad de reiterar sobre ellos para pulir los detalles en cualquier momento y sin que un error produzca males irreparables.

Esta serie de objetivos o 'sprints' nacen de forma orgánica a la hora de iniciar el trabajo. Es necesario analizar los requisitos antes de emprender, de la misma forma que será necesaria una correcta definición de las herramientas con las que se pretende trabajar. Una vez logrado esto, el desarrollo presenta una serie de secciones claramente diferenciadas que se podían trabajar individualmente. Adicionalmente, la experiencia personal ha permitido estipular una temporalidad bastante precisa para alcanzar el cumplimiento de cada uno de los hitos fijados.

En definitiva, el hecho de que la idea estuviera claramente definida ha facilitado la creación de una planificación certera, cuyos objetivos se ceñían a la realidad del contexto (académico y temporal) y finalmente la capacidad de adaptarse a esta metodología ágil que brilla principalmente por su capacidad de adaptación.

1.4 Planificación del Trabajo

El primer paso ha sido estipular las tecnologías que a priori se antojan necesarias para materializar el desarrollo y cuyo núcleo se basa en Javascript (consultas Ajax y jquery) en la parte de cliente y php para el lado del servidor. Con estas se han trabajado las funcionalidades más elaboradas del proyecto. Por su parte, se ha utilizado Mysql para trabajar con las consultas a las bases de datos y Css para configurar el diseño de la web. Es importante destacar que parte de las herramientas que aquí se mencionan forman parte de la suite Xampp que se ha utilizado para realizar el desarrollo en local.

Respecto al entorno de trabajo se ha optado por la herramienta Adobe Dreamweaver para implementar el desarrollo por la cantidad de facilidades que aporta al desarrollador como es la posibilidad de mover archivos de local al servidor y su posterior testeo, mientras que los navegadores Google Chrome y Mozilla Firefox, junto con sus herramientas para desarrollo, han sido las herramientas para realizar las comprobaciones.

Por último, se considera reseñable la mención de dos herramientas con fines más específicos. Por una parte, la web Monday.com ha permitido implementar un diagrama de Gantt cuyo diseño es bastante amigable. Por otra parte, la herramienta online MockFlow ha permitido desarrollar bocetos para las distintas secciones de la web. En la sección '2.Tecnologías utilizadas' (en este mismo documento), se desglosarán con más detalle tanto las ya mencionadas, como otras de menor alcance.

El segundo punto tratado en la planificación es la capacidad de localizar los objetivos que deberán superarse con el fin de obtener el producto deseado y acotarlos a un rango de fechas que permita cumplir con los plazos acordados. De esta necesidad surge un diagrama de Gantt que, por motivos de adaptabilidad, no es posible insertar directamente en este documento. Sin embargo, se aportan los siguientes enlaces con la intención de facilitar el visionado en una calidad óptima:

Opción 1. Google Drive (permisos de acceso como lector a todo miembro de la UOC):

https://drive.google.com/file/d/1Wk6lmi98nQsv2z4fHbwKQ66OLXTJg3Y3/view?usp=sharing

Opción 2. Mediante la web imgbb.com (acceso limitado a la disponibilidad de la imagen en dicho servidor):

https://ibb.co/WvrCvZx

Ilustración 1. Diagrama de Gantt con la planificación del proyecto.

1.5 Breve sumario de productos obtenidos

La realización del proyecto solamente obtendrá como resultado un producto, que en este caso no es otro que la aplicación web Futbol Pro Manager, cuyas funcionalidades más destacadas (básicas para su funcionamiento) han sido las siguientes:

- Identificación y registro del usuario.
- Creación de una comunidad (espacio de juego).
- Gestiones 'manager' (alinear, elegir sistema táctico, etc.).
- Gestiones económicas en el mercado (fichar, vender, especular, etc.).
- Consultar estadísticas de jugadores.
- Revisar clasificaciones.
- Secciones de las reglas del juego y términos y condiciones.
- Sección de contacto con el creador del juego.
- Panel administrativo (para el desarrollador).

1.6 Breve descripción de los otros capítulos de la memoria

Los siguientes capítulos de la memoria pretenden describir con detalle cada uno de los objetivos y subobjetivos fijados en el diagrama presentado en la ilustración 1, es decir, se estarán describiendo cada uno de los pasos realizados en el presente proyecto, o lo que es lo mismo, el camino que se ha seguido para llegar al producto entregable finalmente.

Adicionalmente, se desglosarán también las tecnologías utilizadas, con su correspondiente justificación, mientras que, por otra parte, se mostrará una pequeña sección relativa a la evaluación de los riesgos, es decir, aquellas circunstancias que se han tenido en cuenta desde un inicio con el fin de mitigar las consecuencias derivadas de ciertas problemáticas.

2. Tecnologías utilizadas

Javascript.[4]

Se trata de un lenguaje de programación pensado para ser interpretado por los navegadores (clientes) de forma nativa, evitando de esta manera la necesidad de un compilador. Junto al lenguaje html y css, conforman el tridente básico nativo para la programación web en la parte cliente.

Javascript permite añadir dinamismo a los elementos establecidos en la estructuración de nuestra web. Para lograrlo, toma como base las posibles interacciones que un usuario podría requerir durante la navegación web y prepara eventos que se podrían producir, como puede ser; realizar click sobre cualquier elemento, utilizar el scroll del ratón, o bien, mantener, pulsar o levantar una tecla específica del teclado. En definitiva, añade funcionalidades a la estructura principalmente estática que aporta html.

Ajax.[5]

Ajax no es un lenguaje de programación, sino un conjunto de prácticas basadas en un lenguaje, como bien puede ser javascript, que permiten la comunicación con los datos (normalmente almacenados en un servidor) sin la necesidad de forzar una recarga completa de la página en la que se ubica el usuario, es decir, permite realizar actualizaciones sobre fragmentos de la web.

Su aparición surge de la necesidad de optimizar los tiempos de respuesta del servidor. Para ello, incorpora una serie de procesos en segundo plano que permiten la interacción con los datos. Datos, que finalmente se procesarán con el dinamismo que añade javascript.

jQuery.[6]

jQuery es una librería dentro del enorme espectro que presenta javascript cuya finalidad reside en la idea de simplificar el código que suele implementarse con javascript puro. En definitiva, permite desarrollar funcionalidades con mayor sencillez y también de una forma más rápida.

Si bien es cierto que cualquier desarrollo se podrá implementar con las dos alternativas, en determinadas ocasiones el uso de esta librería puede evitarte quebraderos de cabeza al reducir código extenso a una o dos líneas. En definitiva, cuantas menos líneas de código haya, menos posibilidades de incurrir en errores y de localizarlos.

Php.[7]

Es uno de los lenguajes de programación web más utilizado en el lado del servidor. Se trata de un lenguaje interpretado, por lo que su código es procesado por un servidor (previamente al envío de la página al cliente) que retorna un resultado dinámico a las páginas.

Uno de sus mayores beneficios es que permite la incrustación con html. Además, presenta un fácil acceso a las bases de datos, con quien se comunica frecuentemente para generar resultados dinámicos personalizados. El hecho de estar incorporado en la suite 'xampp' es uno de los motivos que ha llevado a seleccionarlo como lenguaje del lado de servidor.

Mysql.[8]

Se trata de un sistema gestor de bases de datos de código abierto y como tal, aporta una serie de facilidades para la comunicación web/servidor. Entre sus funcionalidades más específicas se pueden destacar la creación y administración de bases de datos relacionales.

Así pues, será sencillo interconectar tablas de datos que den sentido al "ecosistema" de datos relacionados con el contenido de una determinada página web. En definitiva, es un almacén de datos que permite adaptar el contenido en cualquier momento, ya sea una modificación concreta de un valor, o bien, un cambio en la estructura completa de la base de datos.

Css.[9]

Css (hojas de estilo en cascada) es un lenguaje de programación relacionado con la apariencia del sitio web. Concretamente, se encarga de aplicar estilos a los elementos definidos con html, ya sea coloreándolos, o cambiando su tamaño.

Su nombre viene dado por la implementación en cascada, es decir, la aplicación de arriba a bajo a partir de una serie de prioridades derivadas de la herencia, es decir, en aquellos momentos donde un elemento tenga asociados distintos estilos, provocando ambigüedad, se aplican una serie de estándares para resolverlo.

Una de las cosas más importantes a tener en cuenta con este lenguaje es que se ciñe a unos estándares que deberán seguir los navegadores para que la interpretación que realiza cada uno de ellos sea coherente, es decir, si se colorea un elemento de color rojo, este será rojo en cada navegador.

Adobe Dreamweaver.[10]

Es una aplicación orientada a la elaboración y diseño de aplicaciones web. En sus últimas versiones, la adaptación a los nuevos estándares ha permitido que esta herramienta sea de gran utilidad al proporcionar facilidades tales como el autocompletado, el copy/paste inteligente, testear el código dentro de la propia aplicación, o bien, en varios navegadores, etc.

Para añadir estas funcionalidades, la herramienta utiliza un diseño basado en Live View e incorpora un editor de texto inteligente que apoyará el desarrollo en tiempo real mediante la aportación de sugerencias, verificación de la sintaxis, etc.

Adobe Photoshop.[11]

Para crear los diseños representativos de la marca FPM se ha utilizado la herramienta Adobe Photoshop cuya finalidad es precisamente el tratamiento de imágenes. Específicamente para el diseño del banner y el logotipo de la aplicación se han utilizado distintas composiciones dentro del espectro de posibilidades que aporta este software sobradamente reconocido.

Chrome/Mozilla/Edge (herramientas de desarrollo).

Con las herramientas de desarrollo se podrán inspeccionar elementos de la página e incluso aplicar modificaciones que permiten observar cómo afectan en tiempo real. De esta forma, la búsqueda de una solución se optimiza. Además, su sistema de consola permitirá mostrar errores que indicarán la línea exacta que se debe revisar. También se podrá utilizar para hacer debug.

Otro de sus usos relevantes es la posibilidad de consultar los datos devueltos en las consultas Ajax que se realizan, de tal forma que se puedan detectar los errores de forma visual, en lugar de tener que intuir errores, algo común en el desarrollo web. En definitiva, es una herramienta indispensable para superar los obstáculos que van surgiendo durante el desarrollo.

Datatables JS.[12]

Es una librería que se aplica en el 'frontend', basada en jQuery y cuya funcionalidad es lograr que el formato de las tablas sea más completo, ya que permite aplicar filtrados, y listados ordenados de una forma bastante sencilla. Su funcionamiento se basa en una configuración inicial del Datatable mediante html, css y javascript que el navegador se encarga de reinterpretar para transformar una tabla estática en una tabla más completa.

Xampp.[13]

Se trata de una suite que viene preparada (configuraciones por defecto) y que permite realizar pruebas de un desarrollo en un servidor local. En esencia se compone de un sistema operativo Linux por ser libre y por su buen rendimiento. Añade también el servidor Apache, un servidor web 'free code' para el intercambio de contenidos. En relación a las bases de datos, ya se ha mencionado que se utilizará MySQL, que ya viene incorporado, aunque se podría haber optado por MariaDB. Por último, destacar PHP como lenguaje de programación del lado servidor.

Monday.com

Con Monday.com se ha podido desarrollar el diagrama de Gantt que incluye las tareas y subtareas de forma organizada para la planificación del proyecto. La diferenciación de colores y tamaños, junto a los distintos indicadores temporales permiten visualizar de forma sencilla en qué punto del desarrollo estás. Además, aunque en el diagrama no se visualiza, también ofrece otras

vistas que permiten indicar el estado en el que se encuentra cada tarea (en curso, finalizada, aceptada, etc.). Sin embargo, a la hora de realizar la exportación, no ofrece posibilidades dignas, solamente una versión Excel que no deja claro a simple vista cómo se ha planificado. Es decir, pierde todo aquello que un buen diagrama de Gantt aporta. Por este motivo, se ha optado por aplicar una solución manual que ha consistido en realizar capturas y pegarlas en el Paint de tal forma que el diagrama quede bien alineado y se permita comprender mejor con un solo vistazo.

MockFlow.

Esta herramienta online permite diseñar el ‘wireframe’ de la web. Se ha optado por esta web por su facilidad de implementación sin necesidad de realizar ningún tipo de instalación en el equipo. Sus distintos módulos permiten navegar entre las distintas secciones de la web fácilmente, o bien, entre los distintos elementos que podrán ser arrastrados dentro de una web (desde inputs hasta rectángulos sin etiquetado) y posteriormente redimensionados y posicionados según se desee.

3. Selección del proyecto

La selección del proyecto se puede desglosar en tres fases descritas a continuación:

- **Identificación de las diferentes áreas propuestas.**
  Esta primera etapa consiste en recabar información relativa a la realización del trabajo final de grado en la Universitat Oberta de Catalunya. Se trata de conocer su propuesta, su funcionamiento y ser capaz de ubicarse dentro del contexto donde se desarrollará la idea. En definitiva, comprender en qué va a consistir realizar un TFG, lo que se espera del alumno y del profesor.

- **Análisis en base a mis capacidades.**
  La siguiente fase busca que el alumno sea capaz de decidir qué camino específico va a tomar. Las diversas áreas que propone la universidad junto a sus buenas descripciones han permitido discriminar aquellas posibilidades que distan del conocimiento del alumno. Así pues, se decide preseleccionar tres áreas que se consideran adecuadas y que se ajustan a las posibilidades, sin dejar de lado el cumplimiento de los requisitos que cada una de estas presenta, acotando un poco más la posibilidad de elección.

- **Plantear posibilidades.**
  En este punto, las áreas genéricas preseleccionadas deben materializarse en proyectos específicos. Así pues, se aplica un ‘brainstorming’ a nivel particular, donde se pueden contemplar ideas relacionadas con la gestión empresarial, el deporte o la creación y gestión de una base de datos. Finalmente se opta por una idea que siempre había estado presente y que versará sobre la creación de un sitio web cuya temática será deportiva. Sin embargo, las alternativas no se descartan hasta que el profesor valida la propuesta inicial.

4. Evaluación de riesgos

Si bien el proyecto es el producto de un proceso meditado y analizado donde pueden convivir la motivación y la aplicación de las destrezas adquiridas en el grado, siempre hay que tener presentes las posibles complicaciones, de entre las cuales conviene destacar las siguientes:

- **Complicaciones en la comunicación**/seguimiento de la actividad por causas ajenas y que escapan del control personal, lo que puede desencadenar malas decisiones en el transcurso del proyecto.

- **Objetivos irreales**. Es decir, la posibilidad de no haber valorado con suficiente cautela el alcance del proyecto, cuyo plazo ha sido fijado con antelación. El resultado podría ser un proyecto de menor calidad o que no llegue al nivel requerido para cumplir con el objetivo.

- **Planificación imprecisa**. Es posible que durante la elaboración de este plan se hayan marcado unos objetivos cuyas fechas puedan variar en cualquiera de sus vertientes (de forma positiva o negativa). Sin embargo, se han analizado exhaustivamente los objetivos con el fin de que sean acordes a los plazos marcados, confiando en que la tendencia se decline hacia la vertiente positiva. Además, el hecho de trabajar sobre una metodología 'agile' como es Scrum facilitará la posibilidad de retomar 'sprints' anteriores para mejorarlos o corregirlos.

- **Problemáticas derivadas del entorno de trabajo.** Se puede dar la posibilidad de que las herramientas de trabajo que se pretende usar, presenten algún problema puntual, de la misma manera que el propio hardware pueda estropearse. Para el caso, se mantendrá una copia del proyecto y sus entregables asociados en la nube. Respecto a los problemas técnicos, la propia comunidad de desarrolladores/informáticos en la red permite localizar soluciones para prácticamente cualquier problema.

5. Primer acercamiento al campus

La planificación del trabajo viene marcada por una serie de pautas vinculadas a la realización de la 1ª prueba de evaluación continua de la asignatura. No obstante, de forma simultánea a su ejecución se realizan también una serie de acciones que afectan a su cumplimiento y que a continuación se describen:

- **Toma de contacto aula.**
  El primer punto que tratar tiene que ver con la familiarización del alumno y el entorno de trabajo que propone la universidad, es decir, la concreción de un profesor asociado al área de TFG seleccionada, la creación de un aula específica para la materia y sus correspondientes publicaciones a nivel informativo sobre la materia.

- **Plantear la idea.**
  Tras varios días de análisis que dieron como resultado una serie de ideas ordenadas por nivel de prioridad, se plantea la idea inicial al profesor, de tal forma que pueda notar un interés por parte del alumno y con el fin de establecer un 1er contacto que permita intuir el seguimiento que el profesor desea ofrecer.

- **Negociación con el profesor.**
  De la comunicación con el profesor se concluye que la interacción a nivel individual mediante correo electrónico será muy accesible. Además de mostrar interés en la idea propuesta (y paralelamente confirmar el alumno que esta será la idea que se llevará a cabo) el profesor propone la realización de borradores de las pec para poder asesorar, previa entrega oficial, sobre los errores más evidentes.

- **Producción borrador PEC 1.**
  Con el fin de presentar de forma ordenada y consecuente la idea de TFG se siguen las indicaciones de la PEC 1, consistente en la elaboración del plan de trabajo, que ha servido como base para la realización de la 2ª PEC, ya que se desarrolla sobre este primer documento. Así pues, se produce un 1er borrador del plan de trabajo y se envía al profesor según lo hablado.

- **Modificaciones post-feedback.**
  Del entregable propuesto se obtiene una respuesta en forma de 'feedback' que permite pulir los errores más relevantes mientras que a su vez, se estrecha la actividad comunicativa entre el profesor y el alumno.

- **Producción y entrega definitiva PEC 1.**
  Se procede a la corrección de los errores mencionados y a la presentación del entregable dentro del apartado del campus virtual desarrollado para este fin y sin exceder el plazo de entrega estipulado.

6. Análisis de requisitos

El análisis de requisitos tratará de establecer si las necesidades del proyecto pueden ser cubiertas y también permite definir el conjunto de elementos que deben existir para llevarlo a cabo, confirmando así su viabilidad.

- **Determinar el alcance.**
  Considerar la magnitud del proyecto y evaluar si es posible su cumplimiento es uno de los mayores temores a la hora de plantear una idea. Sin embargo, la planificación previa y el tiempo dedicado a analizar las posibles causas de fracaso han permitido aportar la seguridad necesaria para confirmar que prácticamente la totalidad del producto será realizado dentro del plazo establecido.

  Para amortiguar las complicaciones que pueden derivarse de un proyecto de este calibre se ha confeccionado un plan de trabajo (la 1ª PEC) cuyos plazos establecidos para las tareas tienden a ser superiores a los que realmente se consideran necesarios para su realización. Esta es la principal baza para equilibrar los riesgos que no se pueden prever. Además, esta posibilidad permitirá que, de cumplirse los objetivos con antelación, el resto del tiempo se pueda dedicar a la ampliación o perfeccionamiento de la idea.

  Por último, cabe indicar que la experiencia personal en este tipo de áreas (desarrollos web) ha facilitado la capacidad de estimar plazos para cada uno de los objetivos, es decir, los plazos han sido fijados coherentemente, por lo que el nivel de precisión en este aspecto es elevado, dando lugar a la creación de un proyecto que objetivamente se puede lograr. En definitiva, se reduje el grado de incertidumbre.

- **Confección del entorno de trabajo / Selección de herramientas de producción.**
  A la hora de emprender las actividades es necesario definir en qué contexto quieres aplicar los conocimientos y cuáles serán las herramientas sobre las que te apoyarás para poder materializar el desarrollo.

  En este caso, se parte de la experiencia adquirida a nivel laboral donde ya existe un entorno de trabajo adecuado que se ha optado por extender a este proyecto. De la misma forma que los lenguajes de programación utilizados o las herramientas de trabajo seleccionadas. Sobre estas ya se ha hablado en el punto número 2 (Tecnologías utilizadas) de este documento.

  La decisión se ha tomado partiendo de la premisa de que, si algo se ha comprobado que funciona bien, lo mejor es no tocarlo, o bien, no cambiarlo en exceso. En este caso particular, se entiende que el plazo de entrega es muy estricto, por lo que no es el contexto oportuno para realizar probaturas.

7. Definición de la base de datos

Para la correcta definición de la base de datos se han tenido en consideración los siguientes puntos:

- **Revisión de restricciones y relaciones.**
  Este proceso se encarga de la revisión constante de las relaciones y las restricciones que necesita la base de datos para ajustarse a los requisitos del proyecto. No se trata de un proceso automático, sino de la capacidad de análisis personal para adaptar los datos reales en un entorno digital.

- **Análisis de su propósito.**
  En este punto, se plantea el motivo por el cual una base de datos es adecuada para el proyecto. La necesidad surge en el momento en el que se evidencia la necesidad de cargar, modificar y, en definitiva, trabajar con una serie de datos.

  A grandes rasgos se pueden intuir datos relativos a los jugadores de fútbol, a los usuarios que participarán en la competición, a los clubs, a las comunidades que crearán los usuarios o incluso a los datos relativos a los mercados de fichajes dentro de cada comunidad lo que justifica la necesidad de trabajar con datos.

- **Estructuración.**
  La composición global y desordenada de información existente en la mente inicialmente debe convertirse en datos de utilidad bien estructurados, para ello, se desarrolla un modelo entidad-relación que establece las tablas (entidades) y las relaciones que presentan entre sí. El resultado es el que se puede visualizar en la Ilustración 2:

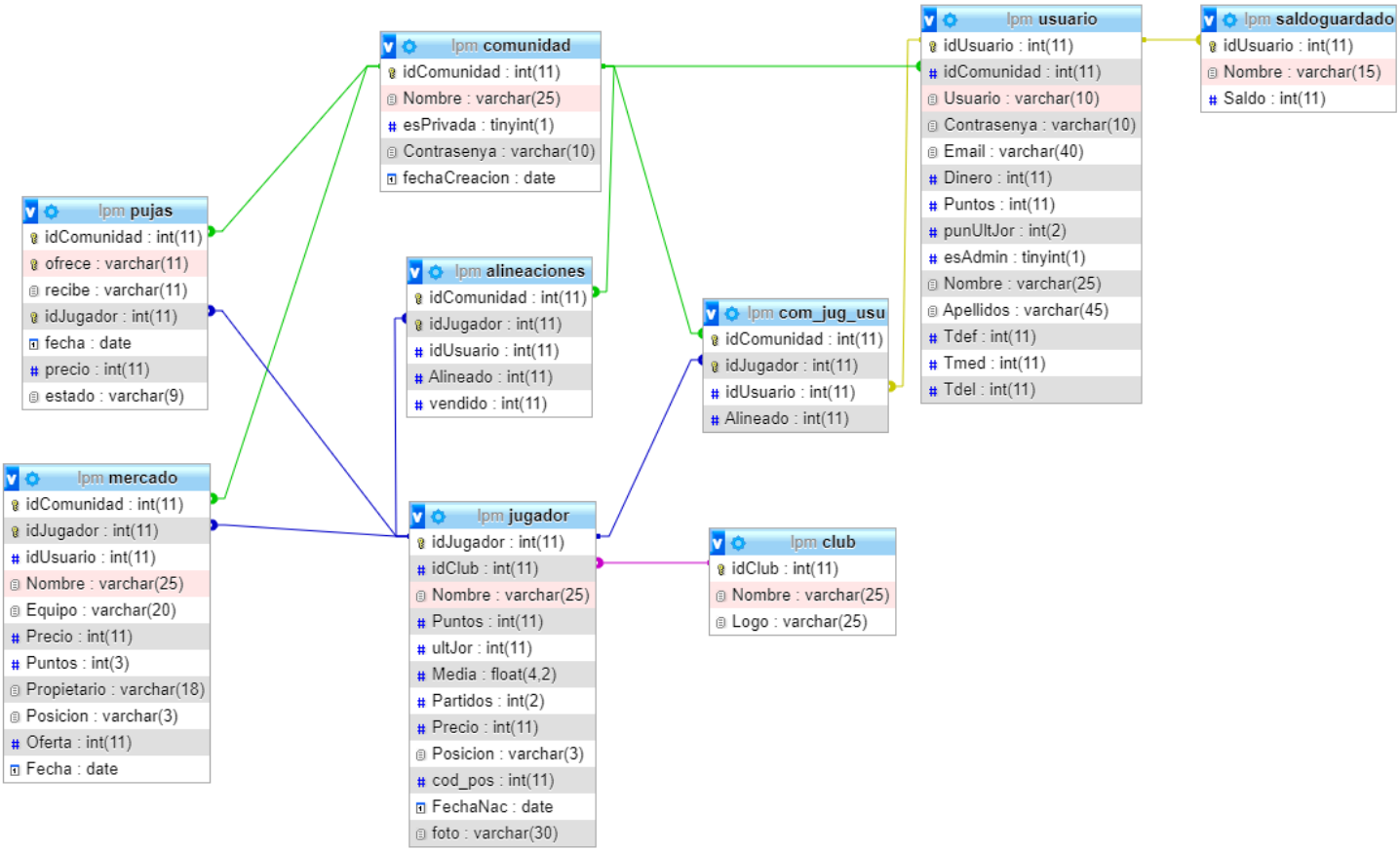


Ilustración 2. Modelo entidad-relación de la base de datos Futbol Pro Manager.

Las entidades que se han localizado se desglosan a continuación:

7.1 Tabla Comunidad.

Contiene los datos relativos a las comunidades creadas por los usuarios. Sus atributos son:

- **idComunidad:** Contiene el identificador de la comunidad. Se trata de un identificador autoincrementable que como es lógico, será de tipo entero. Además, se utilizará este atributo como clave primaria de tal forma que cada número identificará unívocamente a una comunidad.
- **Nombre:** Especifica el nombre que el usuario ha otorgado a una comunidad en el momento de su creación. Será de tipo varchar con un máximo de 25 caracteres y no podrá asignarse un valor nulo.
- **esPrivada:** Determina si la comunidad es pública o privada, es decir, si tendrá asociada una contraseña o si por el contrario será de libre acceso. Se configura un tipo de valor tinyint que solamente permitirá un valor de 1 (privada) o 0 (pública). Se trata de un valor que no permite nulos.
- **Contrasenya:** Almacena la contraseña definida por un usuario (creador de la comunidad) para una comunidad. En esta ocasión es posible el valor nulo cuando la comunidad es pública.

- **fechaCreacion:** Se considera interesante almacenar la fecha de creación de la comunidad para futuros usos. Se trata de un valor de tipo date y que no permite valores nulos.

7.2 Tabla jugador.

Contiene los datos relativos a los datos de los jugadores de fútbol y sus atributos son los siguientes:

- **idJugador:** Identifica a un jugador de forma unívoca. Se trata de un id autoincrementable que actúa como clave primaria.
- **idClub:** Se trata de un campo que hace referencia al identificador de un club que se puede ubicar en la tabla club. De esta forma, este campo no puede contener valores que no existan en dicha tabla de referencia.
- **Nombre:** Almacena el nombre del jugador de fútbol mediante un campo de tipo varchar con un máximo de 25 caracteres cuyo valor no puede ser nulo.
- **Puntos:** Contiene el número de puntos totales que ha obtenido un jugador hasta el momento. Se trata de un campo de tipo entero y no puede ser nulo.
- **ultJor:** Guarda la cantidad de puntos que ha obtenido un jugador en la última jornada que se ha desarrollado. La última jornada según el momento puede hacer referencia a la jornada actual (distintos términos para dirigirse al mismo concepto). Se trata de un valor de tipo entero y no será nunca nulo.
- **Media:** Almacena la media de puntos del jugador en relación al nº de partidos en los que ha participado. Se trata de un valor de tipo float al poder producir decimales. Tampoco puede ser nulo.
- **Partidos:** Define el nº de partidos en los que ha participado un jugador (partidos jugados). Valor de tipo entero y no podrá ser nulo.
- **Precio:** Contiene el coste del jugador de cara a que los usuarios puedan ficharlos. Es un valor de tipo entero que no puede ser nulo.
- **Posicion:** Se trata de la posición del jugador (POR, DEF, CEN o DEL). Así pues, almacena valores de tipo varchar con un máximo de 3 caracteres y no podrá ser nulo.
- **cod_pos:** Está relacionado con el campo posición. Define un valor de 1 para los porteros, 2 para los defensas, 3 para los centrocampistas y 4 para los delanteros. Se trata de un campo de tipo entero que nunca será nulo.
- **FechaNac:** Indica la fecha de nacimiento del jugador. Es un valor de tipo date, que en este caso si podrá ser nulo, pues es prescindible para los intereses de la aplicación.
- **foto:** Guarda el nombre que tendrán las fotos de los jugadores (incluida la extensión del archivo) para posteriormente poder cargarlas. Se trata de un campo de tipo varchar con un máximo de 30 caracteres.

7.3 Tabla usuario.

Contiene los datos relativos a los usuarios de la aplicación, ya sea a nivel general (nombre, apellidos, etc) o más específicos del juego (puntos, táctica, etc.). Sus atributos son los siguientes:

- **idUsuario:** Indica el identificador del usuario. Es la clave primaria de la tabla ya que solamente refiere a un jugador unívocamente. Es un id del tipo autoincrementable.
- **idComunidad:** Guarda el id de la comunidad a la que pertenece el usuario y que obtiene mediante una restricción de clave foránea, por lo que los valores que almacenará dependerán de como esté definido este campo en la tabla originaria (comunidad).
- **Usuario:** Almacena el nombre del usuario en base a lo que este introduzca al registrarse en la aplicación. Es un varchar que no podrá tener más de 10 caracteres y su valor no será nulo.
- **Contrasenya:** Mantiene el valor de las contraseñas con las que se registra el usuario. Es un campo de tipo varchar y su tamaño máximo será de 10 caracteres y es obligatorio su valor.
- **Email:** Guarda el correo electrónico con el que se registra el usuario. Se trata de un campo de tipo varchar que permitirá un máximo de 40 caracteres.
- **Dinero:** Indica el saldo del que dispone el usuario para desenvolverse en el mercado de fichajes y hacer ofertas a otros jugadores. Por defecto su valor es de 20.000.000 y nunca podrá ser nulo.
- **Puntos:** Indica el nº de puntos que ha obtenido el usuario desde que se registró inicialmente hasta el momento actual. Se trata de un campo de tipo entero que no puede dejarse vacío.
- **punUltJor:** Muestra los puntos que ha obtenido un usuario en la última jornada (jornada actual si sigue en curso). Se trata de un campo de tipo entero que no puede ser nulo.
- **esAdmin:** Indica si el usuario es propietario de una comunidad o no. Es un campo de tipo tinyint cuyo valor 1 indica que sí que es propietario y su valor 0 indica que no lo es. También es un valor que no podrá ser nulo.
- **Nombre:** Almacena el nombre “real” del usuario. Es de tipo varchar y será máximo de 25 caracteres.
- **Apellidos:** Guarda los apellidos del usuario. Es de tipo varchar y será máximo de 45 caracteres.
- **Tdef:** Almacena el nº de defensas que tiene la táctica que ha seleccionado el jugador, por defecto: 4. Se trata de un campo de tipo entero que no puede ser nulo. Permite cargar la táctica guardada por el usuario.
- **Tmed:** Almacena el nº de centrocampistas que tiene la táctica que ha seleccionado el jugador, por defecto: 3. Se trata de un campo de tipo entero que no puede ser nulo. Permite cargar la táctica guardada por el usuario.
- **Tdel:** Almacena el nº de delanteros que tiene la táctica que ha seleccionado el jugador, por defecto: 3. Se trata de un campo de tipo entero que no puede ser nulo. Permite cargar la táctica guardada por el usuario.

7.4 Tabla alineaciones.

Tabla clave que se encarga de almacenar los jugadores que tiene alineados un usuario perteneciente a una comunidad concreta justo antes del inicio de la jornada. La idea es capturar un instante concreto que es 5 minutos antes de

que de comienzo una jornada de futbol para que los cambios posteriores o durante la jornada (ventas, fichajes, cambios de alineación) no afecten a la decisión tomada en ese instante específico. Esta tabla se resetea cada inicio de nueva jornada para almacenar las nuevas decisiones del usuario. En cuanto a sus atributos, son los siguientes:

- **idComunidad + idJugador:** Ambos atributos conjuntamente conforman la clave primaria de esta tabla. De esta forma se permite que un mismo jugador pueda ser alineado por más de un usuario (Messi puede ser alineado por Pepe en una comunidad y por Jordi en otra comunidad distinta). Ambos atributos de forma individual son clave foránea, haciendo referencia a valores existentes en las tablas originales.
- **idUsuario:** Almacena el identificador del usuario. También es clave foránea de la tabla usuario. Así pues, su valor y forma dependerán de la definición que tenga en la tabla original.
- **Alineado:** Indica si dicho jugador ha sido alineado o se queda en el banquillo esta jornada. Es un campo de tipo entero, con un 1 indica que sí que está alineado, con un 0 que no lo está.
- **vendido:** Indica si un jugador que ha sido alineado al comienzo de la jornada se ha vendido a otro jugador o a la computer durante el transcurso de la propia jornada. Esto permitirá controlar diversos factores dentro de la aplicación. Por defecto, su valor es 0 (no vendido), mientras que puede pasar a ser un valor de 1 en el momento de su venta.

7.5 Tabla com_jug_usu.

Tabla que almacena cada posible combinación de jugador + usuario + comunidad. Permite controlar diversas cosas, por ejemplo, que, al registrarse un usuario en una comunidad existente, no se le puedan asignar jugadores que ya tiene otro jugador en esa misma comunidad. Otro ejemplo de uso se puede ver a la hora de actualizar el mercado de fichajes ya que al recargar el listado de jugadores del mercado no podrán aparecer jugadores que ya posea algún jugador dentro de esa misma comunidad. Así pues, las restricciones que aplica esta tabla permitirán conectar esa serie de circunstancias mediante el uso de los siguientes atributos:

- **idComunidad + idJugador:** Ambos atributos conjuntamente conforman la clave primaria de esta tabla. De esta forma se podrán encontrar jugadores repetidos, pero que pertenecen a una comunidad distinta (1 jugador determinado solamente podrá existir 1 sola vez en cada comunidad). Ambos atributos de forma individual son clave foránea, haciendo referencia a valores existentes en las tablas originales.
- **idUsuario:** Almacena el identificador del usuario. También es clave foránea de la tabla usuario. Así pues, su valor y forma dependerán de la definición que tenga en la tabla original.
- **Alineado:** Indica si dicho jugador ha sido alineado o se queda en el banquillo esta jornada. Es un campo de tipo entero, con un 1 indica que sí que está alineado, con un 0 que no lo está.

7.6 Tabla club.

Almacena los datos relativos al listado de clubs de fútbol de la 1ª división española, por lo que contendrá los siguientes atributos:

- **idClub:** Guarda el identificador de cada club. Es la clave primaria y se define como un valor de tipo entero autoincrementable.
- **Nombre:** Indica el nombre de cada club de forma textual. Se trata de un campo de tipo varchar de 25 caracteres. Es un campo que no puede contener valores nulos.
- **Logo:** Guarda de forma textual el nombre del fichero que hace referencia al escudo de cada equipo, incluida la extensión del mismo. Es un campo de tipo varchar que permite un máximo de 25 caracteres.

7.7 Tabla mercado.

Contiene el listado de jugadores que hay en el mercado de fichajes para cada comunidad (el mercado se actualiza cada x tiempo). Aquí se incluyen jugadores generados automáticamente por la computer, o bien, jugadores que un usuario ha puesto a la venta. Contiene los siguientes atributos:

- **idComunidad + idJugador:** Ambos atributos conjuntamente conforman la clave primaria de esta tabla. De esta forma se podrán encontrar jugadores repetidos, pero que pertenecen a una comunidad distinta (1 jugador determinado solamente podrá existir 1 sola vez en cada comunidad). Ambos atributos de forma individual son clave foránea, haciendo referencia a valores existentes en las tablas originales.
- **idUsuario:** Almacena el identificador del usuario. Se trata de una clave foránea que remite a la tabla original usuario. El valor y forma de este campo dependerá de la definición en la tabla originaria.
- **Nombre:** Guarda el nombre del jugador de fútbol. Es un campo varchar de máximo 25 caracteres que no puede ser nulo.
- **Equipo:** Contiene el nombre del fichero que hace referencia al escudo del club al que pertenece. Campo varchar de un máximo de 20 caracteres que no puede ser nulo.
- **Precio:** Guarda el coste de cada jugador, es el valor mínimo que deberá pujar un jugador para tratar de hacerse con sus servicios. Es un campo del tipo entero que no puede ser nulo.
- **Puntos:** Indica el número de puntos que ha obtenido un jugador hasta el momento. Es un campo de tipo entero con un máximo de 3 dígitos (cifras de 4 dígitos son impensables en este contexto).
- **Propietario:** Almacena el nombre del propietario de dicho jugador en ese instante (siempre dentro del contexto de una comunidad concreta, ya que un jugador en el mercado en una comunidad, puede ser propiedad de otro usuario en una comunidad distinta). Se trata de un campo de tipo varchar con un máximo de 18 caracteres y que no podrá ser nulo.
- **Posición:** Indica con 3 letras la posición del jugador (POR, DEF, CEN o DEL). Es un campo de tipo varchar, cuya extensión no puede ser superior a 3 caracteres y el valor no podrá ser nulo.
- **Fecha:** Guarda la fecha en la que dicho jugador apareció en el mercado de fichajes. Es un campo del tipo date.

7.8 Tabla pujas.

Contiene todas las pujas (ofertas) que han realizado los usuarios en las distintas comunidades por un jugador de los que aparecen en el mercado de fichajes. Consta de los siguientes atributos:

- **idComunidad + idJugador + ofrece:** Estos 3 atributos conforman la clave primaria de la tabla. En una misma comunidad puede haber varias pujas de distintos usuarios por un mismo jugador, pero nunca podrá haber dos pujas de un mismo usuario por un mismo jugador. De esta forma se controla. Si se desglosa, el campo idComunidad e idJugador son claves foráneas de sus respectivas tablas comunidad y jugador, mientras que el campo ofrece contiene el valor del usuario que realiza la oferta/puja a otro jugador. Es un campo de tipo varchar con un máximo de 11 caracteres y cuyo valor no será nulo.
- **recibe:** Contiene el usuario que ha recibido la puja. Es un campo de tipo varchar que permite un máximo de 11 caracteres. No podrá ser nulo.
- **fecha:** Guarda el día en el que se realizó la oferta. Es un campo de tipo date que no podrá ser nulo.
- **precio:** Almacena la cantidad de saldo que ha propuesto un usuario a otro para hacerse con los servicios de un jugador. Es un campo de tipo entero que no podrá ser nulo.
- **estado:** Indica en qué estado se encuentra o ha concluido el resultado de la negociación (aceptada, rechazada, pendiente, etc.). Es un campo del tipo varchar que no permite valores nulos.

7.9 Tabla saldoguardado.

Al inicio de la jornada se guarda en esta tabla el saldo de cada usuario. Puesto que los usuarios que tengan un saldo negativo (inferior a 0) no sumarán ningún punto en la jornada, conviene almacenar su saldo en dicho instante para posteriormente ser consecuente. Ha sido necesario definir los siguientes atributos:

- **idUsuario:** Es la clave primaria. Identifica a un usuario concreto. Este campo además es clave foránea de la tabla usuario, por lo que el valor de dicho campo vendrá restringido por los valores existentes en la tabla originaria.
- **Nombre:** Contiene el nombre del usuario. En estos momentos no tiene demasiada relevancia, pero se conserva para posibles modificaciones futuras. Es un campo de tipo varchar que puede ser nulo.
- **Saldo:** Indica el saldo del jugador en ese instante. Es un campo de tipo entero que permitirá hacer las consultas oportunas a la hora de otorgar puntos a un usuario. No podrá ser nulo.

7.10 Relaciones entre las tablas.

comunidad – pujas (1,M): Una comunidad puede tener ninguna o muchas pujas, mientras que una puja pertenecerá a una única comunidad concreta.

comunidad – mercado (1,1): Una comunidad tiene necesariamente un solo mercado (ni 0, ni más de 1), mientras que un mercado pertenece a una comunidad concreta.

comunidad – usuario (M,1): Una comunidad tiene mínimo un usuario, o bien puede tener muchos. Por su parte, un usuario solamente puede existir en una comunidad.

comunidad – alineaciones (M,1): Una comunidad puede tener una o muchas alineaciones mientras que una alineación podrá aparecer en una comunidad.

jugador – alineaciones (M,M): Un jugador puede pertenecer a ninguna o varias alineaciones, mientras que en una alineación puede haber muchos jugadores.

jugador – mercado (M,M): Un jugador puede aparecer en ninguno o muchos mercados, mientras que un mercado puede contener uno o muchos jugadores.

jugador – pujas (M,1): Un jugador puede haber recibido ninguna o varias pujas, mientras que una puja contiene solamente un jugador.

club – jugador (M,1): Un club contiene muchos jugadores, mientras que un jugador pertenece a un solo club.

usuario – saldoguardado (1,1): Un usuario tiene un único saldo guardado, mientras que el saldoguardado corresponde a un único jugador.

com_jug_usu: Aunque se trata de una tabla, surge de la relación ternaria que hay entre las tablas comunidad, jugador y usuario. Esta relación es clave para lograr el funcionamiento ideado inicialmente (1 jugador no se puede repetir en 1 misma comunidad).

- **Obtención de datos.**
  Una vez creada la estructura con la que se iba a trabajar era necesario localizar fuentes de datos relativas a aquella información que no va a ser generada por la propia aplicación web. Estos datos hacen referencia a la tabla de jugadores (información de los jugadores) y la tabla de clubs (datos de los clubs).

  Puesto que el contexto es deportivo y concretamente, futbolístico, se considera que las mejores fuentes de información para rellenar estas tablas serán: la web de la liga de futbol profesional y las webs de los propios clubs. Adicionalmente, se obtendrán datos también de otros fantasy de futbol existentes como la liga fantasy de marca, misterfantasy, etc.

- **Implantación de datos.**
  Con las fuentes de datos definidas, se procede de forma manual a rellenar nuestras tablas con más de 500 jugadores y los 20 clubs a través de phpmyadmin, lo que retrasa ligeramente el planteamiento

temporal definido, pero que, a cambio, permite “rellenar” de contenido la aplicación.

Los datos relativos a las tablas restantes (inicialmente sin contenido) serán generados con la puesta en marcha de la aplicación (registros de usuarios, creación de comunidades, obtención de puntos, etc.), por lo que, hasta el momento, la implantación de datos queda finalizada en este punto.

8. Diseño de la interfaz

Para diseñar la interfaz de la aplicación se han implementado los siguientes procesos:

- **Recopilación de ideas.**
  Al tratarse de una propuesta existente y que actualmente tiene una gran presencia en internet se ha optado por analizar las distintas alternativas existentes en el mercado para extraer aquello más significativo. Sin embargo, es necesario ser consciente de los plazos definidos para las tareas, ya que, de plantearse un diseño excesivamente elaborado, solamente el diseño podría llevarse la dedicación total del desarrollo, lo que quizás sería más adecuado en un plan de estudios distinto al que ocupa la atención (quizás multimedia o diseño gráfico). Así pues, se debía coger lo más interesante y aplicarlo sin que esto conllevara una dedicación adicional que restase horas de trabajo reservadas al funcionamiento de la aplicación.

- **Diseño de banner y logo.**
  Mediante el uso de adobe Photoshop y una plantilla inicial básica se ha diseñado un logo (ver ilustración 3) con los colores que representan la marca FPM. De la misma manera, se ha diseñado un banner (ver ilustración 4) que ocupara el encabezado de la página. En ambos casos se ha optado por la implementación de elementos relacionados con el mundo del fútbol. Para la tarea se han podido aplicar los conocimientos adquiridos a través del minor de programación web cursado en la UOC.

  Además, este punto que no estaba contemplado inicialmente (al poner el foco en las funcionalidades y no tanto en el diseño) ha dado pie a la selección de la gama cromática que representará la aplicación y que son el azul, el negro y el blanco. Estos colores pasarán a ser los más utilizados también a la hora de aplicar estilo en los distintos elementos de la web.

Ilustración 3. Logo de la aplicación FPM diseñado con Photoshop.

Ilustración 4. Banner de la aplicación FPM diseñado con Photoshop.

- **Bocetaje.**
  Con una serie de ideas interesantes anotadas, el siguiente paso ha sido crear bocetos para cada sección que se pretende implementar. Para su realización, se ha optado por la herramienta online MockFlow, que precisamente ofrece la posibilidad de crear los wireframe necesarios.

  En las siguientes páginas se exponen los diseños realizados para cada sección de la página:

Boceto de la página de inicio.

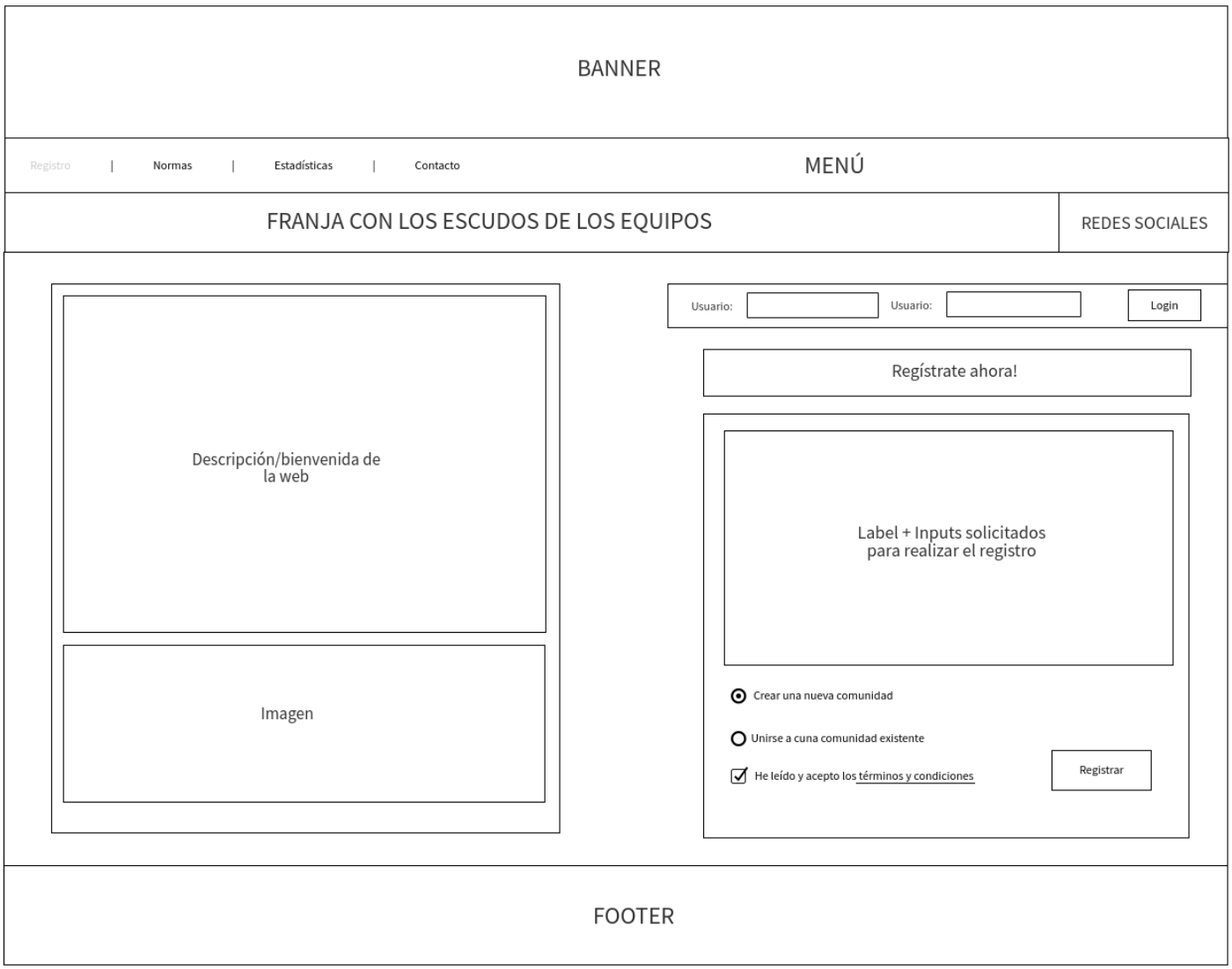


Ilustración 5. Boceto página de inicio.

Boceto de la página de alineación.

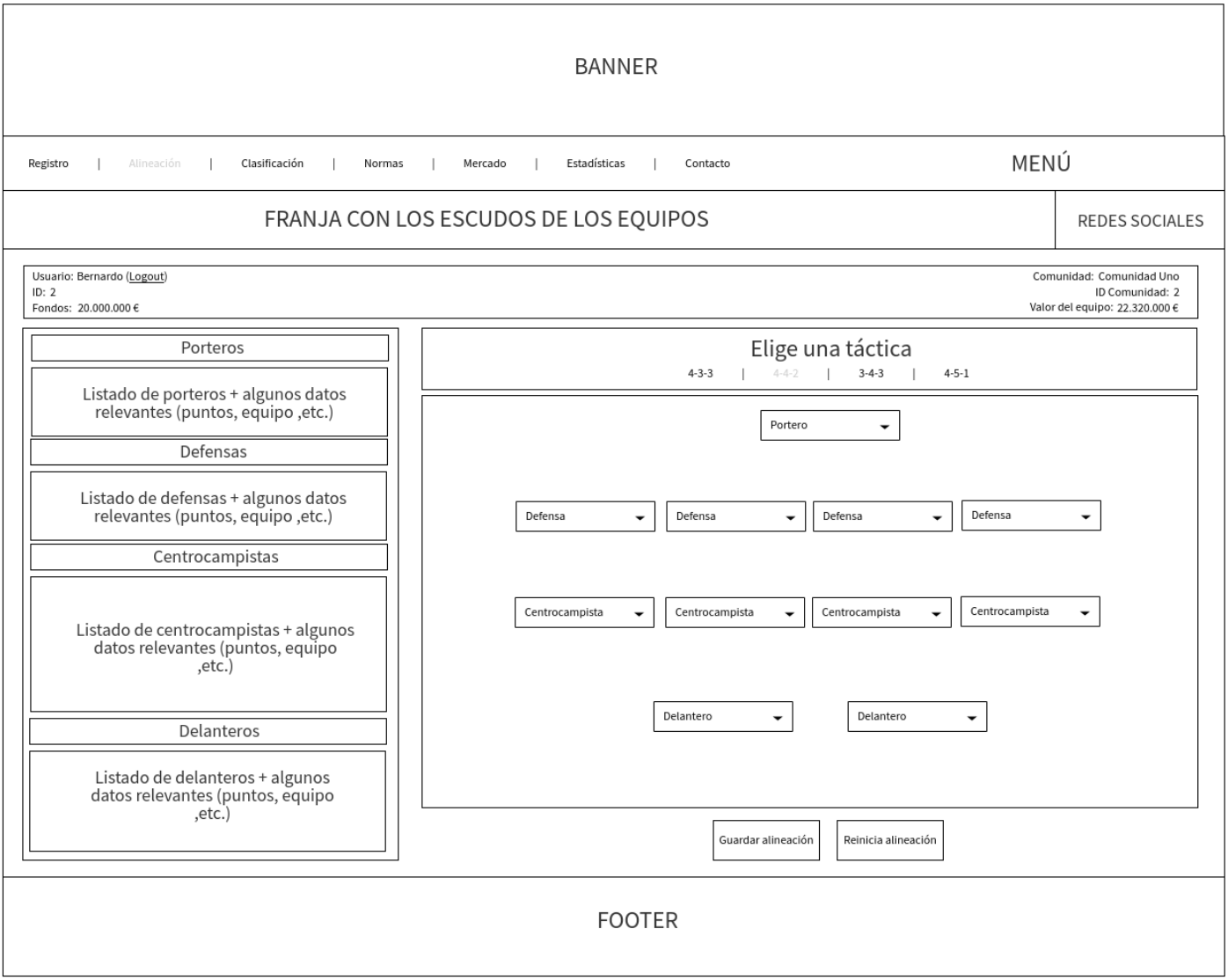


Ilustración 6. Boceto página de alineación.

Boceto de la página de clasificación.

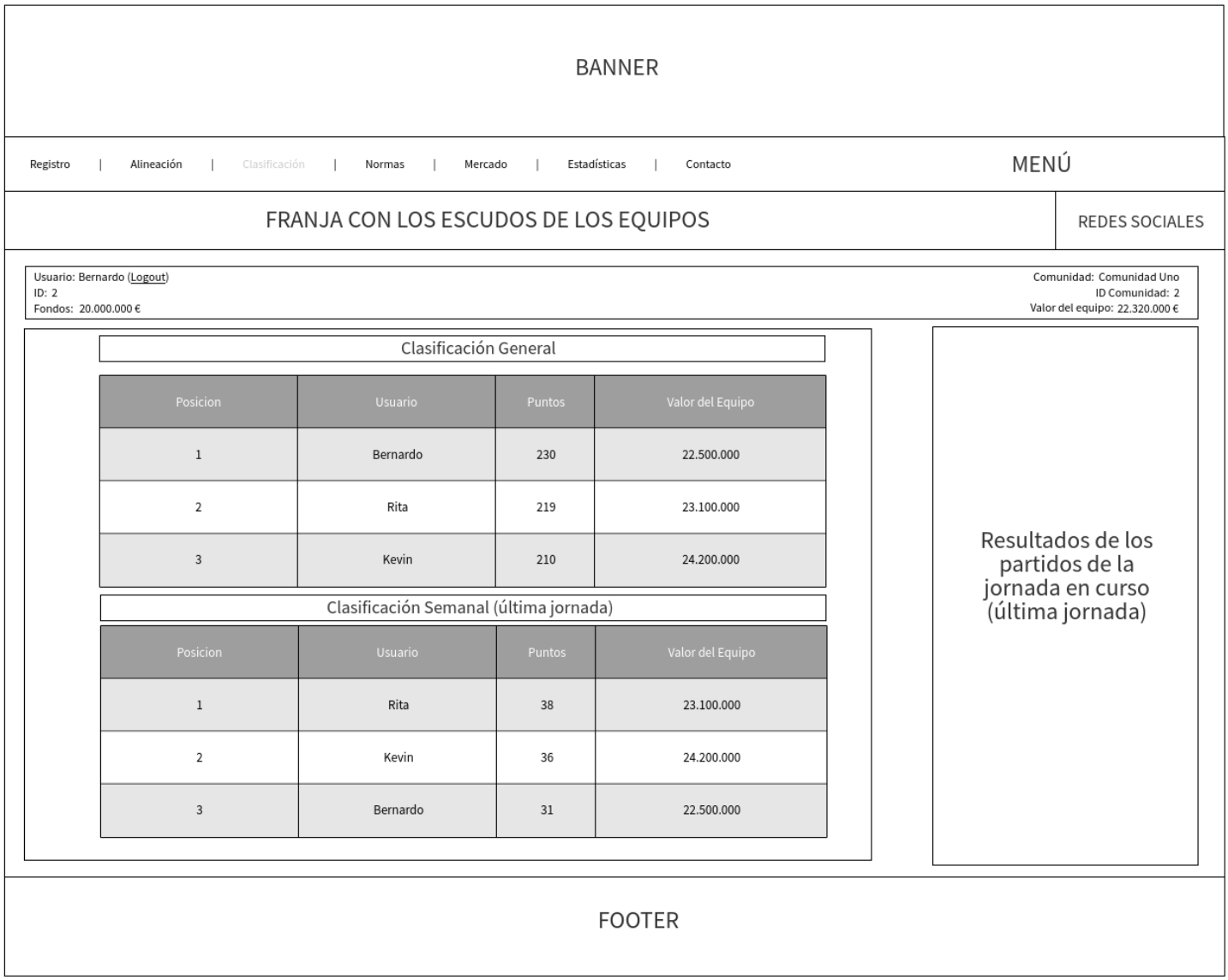


Ilustración 7. Boceto página de clasificación.

Boceto de la página de normas.

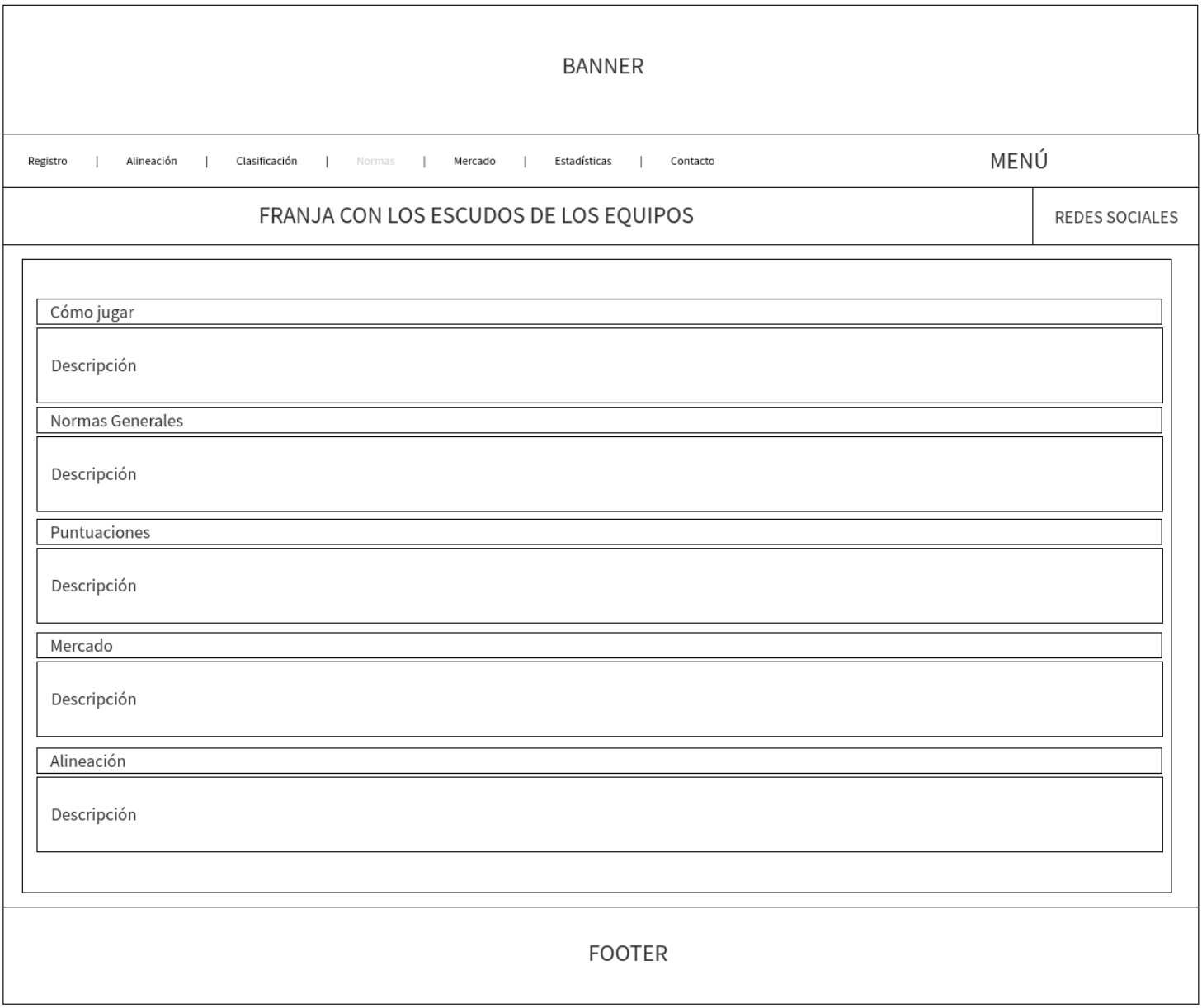


Ilustración 8. Boceto página de normas.

Boceto de la página de mercado.

BANNER

Registro | Alineación | Clasificación | Normas | Mercado | Estadísticas | Contacto

MENÚ

FRANJA CON LOS ESCUDOS DE LOS EQUIPOS

REDES SOCIALES

Usuario: Bernardo (Logout)
ID: 2
Fondos: 20.000.000 €

Comunidad: Comunidad Uno
ID Comunidad: 2
Valor del equipo: 22.320.000 €

Mercado

Jugador	Equipo	Precio	Puntos	Propietario	Posición	Desde	Oferta
jugador1	escudo	valorJugador	puntos	Bernardo	DEL	2021-02-29	
jugador2	escudo	valorJugador	puntos	computer	DEF	2021-02-28	
jugador3	escudo	valorJugador	puntos	computer	DEL	2021-02-28	
jugador4	escudo	valorJugador	puntos	computer	CEN	2021-02-28	
jugador5	escudo	valorJugador	puntos	computer	CEN	2021-02-28	
jugador6	escudo	valorJugador	puntos	computer	POR	2021-02-28	
jugador7	escudo	valorJugador	puntos	computer	POR	2021-02-28	
jugador8	escudo	valorJugador	puntos	computer	DEL	2021-02-28	
jugador9	escudo	valorJugador	puntos	computer	DEL	2021-02-28	
jugador10	escudo	valorJugador	puntos	computer	DEF	2021-02-28	

Ver Ofertas | Ofrece Jugador | Quitar jugador del mercado | Pujar

FOOTER

Ilustración 9. Boceto página de mercado.

Boceto de la página de ver ofertas.

BANNER

Registro | Alineación | Clasificación | Normas | Mercado | Estadísticas | Contacto

MENÚ

FRANJA CON LOS ESCUDOS DE LOS EQUIPOS

REDES SOCIALES

Usuario: Bernardo (Logout)
ID: 2
Fondos: 20.000.000 €

Comunidad: Comunidad Uno
ID Comunidad: 2
Valor del equipo: 22.320.000 €

Ofertas Realizadas

Por	A	Equipo	Precio	Desde	Estado	Acción
nombreJugador	propietario	escudo	400.000	fechaOferta	pendiente	Retirar
nombreJugador	propietario	escudo	600.000	fechaOferta	pendiente	Retirar
nombreJugador	propietario	escudo	10.244.000	fechaOferta	aceptada	

Ofertas Recibidas

Por	De	Equipo	Precio	Desde	Estado	Acción
nombreJugador	propietario	escudo	1.000.000	fechaOferta	pendiente	Aceptar Rechazar
nombreJugador	propietario	escudo	350.000	fechaOferta	pendiente	Aceptar Rechazar
nombreJugador	propietario	escudo	4.300.230	fechaOferta	aceptada	

FOOTER

Ilustración 10. Boceto de la sección para consultar las ofertas realizadas y recibidas.

Boceto de la página ofrece jugador.

BANNER

Registro | Alineación | Clasificación | Normas | Mercado | Estadísticas | Contacto

MENÚ

FRANJA CON LOS ESCUDOS DE LOS EQUIPOS

REDES SOCIALES

Usuario: Bernardo (Logout)
ID: 2
Fondos: 20.000.000 €

Comunidad: Comunidad Uno
ID Comunidad: 2
Valor del equipo: 22.320.000 €

Ofrece Jugador

Jugador	Equipo	Precio	Puntos	Posición	Asigna Precio de Venta		
jugador1	escudo	valorJugador	puntos	POR		Ofrecer	Vender ya
jugador2	escudo	valorJugador	puntos	POR		Ofrecer	Vender ya
jugador3	escudo	valorJugador	puntos	DEF		Ofrecer	Vender ya
jugador4	escudo	valorJugador	puntos	DEF		Ofrecer	Vender ya
jugador5	escudo	valorJugador	puntos	DEF		Ofrecer	Vender ya
jugador6	escudo	valorJugador	puntos	DEF		Ofrecer	Vender ya
jugador7	escudo	valorJugador	puntos	CEN		Ofrecer	Vender ya
jugador8	escudo	valorJugador	puntos	CEN		Ofrecer	Vender ya
jugador9	escudo	valorJugador	puntos	CEN		Ofrecer	Vender ya
jugador10	escudo	valorJugador	puntos	CEN		Ofrecer	Vender ya
jugador11	escudo	valorJugador	puntos	DEL		Ofrecer	Vender ya
jugador12	escudo	valorJugador	puntos	DEL		Ofrecer	Vender ya
jugador13	escudo	valorJugador	puntos	DEL		Ofrecer	Vender ya

FOOTER

Ilustración 11. Boceto de la sección para poner a la venta jugadores.

Boceto de la página Estadísticas.

BANNER

Registro | Alineación | Clasificación | Normas | Mercado | Estadísticas | Contacto

MENÚ

FRANJA CON LOS ESCUDOS DE LOS EQUIPOS

REDES SOCIALES

ZONA PARA AÑADIR FILTROS PARA LA TABLA

Buscar:

Estadísticas jugadores

Jugador	Posición	Equipo	Precio	Partidos	Media	Última Jorn.	Puntos
jugador1	DEL	escudo	20.850.000	19	6.4	5	121
jugador2	DEF	escudo	7.540.000	22	5.4	7	118
jugador3	MED	escudo	9.030.000	22	5.3	5	117
jugador4	DEL	escudo	20.850.000	19	6.4	5	121
jugador5	DEF	escudo	7.540.000	22	5.4	7	118
jugador6	MED	escudo	9.030.000	22	5.3	5	117
jugador7	DEL	escudo	20.850.000	19	6.4	5	121
jugador8	DEF	escudo	7.540.000	22	5.4	7	118
jugador9	MED	escudo	9.030.000	22	5.3	5	117
jugador10	MED	escudo	9.030.000	22	5.3	5	117

N° Resultados

Primero << >> Último

FOOTER

Ilustración 12. Boceto para la página de estadísticas.

Boceto de la página de contacto.

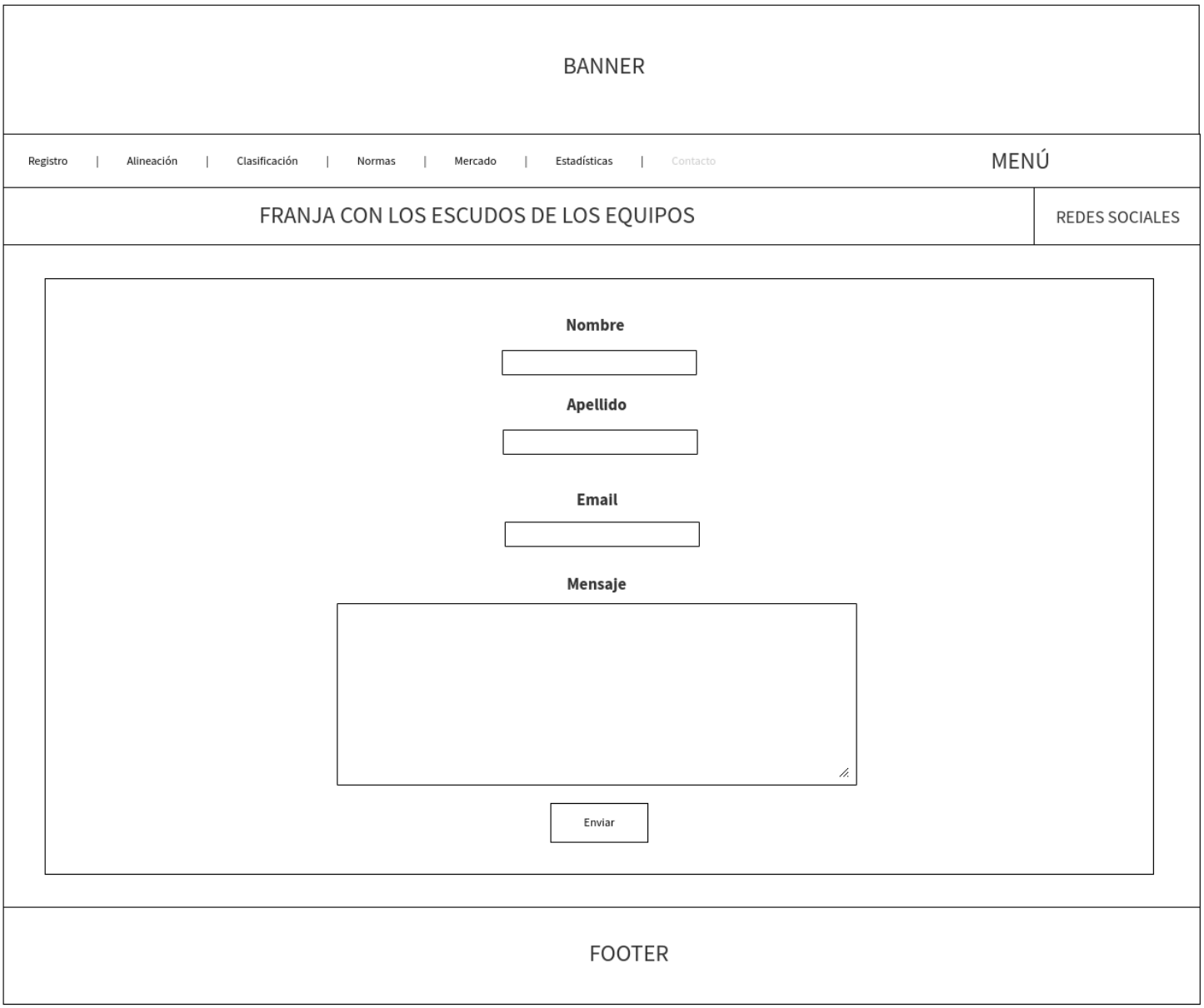


Ilustración 13. Boceto para la página de contacto.

Boceto de la página de administración.

Ilustración 14. Boceto para la página de administración.

9. Implementación

Una vez se han dejado listos los preparativos, se lleva a cabo la implementación de las distintas funcionalidades de la web, por lo que es la hora de abrir Dreamweaver y aplicar los distintos lenguajes de programación que se han descrito con el objetivo de cumplir con las tareas mencionadas de la forma más precisa posible. Seguidamente se detalla cada punto.

9.1 Página de inicio.

Ilustración 15. Vista de la página de inicio de la aplicación Futbol Pro Manager.

- **Sistema de login/logout/registro.**

  El sistema para la identificación de los usuarios (login) se implementa en la página de inicio, que es la que se cargará automáticamente al acceder a la aplicación web.

  Tal y como se puede observar en la ilustración 16, consiste en una barra habilitada para este fin que contiene un par de campos relativos al usuario y la contraseña que deberá introducir el usuario para conectarse a su cuenta.

Ilustración 16. Barra para las acciones relativas a la identificación del usuario.

La validación de los datos introducidos se gestiona mediante la petición desde php a la base de datos mysql, donde se consultará si existe dicho usuario.

En el caso de que los datos insertados no encuentren ninguna coincidencia se obtendrá un mensaje informativo junto a un segundo mensaje (hipervínculo) que permitirá volver a intentar introducir los datos nuevamente como se puede visualizar en la ilustración 17.

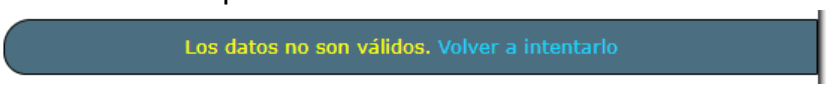


Ilustración 17. Barra de identificación muestra error en la información introducida.

Si, por el contrario, la validación devuelve un usuario existente, la barra mostrará el usuario que está conectado en ese momento y la posibilidad de finalizar la sesión (logout) a través del hiperenlace 'Salir' (ver ilustración 18).

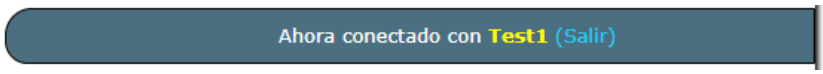


Ilustración 18. Barra de identificación muestra datos del usuario conectado.

En cuanto al registro de usuarios, también se puede localizar fácilmente dentro de la página de inicio, incorporando un formulario que solicitará los datos clásicos a la hora de crear una cuenta en prácticamente cualquier sitio web.

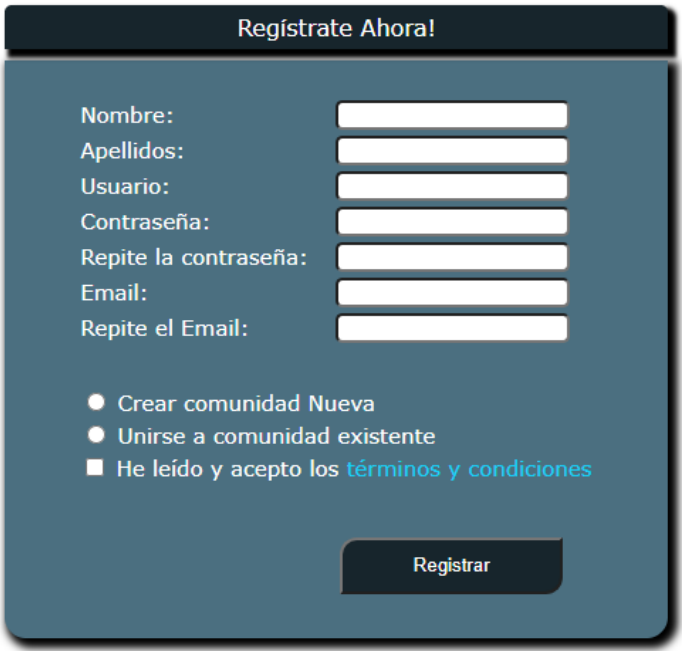


Ilustración 19. Panel de creación de cuenta de usuario.

Su funcionamiento es el ya conocido por cualquier sitio web. Se deberán rellenar todos los campos de forma coherente y finalmente pulsar el botón de registro. A través de php se realizarán diferentes validaciones, como la comprobación de las dos contraseñas que deberán ser

idénticas, de la misma forma que el Email, o bien, la comprobación de si un usuario ya existe en la base de datos, evitando que existan repetidos.

Algunas de las respuestas que puede devolver el servidor al provocar una de estas situaciones son las que muestran las ilustraciones 20 y 21 seguidamente:

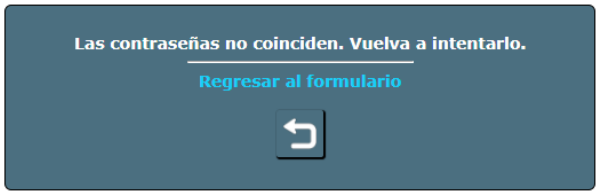


Ilustración 20. Fallo de registro por contraseñas.

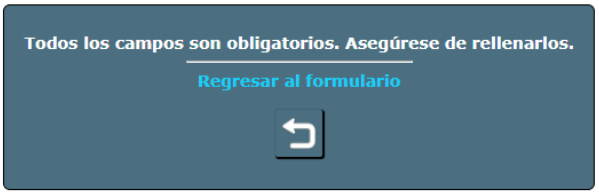


Ilustración 21. Fallo de registro por campos vacíos.

En cualquier caso, todo error emitido como respuesta posibilitará la opción de regresar al formulario de registro nuevamente, ya sea a través del icono con forma de flecha, o bien, del hipervínculo con texto Regresar al formulario.

- **Sistema de creación de comunidad.**
  Inicialmente la creación de la comunidad pretendía realizarse de forma independiente al registro de usuario, de tal forma que un usuario pudiera registrarse y en cualquier otro momento, crear una comunidad, o bien, unirse a alguna existente. Sin embargo, finalmente se ha considerado que permitir la creación de usuarios que pudieran quedar en un "limbo" donde realmente no están participando en ninguna comunidad, no aportaba nada. Es por este motivo, que se incluye la posibilidad de elegir si se desea crear una comunidad o unirse a una nueva en este punto.

  Como ya se ha visto, no se trata de una sección independiente, sino que sus opciones (escuetas) vienen asociadas al propio formulario de registro.

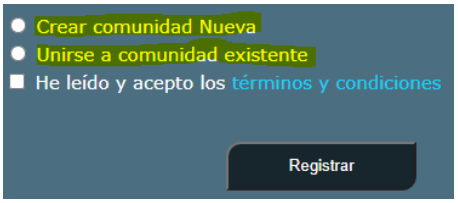


Ilustración 22. Opciones relativas a las comunidades.

Así pues, se proponen dos vertientes. Por una parte, si se selecciona la opción "Crear comunidad Nueva" se estará configurando un nuevo espacio en la aplicación (mercado propio, disponibilidad de jugadores propia, competir contra unos usuarios específicos, etc.). Eso sí, previamente se deberá definir el nombre de dicha comunidad y la contraseña a partir del formulario que se observa en la ilustración 23.

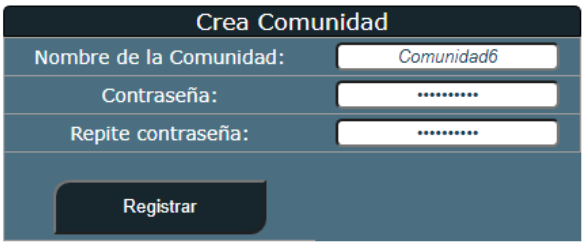


Ilustración 23. Configuración de la comunidad.

Será necesario rellenar correctamente los tres campos para que la aplicación permita avanzar. De lo contrario, se mostrará tal y como se ve en la ilustración 24.

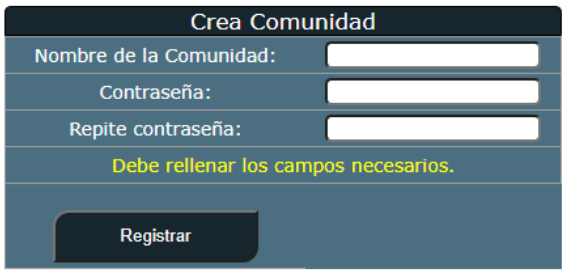


Ilustración 24. Pantalla de error de configuración.

Por su parte, si se opta por la opción de unirse a una comunidad existente a la hora de realizar el registro de usuario, la pantalla mostrará lo reflejado en la siguiente imagen:

Ilustración 25. Pantalla selección de comunidad.

Por último, solamente faltará introducir un par de veces la contraseña de la comunidad en el formulario que se observa en la ilustración 26.

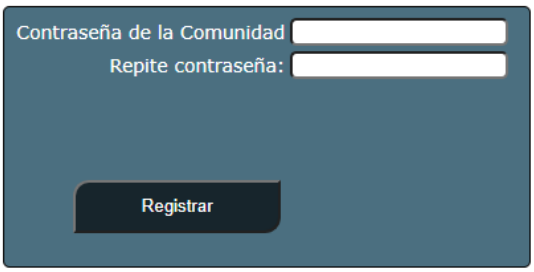


Ilustración 26. Solicitud de clave para acceder a una comunidad.

En definitiva, cualquiera de estos dos caminos llevará al usuario a participar dentro de una comunidad.

Respecto al aspecto técnico, se aplica la misma metodología que para la creación del 'login'. Se aplican validaciones mediante php, que se encarga de consultar la información oportuna en la base de datos mediante el lenguaje mysql. Además, el sistema de redirección de php (header), permite que el usuario se encuentre en cada momento en la sección oportuna para continuar con el proceso.

- **Generación equilibrada de plantilla post-registro.**
  Una vez el usuario finaliza todos los pasos para realizar su registro, independientemente de si se ha optado por crear una nueva comunidad, o unirse a una comunidad existente, la aplicación redirigirá a la pantalla de Alineación, donde se mostrará la plantilla de la que dispones inicialmente.

  Para que el reparto de jugadores a cada usuario que se une a una comunidad no sea del todo desequilibrado se aplican una serie de consideraciones.

  1. **El reparto incluye la asignación de 2 porteros, 4 defensas, 6 centrocampistas y 3 delanteros a cada usuario**. Esto siempre será así, aunque posteriormente el usuario podrá vender y fichar sin limitaciones por posición.

     Para lograr este objetivo se ha realizado una consulta por cada posición (portero, defensa, etc.) a la tabla que contiene los jugadores libres de la comunidad en cuestión (puede haber jugadores ocupados por otros usuarios que se han registrado antes o incluso jugadores en el mercado de fichajes que por lo tanto no podrán ser asignados a un jugador, ya que de existir en el mercado de fichajes y en la plantilla de un usuario, se estarían duplicando, algo que no es correcto ya que un jugador solamente debe existir 1 vez en cada comunidad).

     Seguidamente se han almacenado los resultados de estas consultas en 4 vectores diferenciados y que corresponden a porteros, defensas, centrocampistas y delanteros. Mediante la funcionalidad shuffle de php, se desordenan los cuatro vectores para finalmente seleccionar y obtener el valor de los 2 primeros porteros que hay en el vector de porteros, los 4 primeros defensas de la lista en el array de defensas y el mismo proceso para los centrocampistas y los delanteros.

  2. **El equipo asignado al usuario aleatoriamente no podrá tener un valor inferior a 18.000.000 € y tampoco podrá ser superior a 24.500.000 €.**

     Para poder controlar este punto, tras el paso anterior se suma el valor de todos los jugadores que han sido propuestos y en caso de

que el resultado no se encuentre entre los dos valores mencionados, se repetirá todo el proceso nuevamente hasta que finalmente se cumpla esta condición. Con esto se pretende evitar que un equipo posea jugadores de un valor excesivo para especular en el mercado, o bien, que obtenga una plantilla devaluada que no permita trabajar en el mercado. Además, no hay que olvidar que el valor de un jugador indirectamente indica su potencial, por lo que un usuario con una plantilla muy cara, normalmente será un jugador con una plantilla bastante buena y viceversa.

- **Asignación de saldo y plantilla al usuario.**
  Una vez se ha completado el proceso anterior, se realizará la inserción de la información en las tablas oportunas mediante php y mysql. Es decir, quedará almacenado que para la comunidad donde se ha registrado el usuario, hay jugadores que ya no están libres. De este modo, no podrán aparecer en las actualizaciones de los mercados de fichajes, ni tampoco en el reparto aleatorio cuando un nuevo usuario ingrese en la comunidad.

  Pero no solamente se asigna una plantilla, sino que de forma inicial se reparte un saldo de 20.000.000 € a cada usuario que ingresa en una comunidad para permitir mayor flexibilidad a la hora de moverse en el mercado de fichajes, o bien, para tentar a los rivales con buenas ofertas y hacerte con alguno de sus jugadores. Evidentemente, este valor queda reflejado en la base de datos e irá modificándose consecuentemente a los movimientos que realice el usuario.

- **Tests.**
  Para verificar el funcionamiento de los pasos anteriores la mayor comprobación ha sido repetir el proceso de registro en sus dos vertientes reiteradas veces, probando distintas circunstancias. Además, se ha otorgado la posibilidad a un par de personas (sin conocimientos informáticos) de simular varios registros, para detectar problemáticas o detalles no contemplados.

  El proceso reiterativo de pruebas ha sido necesario para poder verificar que las cosas que se iban implementando funcionaban. Se trata de un proceso que va de la mano de cualquier desarrollo inevitablemente, ya que esto marca el camino para poder pasar a la siguiente funcionalidad prevista. Así pues, la mayor parte de las problemáticas se han ido solucionando conforme aparecían los errores. Una vez se ha logrado la funcionalidad planteada, ha sido cuando los usuarios externos han probado esta parte de la aplicación, no localizando incidencias destacadas, aunque sí que se han aportado mejoras que se anotan para implementar en un futuro.

9.2 Página alineación.

Ilustración 27. Página de alineación y gestión de la plantilla.

- **Carga de datos del perfil del usuario y de la comunidad.**
  Se considera relevante disponer de una sección informativa que permita al usuario ubicar el contexto en el que se encuentra dentro de la aplicación, ya que una misma persona podrá crear varias cuentas y participar en distintas comunidades. Para ello, se configura un panel que aporta la información básica con un solo vistazo. Este panel luce tal y como muestra la ilustración 28.

Ilustración 28. Panel informativo del usuario.

Mediante distintas consultas a la base de datos se podrá recuperar información útil como el usuario que ésta conectado, la comunidad a la que pertenece, el saldo del qué dispone o el valor de su plantilla.

- **Carga de datos de la plantilla.**
  La sección izquierda de esta pantalla se reserva para mostrar la plantilla desglosada por posición de la que dispone un usuario para poder plantar cara a sus rivales. Se puede observar lo descrito en la ilustración 29.

Porteros				
Juan Pérez		150.000	0	0
Defensas				
Postigo		360.000	50	0
Unai García		260.000	30	0
Etxeita		240.000	25	0
Odriozola		220.000	14	0
Centrocampistas				
Koke		7.850.000	88	5
Inui		2.050.000	68	0
Dani García		820.000	57	4
Waldo Rubio		220.000	17	0
John Finn Patrick		150.000	1	0
Camarasa		180.000	0	0
Delanteros				
Kike García		7.270.000	99	4
Fer Niño		1.850.000	38	0
Bacca		280.000	6	0

Ilustración 29. Vista del panel lateral izquierdo con la plantilla del usuario.

En esta pantalla se muestran datos de utilidad para el usuario, como son el nombre del jugador, el club al que pertenecen, su valor en el mercado, los puntos totales que ha obtenido desde que inició la temporada (en la liga real) y los puntos obtenidos la última jornada.

- **Visualización de alineación activa.**
  Como es lógico, un entrenador no se podrá considerar como tal hasta que pueda decidir algo sobre su equipo. En esta ocasión se permite al usuario confeccionar el 11 que competirá antes de cada jornada a partir de la interfaz que se ve en la siguiente imagen.

Ilustración 30. Vista de la alineación elegida por un usuario.

El funcionamiento para la elección de jugadores es sencillo, cada una de las posiciones consta de un desplegable (ver ilustración 31) para poder decidir quién ocupará una determinada posición.

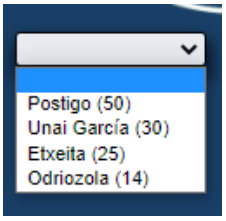


Ilustración 31. Ejemplo de desplegable para elegir defensa.

Sin embargo, aunque para el usuario esta posibilidad es amigable, técnicamente supone todo un desafío. La dificultad reside especialmente en el hecho de que, al seleccionar, por ejemplo, un delantero, los desplegables restantes (para añadir delanteros) no deberán mostrar a este jugador en sus listas (¿jugar con dos Messi? Interesante, pero irreal). Las medidas para resolverlo pasan por la creación de distintas funcionalidades en javascript/jquery junto a diversas llamadas Ajax que permiten trabajar en vivo, sin recargar la totalidad de la página.

- **Indicador ¿a quién alineaste la última jornada?**
  La idea inicial estaba clara, se trataba de destacar de algún modo a aquellos jugadores que el usuario había alineado en la última jornada que se produjo (o jornada actual si todavía está en curso). Esta facilidad se trabaja a través de la combinación de consultas a la base de datos, de donde se obtendrán los resultados relativos a las últimas alineaciones realizadas (guardadas 5 min. antes de dar comienzo la jornada) y la modificación con javascript del diseño de las celdas de la tabla.

  Para la ocasión se ha podido realizar una simulación de lo que sería una vista de usuario con su última alineación destacada. De esta manera, la ilustración 32 permite ver con un fondo azulado aquellos jugadores alineados.

Porteros				
Juan Pérez		150.000	0	0
Defensas				
Postigo		360.000	50	0
Unai García		260.000	30	0
Etxeita		240.000	25	0
Odriozola		220.000	14	0
Centrocampistas				
Koke		7.850.000	88	5
Inui		2.050.000	68	0
Dani García		820.000	57	4
Waldo Rubio		220.000	17	0
John Finn Patrick		150.000	1	0
Camarasa		180.000	0	0
Delanteros				
Kike García		7.270.000	99	4
Fer Niño		1.850.000	38	0
Bacca		280.000	6	0

Ilustración 32. Plantilla con destacado de jugadores alineados en fondo azul y negrita.

  Sencillamente, esta vista permite, como ya se ha mencionado, verificar los jugadores alineados en la última jornada que ha transcurrido de forma sencilla.

- **Selección de táctica.**
  Hasta el momento se ha trabajado con un planteamiento táctico estático de 4 defensas, 4 centrocampistas y 2 delanteros para realizar las comprobaciones y verificar el funcionamiento de las nuevas funcionalidades. En este punto, se añade la posibilidad de decidir entre otras posibles tácticas para exprimir mejor tu plantilla, recargando aquellas posiciones donde tu plantilla sobresale. Para ello se implementa

un panel de selección de tácticas que incluye las más típicas. Es posible ver una representación de lo mencionado en la siguiente imagen:

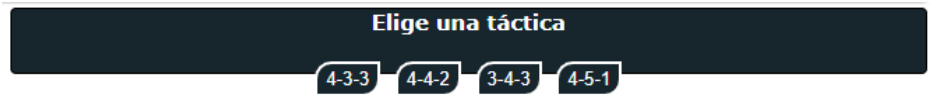


Ilustración 33. Panel de selección de táctica.

Se añaden 4 posibilidades dentro de las tácticas más comunes en el fútbol actual, estas son: 4-3-3, 4-4-2, 3-4-3 y 4-5-1. Sin embargo, esto puede ser modificado en un futuro para permitir más posibilidades.

Como se puede ver en la ilustración 33, la funcionalidad reside en la incorporación de 4 botones, 1 para cada táctica. Cada uno de ellos está asociado a una función javascript que recibe como parámetro el nº de defensas, centrocampistas y delanteros. Evidentemente siempre habrá un portero, por lo que este argumento no se contempla. La función se encarga de limpiar los campos (jugadores alineados en ese momento) y crear mediante DOM tantas listas desplegables por posición como se hayan indicado en los argumentos. Sin dejar de lado la posterior carga de jugadores en cada una de estas listas desplegables.

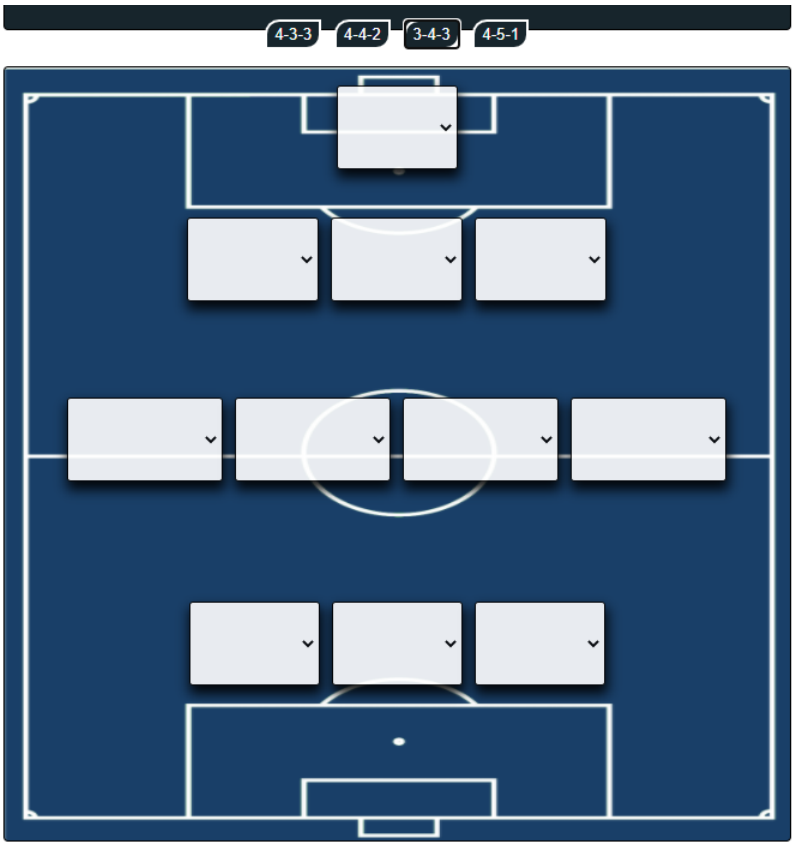


Ilustración 34. Estado de la alineación en el momento que se cambia de táctica.

Se ha capturado una imagen que muestra justo el instante posterior a cambiar de la táctica 4-4-2 a la táctica 3-4-3. Se pueden observar las listas desplegables vacías y que efectivamente, se han creado tantas listas desplegables como requiere la táctica seleccionada.

- **Selección de jugadores para cada posición.**
  Aunque esta tarea se contemplaba de forma independiente en un inicio, se ha ido desarrollando junto a los procesos anteriores, por lo que, llegados a este punto, prácticamente ya ha quedado descrito su funcionamiento.

  La finalidad de esta tarea era lograr cargar en cada una de las listas desplegables tantas opciones como nº de jugadores tiene el usuario en cada posición, teniendo en cuenta que, al seleccionar un jugador en una lista, este deberá desaparecer de las listas adyacentes.

  El logro de esta tarea ha sido posible principalmente gracias a un uso intensivo de DOM para poder moldear la estructura html en tiempo real. Así pues, quedan definidas 4 capas horizontales distribuidas verticalmente de tal manera que arriba del todo quede el portero y seguidamente los defensas, centrocampistas y delanteros. Para dar un toque inmersivo se ha impreso un campo de futbol en el fondo, eso sí, con tonalidad azulada para que no se pierda la gama utilizada a lo largo del desarrollo en el resto de los elementos.

- **Sistema de guardado/reinicio de alineación.**
  Este punto es esencial para que los cambios realizados queden almacenados y se puedan recuperar cada vez que acceda el usuario a su cuenta. Hasta el momento, todo aquello que se ha trabajado funciona, pero no queda almacenado, por lo que un cambio de táctica, o un cambio de jugador en la alineación se perderá al recargar la página, o bien, cuando el usuario cierre sesión y vuelva a acceder.

Ilustración 35. Botones de guardado y reinicio de la alineación.

  Para controlarlo se ha optado por implementar un botón que confirmará que todo cambio realizado es el definitivo. El funcionamiento es sencillo, bastará con pulsar el botón de Guardar Alineación para que una función javascript obtenga los jugadores alineados y la táctica seleccionada y que a través de Ajax se envíe esta información al servidor que procederá a insertar en la base de datos la información oportuna (nº de defensas, nº de centrocampistas, nº de delanteros, id del jugador alineado que se indicará con un 1 si lo esta y un 0 si no lo está, etc.).

  Una vez almacenada la información adecuadamente, cuando el usuario se vuelva a conectar al panel de alineación, una segunda función en javascript se encargará de consultar en la base de datos (nuevamente con Ajax) la táctica elegida (para generar en ese momento con dom las listas desplegables que hacen falta) y posteriormente los jugadores alineados (para insertarlos en cada una de estas listas). Eso sí, una vez guardada la alineación, no se podrán realizar modificaciones sobre esta a no ser que se cambie la táctica, o bien, se utilice el botón de Reiniciar tu Alineación.

El botón de Reiniciar tu Alineación sería el equivalente al desbloqueo de la alineación, es decir, permite que puedas volver a elegir jugadores para cada posición, eso sí, no se debe olvidar que, si finalmente no se guarda la alineación, los cambios no se verán reflejados, por lo que se cargará la última alineación almacenada. Técnicamente su implementación consiste en limpiar los hijos de las listas desplegables con dom y volver a cargar las listas con los jugadores disponibles del usuario en cada posición para que el usuario pueda volver a decidir.

- **Tests.**
  El proceso de pruebas ha consistido en la constante resolución de problemáticas que surgen durante la creación de las nuevas funcionalidades hasta que finalmente se ha logrado obtener el resultado previsto.

  Para ello, se ha trabajado con las herramientas de desarrollo de Google Chrome mediante las cuales se han podido identificar y subsanar errores en nuestras funciones y consultar resultados inesperados en las consultas Ajax.

9.3 Página clasificación.

Ilustración 36. Página de clasificación.

La página de clasificación no tiene funcionalidades con las que el usuario pueda interactuar. Su única finalidad es mostrar información relativa a la clasificación global y semanal a los usuarios de cada comunidad. Adicionalmente, se incorpora un panel con los resultados de los partidos de la última jornada/jornada activa en ese momento.

- **Carga de clasificación general.**
  Para la implementación de la clasificación general se realiza una consulta a la base de datos mediante php que pasará como parámetros el identificador de la comunidad en la que se encuentra el usuario.

  Puesto que la definición inicial de la tabla usuarios contempla un campo relacionado con los puntos totales, la consulta para obtener esta información será sencilla, cuya mayor importancia reside en ordenar los resultados de mayor a menor puntaje.

  Por último, a través de un bucle se irán creando tantas filas en la tabla como usuarios tenga la comunidad y sus correspondientes celdas relativas a la posición, nombre de usuario, puntaje y valor del equipo (ver ilustración 37).

Clasificación General			
Posicion	**Usuario**	**Puntos**	**Precio Equipo**
1	Test1	0	21.900.000
2	Test2	0	22.130.000
3	Test5	0	21.690.000

Ilustración 37. Tabla con la clasificación general en dicha comunidad.

- **Carga de clasificación última jornada.**
  Se sigue el mismo procedimiento que en el punto anterior dado que los puntos de un usuario en la última jornada también se contemplaron como un campo en la tabla usuario al definirlas inicialmente. Así pues, se realizará la consulta oportuna ordenada de mayor a menor puntaje y se crearán tantas filas en la tabla como miembros tenga la comunidad (ver ilustración 38).

Clasificación Última Jornada			
Posicion	**Usuario**	**Puntos**	**Precio Equipo**
1	Test1	0	21.900.000
2	Test2	0	22.130.000
3	Test5	0	21.690.000

Ilustración 38. Tabla con la clasificación de la última jornada en una comunidad con tres miembros.

- **Carga resultados partidos en vivo.**
  La idea de partida para este punto era la de utilizar algún tipo de api que permitiera extraer los resultados de los partidos en vivo, pero debido al pequeño retraso existente según la planificación establecida se ha optado por la incorporación de un widget externo facilitado por SoccerStats247.com que permiten su libre uso (ver ilustración 39). Al fin y al cabo, ambas formas ofrecen un mismo resultado, ya que el estilo del widget se puede personalizar, mientras que el tiempo dedicado para la implementación de cada una de estas posibilidades sí que dista drásticamente.

Ilustración 39. Widget externo con los resultados de la última jornada.

- **Tests.**
  Esta página no requiere un testeo intensivo, ya que su funcionalidad es meramente informativa. La única circunstancia que se ha tenido que adaptar es la de asignar puntajes manualmente en la base de datos a través de phpmyadmin a todos los usuarios con el fin de poder verificar que los usuarios se muestran correctamente ordenados dentro de la clasificación como respuesta del servidor a la consulta realizada.

9.4 Página normas.

Ilustración 40. Página de normas del juego.

La página de normas se crea con el objetivo de describir el funcionamiento del juego y pretende resolver aquellas dudas más específicas sobre cómo se comporta la aplicación en aquellas situaciones más específicas como, por ejemplo, cuando se pospone un partido de fútbol una jornada determinada, o bien, cómo se asignan los puntos a cada jugador.

- **Planteamiento y redacción de las reglas.**
  A la hora de describir el funcionamiento de la aplicación, se opta por definir una serie de categorías en las cuales incluir cada una de las normas, de tal forma que el usuario pueda localizar con más precisión dónde se puede encontrar la respuesta que resuelva su duda. Estas clasificaciones son las que se ven a continuación:

Cómo jugar.
Esta sección, que se puede observar en la ilustración 41, pretende describir brevemente en qué consiste la aplicación y presenta algunas de sus bases como bien es la definición inicial de la plantilla (15 jugadores) o el saldo inicial (20 millones).

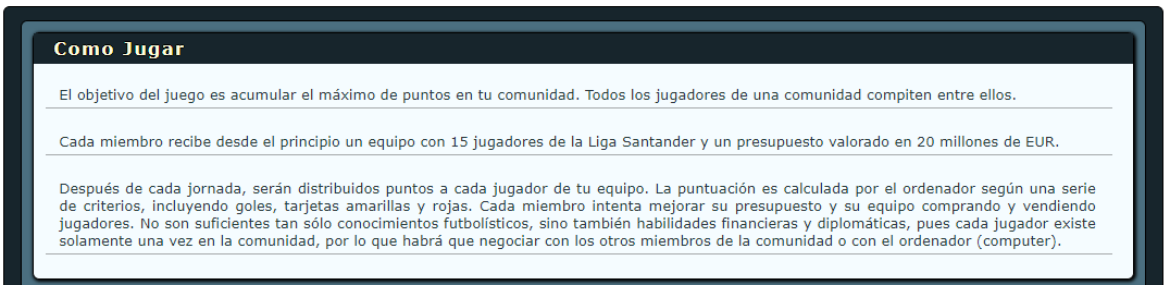


Ilustración 41. Sección normas "cómo jugar".

Normas generales.
Este apartado pretende dar respuesta a una serie de problemáticas o situaciones generales que con toda probabilidad se darán durante el transcurso de una temporada. Es posible ver más detalles en la ilustración 42.

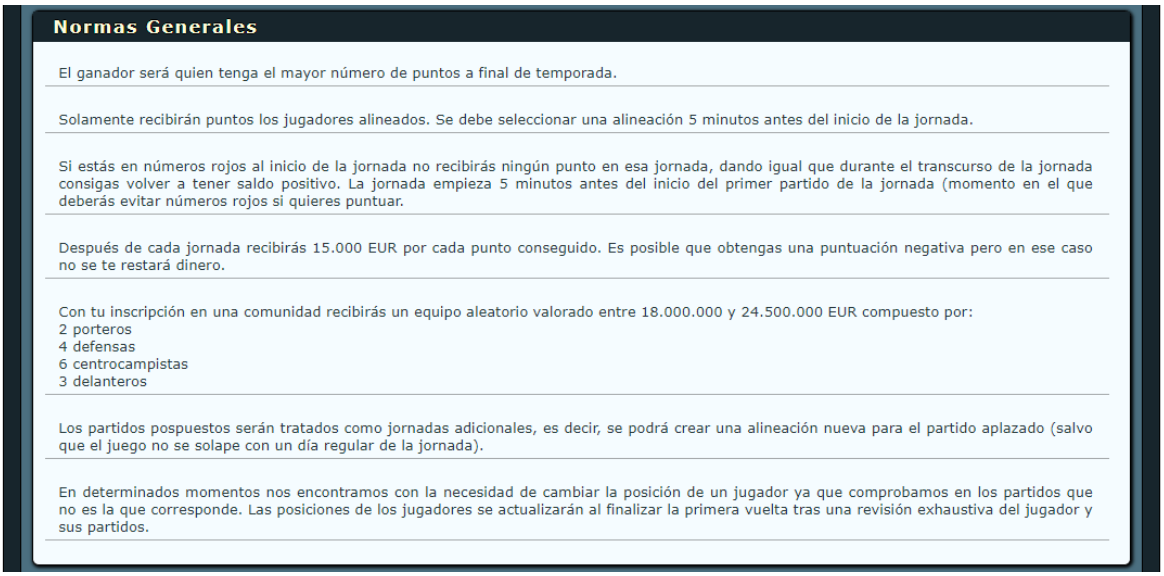


Ilustración 42. Sección normas "Normas Generales".

Puntuaciones.
Con esta sección, el objetivo será desglosar todos los parámetros por los que los jugadores de fútbol oficiales otorgarán puntos, además de la cantidad de puntos logrados por cada acción, como bien puede ser, realizar un robo de balón, o bien, la más típica, anotar un tanto. Es posible observar un breve fragmento (de una larga lista) en la siguiente ilustración.

Ilustración 43. Sección normas "Puntuaciones".

Mercado.

Este subapartado pone sobre la mesa las bases sobre las cuales se fundamenta el sistema de mercado de fichajes. Nuevamente se podrán localizar las claves de su funcionamiento con el objetivo de obtener un beneficio a partir de una buena gestión del mismo. A continuación, se muestra una ilustración relativa a su normativa.

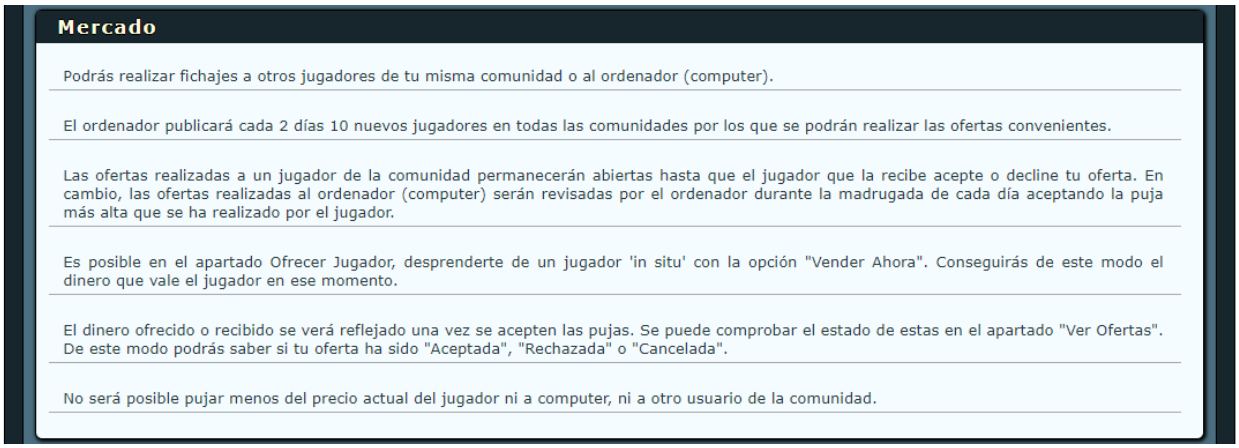


Ilustración 44. Sección normas “mercado”.

Alineación.

Se trata de la última clasificación de normas que se ha realizado. Su finalidad es resolver aspectos relativos al guardado de alineaciones con el fin de que compute correctamente para la próxima jornada que se dispute. Se considera un apartado clave para comprender cómo funciona la aplicación, ya que, de no conocer esta normativa, será sencillo incurrir en alineaciones indebidas o que no se tendrán en cuenta para la jornada. El caso más común podría ser el hecho de no alinear 11 jugadores, dejando alguna posición vacía, lo que impediría al usuario obtener puntos en esta jornada. La siguiente imagen se encarga de ilustrar lo comentado.

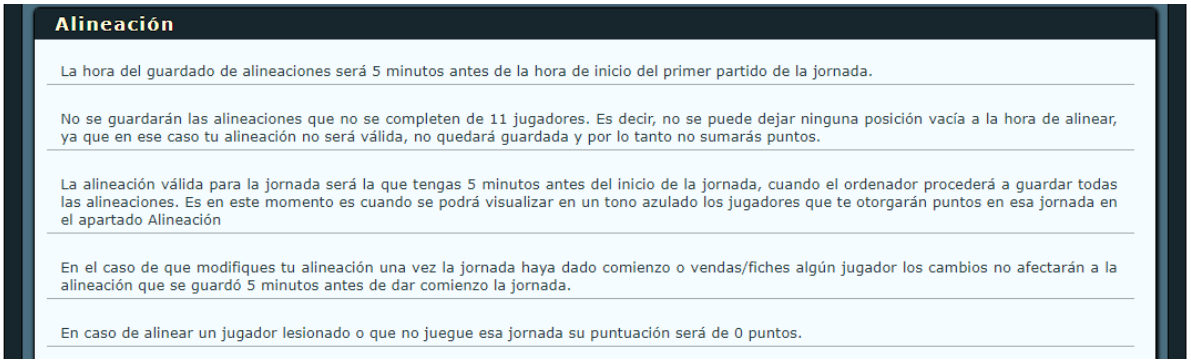


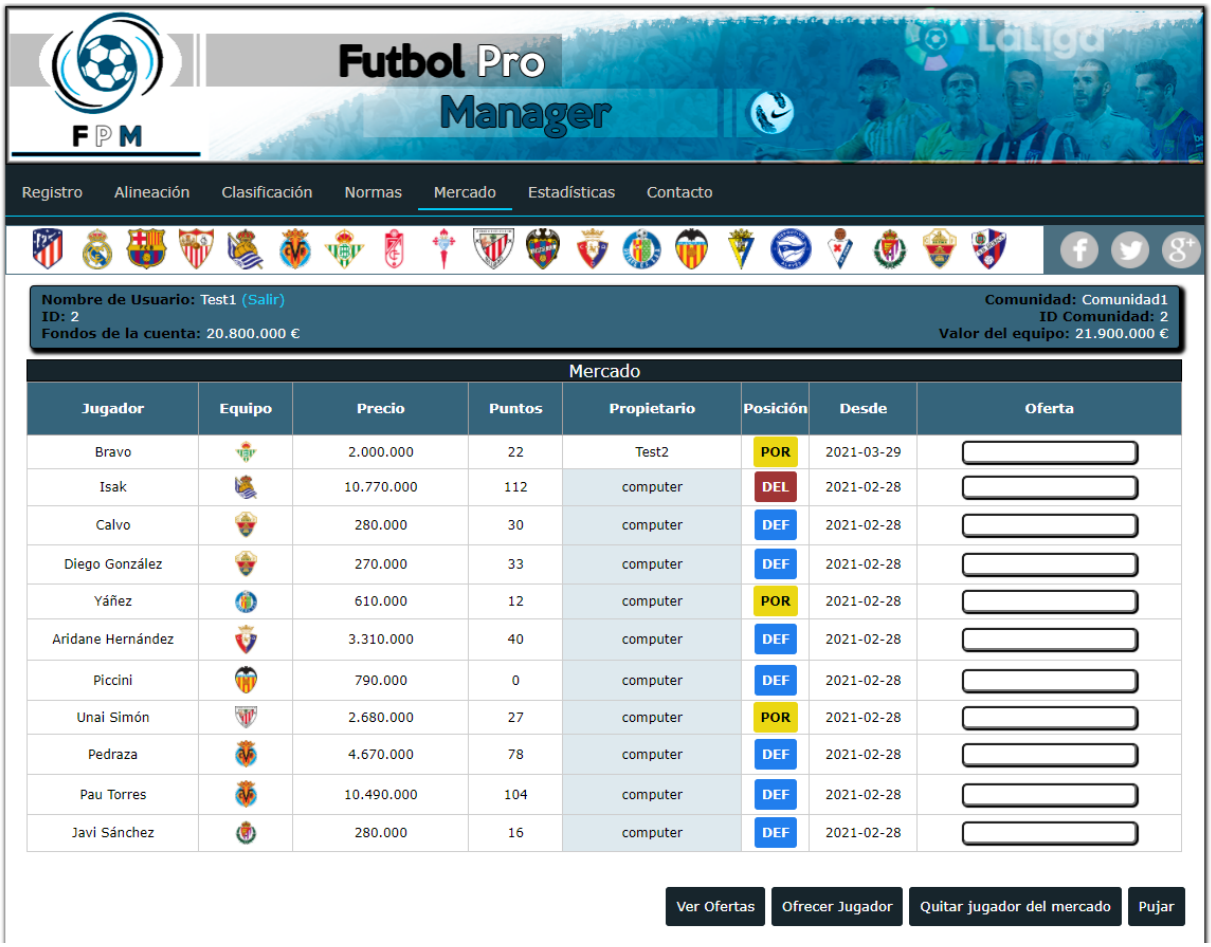
Ilustración 45. Sección normas "Alineación".

- **Paso de información a web.**
  El primer paso fue crear las normas y repartirlas entre las distintas categorías. Con esta decisión tomada y plasmada en un borrador el siguiente paso fue portarlas a la aplicación web, cuyo formato se puede verificar en las ilustraciones anteriores (de la ilustración 40 a la 45).

  La intención en este punto ha sido dejar clara la separación entre las diferentes categorías para que visualmente el usuario pueda navegar entre las distintas secciones sin que la lectura sea excesivamente densa. Con este mismo fin, se ha tratado de resumir el contenido lo máximo posible, evitando explicaciones innecesarias o que divagan.

- **Tests.**
  Se trata de una sección de la aplicación web sencilla puesto que no presenta funcionalidades dinámicas. De esta forma, las verificaciones a realizar han consistido en la corrección de errores ortográficos y aquellos detalles relativos a la disposición de los elementos para lograr la facilidad de la lectura.

9.5 Página mercado.



Jugador	Equipo	Precio	Puntos	Propietario	Posición	Desde	Oferta
Bravo		2.000.000	22	Test2	POR	2021-03-29	
Isak		10.770.000	112	computer	DEL	2021-02-28	
Calvo		280.000	30	computer	DEF	2021-02-28	
Diego González		270.000	33	computer	DEF	2021-02-28	
Yáñez		610.000	12	computer	POR	2021-02-28	
Aridane Hernández		3.310.000	40	computer	DEF	2021-02-28	
Piccini		790.000	0	computer	DEF	2021-02-28	
Unai Simón		2.680.000	27	computer	POR	2021-02-28	
Pedraza		4.670.000	78	computer	DEF	2021-02-28	
Pau Torres		10.490.000	104	computer	DEF	2021-02-28	
Javi Sánchez		280.000	16	computer	DEF	2021-02-28	

Ilustración 46. Vista de la página de mercado para un usuario en una comunidad.

En la ilustración 46 se pueden observar los detalles del mercado de jugadores en una comunidad concreta. Esta sección de la aplicación es crucial para el crecimiento de una plantilla puesto que es el punto de encuentro con otros usuarios de la comunidad, dado que permitirá intercambiar jugadores y negociar por otros aportados por la propia aplicación (mostrada como 'computer') de forma aleatoria cada dos días. Para ello, será clave gestionar el saldo de la forma más equilibrada posible.

- **Generar jugadores aleatorios para cada comunidad.**
  El primer paso para afrontar este hito ha sido conformar un código, en esta ocasión, mediante php + mysql que permitiera generar una lista de 10 jugadores libres (que no estén asignados a ningún usuario) a través de una consulta a la base de datos, para cada una de las comunidades existentes. Esto quiere decir, efectivamente, que cada comunidad tiene su propio mercado independiente, el cual deberá ser coherente con los jugadores que ya están asignados a los usuarios de cada una de las comunidades, evitando que pueda existir duplicado el mismo jugador. En resumen, un jugador puede estar libre o asignado a un usuario en cada una de las comunidades existentes.

Por otra parte, la finalidad de este código es que funcione de forma independiente y automatizada, es decir, que quede configurado como una tarea cron programada, con el objetivo de aplicar dinamismo al mercado. Se busca que cada dos días, los mercados borren los jugadores que todavía siguen libres y aporte 10 jugadores libres nuevos generados aleatoriamente.

- **Carga de dicha información en la web.**
  Una vez se ha conseguido el hito anterior se procede a implementar la interfaz a partir del bocetaje inicial para esta sección. En este, se consideró oportuno mostrar como información relativa a un jugador los parámetros; nombre, equipo, precio, puntos, propietario, posición, fecha en la que apareció en el mercado y un campo que permitirá al usuario realizar una puja para ficharlo.

Mercado

Jugador	Equipo	Precio	Puntos	Propietario	Posición	Desde	Oferta
Bravo		2.000.000	22	Test2	POR	2021-03-29	
Isak		10.770.000	112	computer	DEL	2021-02-28	
Calvo		280.000	30	computer	DEF	2021-02-28	
Diego González		270.000	33	computer	DEF	2021-02-28	
Yáñez		610.000	12	computer	POR	2021-02-28	
Aridane Hernández		3.310.000	40	computer	DEF	2021-02-28	
Piccini		790.000	0	computer	DEF	2021-02-28	
Unai Simón		2.680.000	27	computer	POR	2021-02-28	
Pedraza		4.670.000	78	computer	DEF	2021-02-28	
Pau Torres		10.490.000	104	computer	DEF	2021-02-28	
Javi Sánchez		280.000	16	computer	DEF	2021-02-28	

Ilustración 47. Vista de los datos relevantes de los jugadores activos en el mercado de fichajes.

En la ilustración 47 se puede observar la información que se ha indicado previamente en un formato de tabla simple, pero que trata de ser estéticamente atractivo para el usuario y también funcional. De hecho, la decisión de implementar el escudo del equipo, en lugar del nombre, permite una identificación mucho más rápida para el usuario, de la misma forma que colorear las posiciones de cada jugador. Así pues, la información mostrada se cree suficiente para que el usuario tenga la posibilidad de realizar un análisis inteligente, pero no demasiado extenso, a la hora de hacerse con el jugador que encaja en su plantilla.

- **Función para añadir jugadores al mercado.**
  Además de la vista que incumbe a los datos de los jugadores, este mercado incluye una serie de funcionalidades que aplicarán dinamismo sobre el mismo.
  En esta ocasión, se habla de la función 'Ofrecer Jugador' ubicada dentro de la botonera inferior derecha que se ilustra en la siguiente imagen.

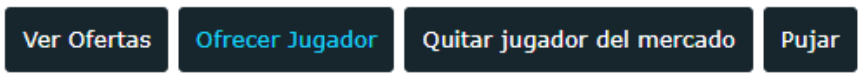


Ilustración 48. Vista de la botonera donde se ubica la función 'Ofrecer Jugador'.

El objetivo al pulsar sobre este botón será poder añadir jugadores de tu plantilla al mercado de fichajes, ya sea porque quieres deshacerte de ellos porque no aportan demasiado, porque quieres conseguir algo de saldo para salir de números rojos, o bien, porque quieres especular y tentar a los otros usuarios de la comunidad. A continuación, se muestra cómo será la vista que permitirá realizar esta operación:

Ofrece Jugador

Jugador	Equipo	Precio	Puntos	Posición	Asigna Precio
Juan Pérez		150.000	0	POR	Ofrecer Vender ya
Postigo		360.000	50	DEF	Ofrecer Vender ya
Odriozola		220.000	14	DEF	Ofrecer Vender ya
Unai García		260.000	30	DEF	Ofrecer Vender ya
Etxeita		240.000	25	DEF	Ofrecer Vender ya
John Finn Patrick		150.000	1	MED	Ofrecer Vender ya
Dani García		820.000	57	MED	Ofrecer Vender ya
Waldo Rubio		220.000	17	MED	Ofrecer Vender ya
Camarasa		180.000	0	MED	Ofrecer Vender ya
Inui		2.050.000	68	MED	Ofrecer Vender ya
Koke		7.850.000	88	MED	Ofrecer Vender ya
Kike García		7.270.000	99	DEL	Ofrecer Vender ya
Bacca		280.000	6	DEL	Ofrecer Vender ya
Fer Niño		1.850.000	38	DEL	Ofrecer Vender ya

Ilustración 49. Vista de un usuario concreto para poner a la venta jugadores.

Nuevamente se puede observar una interfaz sencilla y que va al grano. En ella se puede visualizar un listado con la plantilla de la que dispone un usuario y sus datos esenciales. Sin embargo, la funcionalidad más relevante reside en el campo que permite al usuario fijar un valor mínimo por el cual estaría dispuesto a vender a un jugador. De esta forma, pulsando el botón 'Ofrecer', se añadiría al mercado de fichajes con la intención de que cualquier otro usuario de la misma comunidad pueda verlo y realizar una oferta por él, cuyo mínimo valor será el que haya fijado el propietario. En la ilustración 47 es posible observar cómo el usuario 'Test2' ha añadido al jugador Bravo al mercado de fichajes. Esto permite al usuario activo realizarle una oferta.

- **Función para vender jugadores directamente.**
  Otra posibilidad es la de 'Vender ya' a un jugador. Su uso puede ser útil en situaciones de extrema urgencia, como puede ser el día en que da comienzo la jornada si el usuario tiene su saldo en números rojos. El

hecho de poder vender a un jugador en el momento y sin depender de que otro usuario puje por él, permite obtener ingresos inmediatamente, lo que puede cambiar tu situación financiera y evitar la penalización por comenzar una jornada en números rojos. Así pues, se añade más flexibilidad en la gestión de la economía a cambio de sacrificar jugadores.

- **Función para quitar jugadores del mercado.**
  Y como es lógico, si se permite añadir jugadores al mercado de fichajes, también se deberá facilitar la posibilidad de quitarlos de este mercado, puesto que las decisiones en el juego pueden cambiar constantemente. La siguiente ilustración muestra donde localizar esta funcionalidad dentro del mercado.

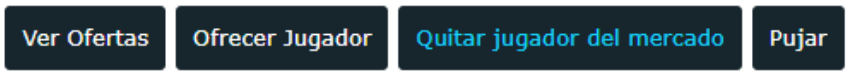


Ilustración 50. Vista de la botonera donde se ubica la función 'Quitar jugador del mercado'.

Si se pulsa sobre dicho botón se deberá aparecer un listado con aquellos jugadores que el usuario activo ha puesto a la venta. Nuevamente se visualizarán los datos más básicos asociados a estos jugadores y la posibilidad, a través de un botón 'Quitar del mercado' (ver ilustración 51), de eliminar a un jugador del listado de jugadores que se encuentran en el mercado de fichajes.

Ilustración 51. Vista del listado de jugadores puestos a la venta por un usuario.

- **Posibilidad de consultar ofertas recibidas y realizadas.**
  Otra de las funcionalidades indispensables para un usuario será la de consultar si alguien ha realizado una oferta por alguno de sus jugadores puestos a la venta, o bien, conocer cual es el estado de las ofertas que este usuario ha realizado en el mercado de fichajes. En definitiva, se trata de ofrecer al usuario un panel general que permita visualizar en qué punto de la negociación se encuentra por cada jugador.

  Para acceder a este panel, será necesario acudir nuevamente a la botonera de funcionalidades del mercado de fichajes y seleccionar la opción 'Ver Ofertas', como muestra la siguiente imagen.

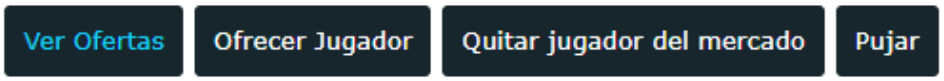


Ilustración 52. Vista de la botonera donde se ubica la función 'Ver Ofertas'.

A través de esta utilidad, se llegará al panel general de negociaciones, que permitirá tomar decisiones definitivas, tales como aceptar o rechazar una oferta, o bien, retirar una oferta realizada antes de que esta sea negociada por el otro usuario. La siguiente ilustración permite visualizar un ejemplo de esta vista.

Ilustración 53. Ejemplo de vista del panel de negociaciones.

En la imagen anterior se pueden observar dos paneles bien diferenciados.
Por una parte, el relativo a las ofertas realizadas por el usuario, donde es posible encontrar una oferta cuyo estado es 'Pendiente' de resolver, lo que significa que el usuario ha realizado una oferta, pero todavía no ha obtenido ninguna respuesta. Y en este punto, es interesante detenerse a explicar una pequeña diferenciación que existe, puesto que el usuario puede realizar ofertas a otros usuarios, o bien, a los jugadores libres que genera la aplicación en el mercado (es decir, cuyo propietario es 'computer'), como es el caso observable en la imagen 52. Si la oferta se realiza a otro usuario, este puede decidir en cualquier momento cómo finaliza la negociación (si acepta o no), mientras que, si el usuario que recibe la oferta es la propia máquina, el proceso varía ligeramente. Para resolver estos casos, se ha programado un script en php que actúa de forma automatizada (tarea cron) diariamente y se encarga de verificar cual es la oferta más alta recibida por cada jugador del mercado, decidiendo de esta manera qué usuario recibirá a este jugador. En caso de empate, la máquina otorgará el jugador al usuario cuyo valor de equipo sea superior.

Por otra parte, se localiza el panel relativo a las ofertas recibidas. En la imagen es posible ver 3 negociaciones que se encuentran en los diferentes estados posibles; la oferta por Koke todavía se está meditando, la oferta por Postigo, se ha rechazado por considerarse demasiado baja, mientras que la oferta por Roberto se ha aceptado al considerarse beneficiosa.

Respecto a la interacción del usuario con ambos paneles es muy intuitiva, ya que depende sencillamente de botones cuyo texto refleja de

forma precisa la acción que se desea ejecutar (Retirar, Aceptar, Rechazar).

- **Función para realizar ofertas.**
  Aunque de forma indirecta se ha mostrado la posibilidad de realizar esta acción (en la ilustración 47), no ha sido hasta este momento cuando se opta por explicar su funcionamiento.
  Es posible visualizar un campo que permite la escritura a la derecha de cada jugador del mercado. Como es evidente, este permitirá añadir la cuantía que está dispuesto a ofrecer un usuario para hacerse con los servicios del jugador en cuestión. Será posible hacer ofertas por tantos jugadores como se considere, pero la acción no se finalizará hasta que se pulse sobre el botón que se muestra en la siguiente ilustración.

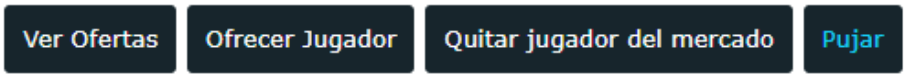


Ilustración 54. Vista de la botonera donde se ubica la función 'Pujar'.

Una vez se pulse la opción 'Pujar', la aplicación leerá todas las pujas realizadas y creará las negociaciones oportunas, que se resolverán tal y como se ha explicado en la sección anterior. Un ejemplo de vista de las pujas realizadas se puede localizar en la ilustración 53 'Ofertas Realizadas'.

- **Validación de campos.**
  Llegados a este punto, ha sido posible observar que la mayor parte de las gestiones dependen de la introducción manual de cifras por parte del usuario. Así pues, surge la necesidad de verificar que aquello que se introduce es válido y coherente con la lógica de la aplicación. Por ello, se implementan una serie de comprobaciones en lenguaje javascript que controlarán que se cumpla con las restricciones.

  En primer lugar, los jugadores en venta tienen fijado un precio mínimo por el cual se deberá pujar a la hora de hacerse con sus servicios. Así pues, se debe controlar que no existan pujas inferiores. Cuando esto ocurre, la aplicación muestra un mensaje informativo tal y como se puede ver en la siguiente imagen.

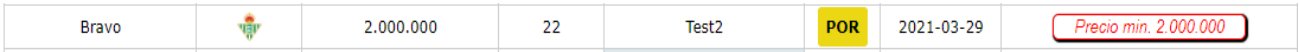


Ilustración 55. Mensaje informativo en rojo al insertar una cifra inferior al precio mínimo.

En segundo lugar, al pulsar el botón 'Pujar', solamente se deberán tener en cuenta aquellos campos en los que haya algo escrito. Así pues, se evita que la aplicación sobreentienda pujas a valor 0 por todos los jugadores del mercado. Puesto que esto es el funcionamiento lógico de la aplicación, no se muestra ningún mensaje informativo, pero a la hora de realizar las pujas, es evidente que se cumple, pues solamente se realizan las ofertas por aquellos jugadores que cumplen esta condición.

En tercer lugar, el hecho de insertar caracteres no válidos, es decir, cualquier cosa que no sean números, se comprueba, aunque en

cualquiera de estos casos, el programa devuelve cual es el valor mínimo del jugador (tal como muestra la ilustración 55), por lo que no se puede avanzar de ninguna manera introduciendo texto incoherente.

En cuarto lugar, se ha concedido la posibilidad de escribir los números con los puntos relativos a la separación de millares. Es la única excepción, ya que se considera lógica. Por lo tanto, escribir 300000, o 300.000 será aceptado.

En quinto lugar, a la hora de añadir un jugador al mercado de fichajes también deberá fijarse un precio cuyo valor deberá ser superior a su precio de mercado actual, de esta forma se impedirá que otro usuario pueda ofrecerte menos dinero de lo que realmente cuesta un jugador en un momento determinado de la temporada.

- **Función para retirar ofertas.**
  Cada usuario, podrá acceder al panel 'Ver Ofertas' en la botonera inferior derecha del mercado de fichajes para consultar las ofertas que ha realizado y también cancelar alguna de ellas antes de que se acepte con solo pulsar el botón 'Retirar' tal y como se muestra en la siguiente captura.

Ofertas Realizadas

Por	A	Equipo	Precio	Desde	Estado	Acción
Yáñez	computer		700.000	2021-05-16	Pendiente	Retirar
Diego González	computer		300.000	2021-05-16	Pendiente	Retirar
Calvo	computer		280.000	2021-05-16	Pendiente	Retirar

Ilustración 56. Vista desde la cual se puede retirar un jugador del mercado de fichajes.

- **Tests.**
  La comprobación y puesta en marcha de esta sección se ha dividido en dos fases:

  Por una parte, se han testeado cada uno de los subapartados que contiene, tratando de lograr la funcionalidad básica para cada uno de ellos, sin contemplar posibles casuísticas que comprometiesen la aplicación. Es decir, dándoles el uso lógico que se espera.
  Por otra parte, una vez se había conseguido el punto anterior, el objetivo ha sido realizar diversas transacciones entre usuarios, para verificar que el intercambio de jugadores se realizaba correctamente. Se trataba de lograr que el saldo de los usuarios se ajustaba al vender/comprar jugadores y que efectivamente los jugadores pertenecían en cada momento al usuario correcto en cada una de las situaciones (aceptar, rechazar o retirar oferta). Adicionalmente, ha sido necesario verificar que las negociaciones creadas no interfieren entre distintas comunidades creadas, sino que se realizan de forma aislada entre sí.

  Una vez se ha verificado que las consultas y los trabajos con la base de datos actuaban de la forma esperada, se ha procedido a ofrecer la aplicación a gente cercana para que le de un uso normal, localizando de

esta forma problemáticas principalmente relacionadas con la inserción de caracteres no válidos en los campos donde se realizan las pujas, o bien, en los campos que se rellenan a la hora de poner un jugador propio en venta. En este punto ha sido crucial darse cuenta de que, el hecho de añadir como válido el punto que separa los millares de los números es clave, ya que en varias ocasiones se ha tratado de implementar un valor con este formato.

Por último, se localizó una problemática que afectaba a la correcta visualización de la plantilla de un usuario que acaba de vender a un jugador. La idea es que si se vende un jugador, este desaparezca de la plantilla, pero aparecía un error mostrado por php, ya que a la hora de indexar, contabilizaba más jugadores de los que existían en la plantilla. Finalmente se modifica el código necesario para evitar esta problemática. Ahora la alineación se resetea al detectar que ha habido un cambio y es necesario pulsar sobre 'Reinicia tu alineación' para volver a confeccionarla, esta vez, sin el jugador que se acaba de vender.

9.6 Página estadísticas.

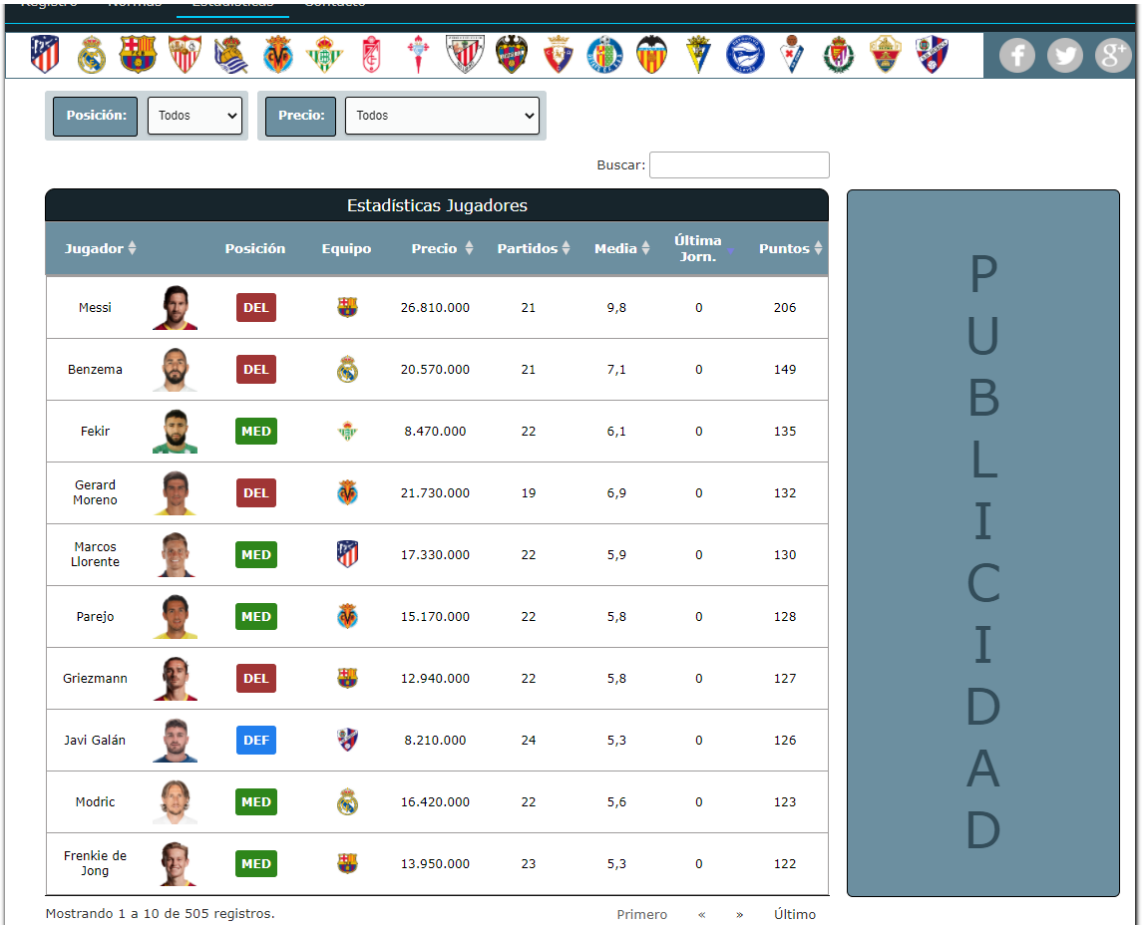


Ilustración 57. Página para consultar datos destacables de todos los jugadores.

La página de estadísticas se compone de un listado con todos los jugadores existentes en la competición y varias de sus estadísticas más relevantes que permitirán al usuario filtrar u ordenar para conocer mejor el contexto de cada jugador, pudiendo de esta manera ahorrar saldo para invertirlo en determinados jugadores el día que aparezcan libres en el mercado de fichajes, o bien, conocer quienes son los mejores jugadores en cada posición.

- **Carga mejores jugadores por posición.**
  La idea con este objetivo ha sido proponer al usuario la posibilidad de visualizar de una forma sencilla un listado de jugadores ordenados de mayor a menor puntuación obtenida y aplicar un filtrado por posición, de esta forma, se podrán realizar análisis como bien el hecho de conocer si los defensas que se encuentran en tu plantilla son de los más destacados o conviene invertir para mejorar esa posición. Aunque, en realidad, esta es solo una forma de las distintas posibilidades en las que se podría utilizar para extraer información útil a la hora de crear la mejor plantilla, siempre condicionados a la espera de que determinados jugadores aparezcan en el mercado de fichajes, por supuesto.

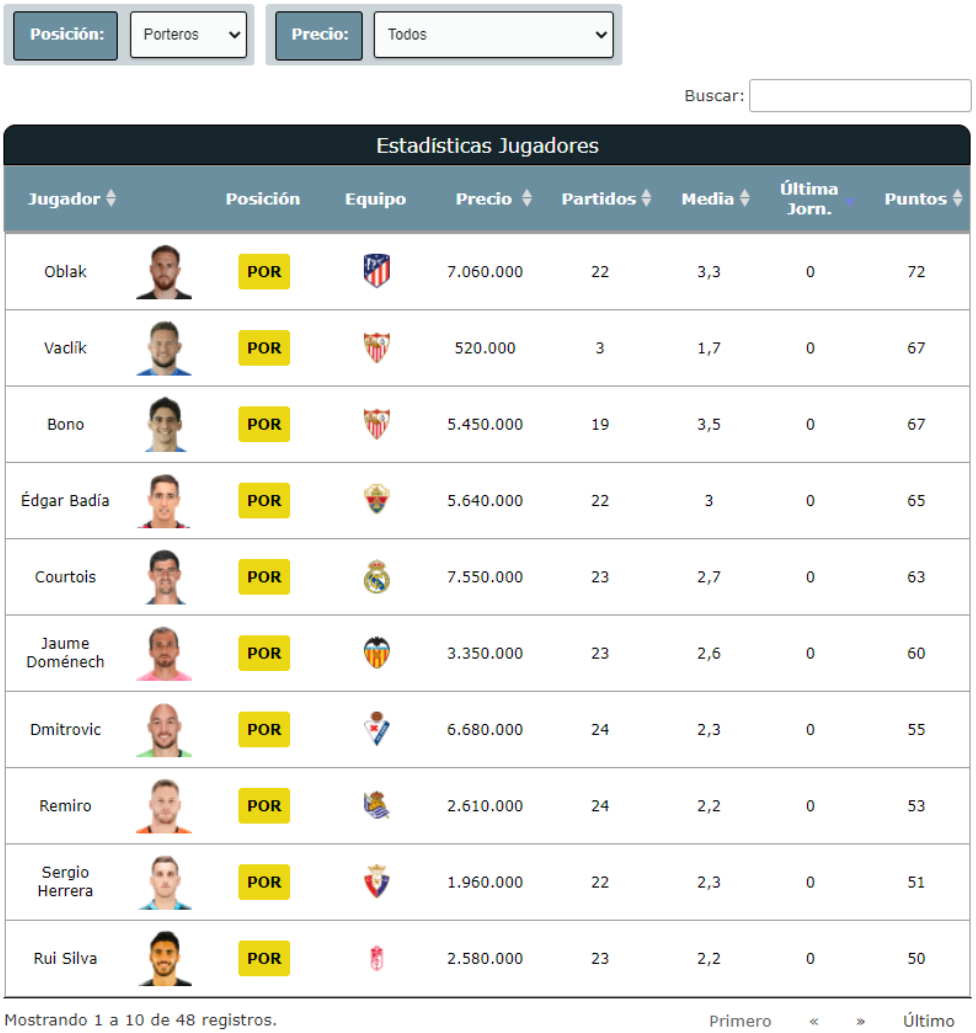
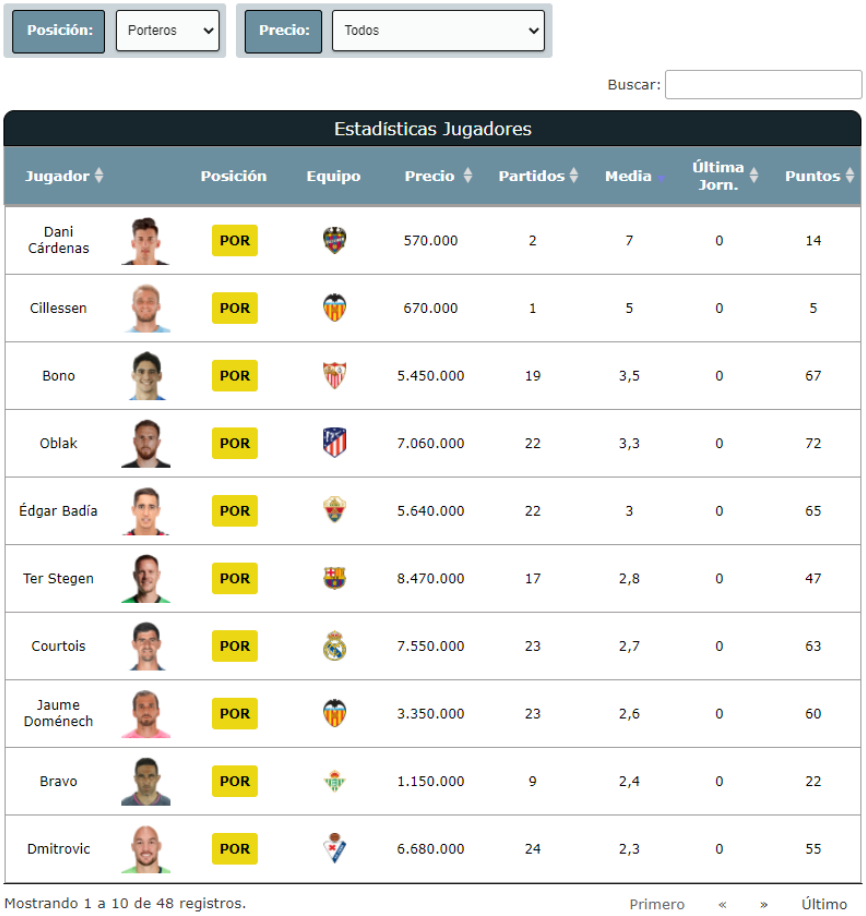


Ilustración 58. Vista de los porteros con más puntos.

La ilustración 57 permite visualizar los porteros que más puntuación han obtenido hasta el momento gracias a la aplicación del filtro que hay en la parte superior y la ordenación descendente por defecto del campo 'Puntos'. Sin embargo, esto no quiere decir necesariamente que estos sean los mejores porteros, puesto que tal vez sea más interesante echar un vistazo a los porteros que rinden mejor en relación con la cantidad de partidos jugados. Para ello, bastaría con ordenar la tabla por el campo 'Media', donde se obtendrían resultados algo diferentes y que, evidentemente, deberían interpretarse correctamente. A continuación, se puede ver un ejemplo.



Jugador	Posición	Equipo	Precio	Partidos	Media	Última Jorn.	Puntos
Dani Cárdenas	POR		570.000	2	7	0	14
Cillessen	POR		670.000	1	5	0	5
Bono	POR		5.450.000	19	3,5	0	67
Oblak	POR		7.060.000	22	3,3	0	72
Édgar Badía	POR		5.640.000	22	3	0	65
Ter Stegen	POR		8.470.000	17	2,8	0	47
Courtois	POR		7.550.000	23	2,7	0	63
Jaume Doménech	POR		3.350.000	23	2,6	0	60
Bravo	POR		1.150.000	9	2,4	0	22
Dmitrovic	POR		6.680.000	24	2,3	0	55

Mostrando 1 a 10 de 48 registros. Primero « » Último

Ilustración 59. Vista de todos los porteros con orden descendente según la media.

- **Carga de otro tipo de estadísticas.**
  Este punto inicialmente se configuró como un hito muy abierto, donde no se especificaba exactamente su finalidad. Finalmente, con la idea de adecuar el desarrollo a la temporalidad fijada en el diagrama, su materialización ha consistido en lo siguiente:

1. **Creación de filtro 'Precio'**, que permite visualizar jugadores cuyo valor queda comprendido en un intervalo determinado:

- De 0 a 1.000.000 euros.
- Entre 1.000.000 y 5.000.000 euros.
- Entre 5.000.000 y 10.000.000 euros.
- Mayor a 10.000.000 euros
- Todos (visualizar todos sin filtro por precio).

2. **Aplicación de la librería 'DataTables' para jQuery**, que permite personalizar funcionalidades para una tabla definida estáticamente. Funcionalidades tales como ordenar por columna, realizar búsquedas en un campo 'search' o paginar los resultados. Además de cargar los resultados asíncronamente a través de Ajax.

3. Se mantiene un **espacio reservado para incorporar anuncios** a la derecha de la tabla de resultados.

- **Tests.**
  En esta ocasión, las pruebas se han desglosado en las siguientes tareas:

  1. Comprobación de las consultas, verificando que devolvieran los resultados de la forma esperada.

  2. Funcionamiento asíncrono de los filtros 'Posición' y 'Precio'. Inicialmente se planificaron para funcionar recargando la totalidad de la página, pero finalmente se aplicó Ajax con el objetivo de mejorar la experiencia del usuario. En este punto, ha sido complicado detectar los errores puesto que las respuestas no aportaban información útil. Sin embargo, en la mayoría de los casos, las problemáticas eran menores (contar mal el número de columnas, llamadas a 'id' inexistentes, etc.).

  3. El aprendizaje e implementación de la librería 'DataTables' ha supuesto un reto gratificante. Esto ha permitido aplicar nuevos conocimientos a cambio de reducir la cantidad de funcionalidades que podrían existir en una sección de estadísticas. Evidentemente, esta fase ha resultado la más estresante a la hora de verificar su funcionamiento, ya que en la mayoría de casos era necesario consultar la documentación de la librería para subsanar errores.

9.7 Página contacto.

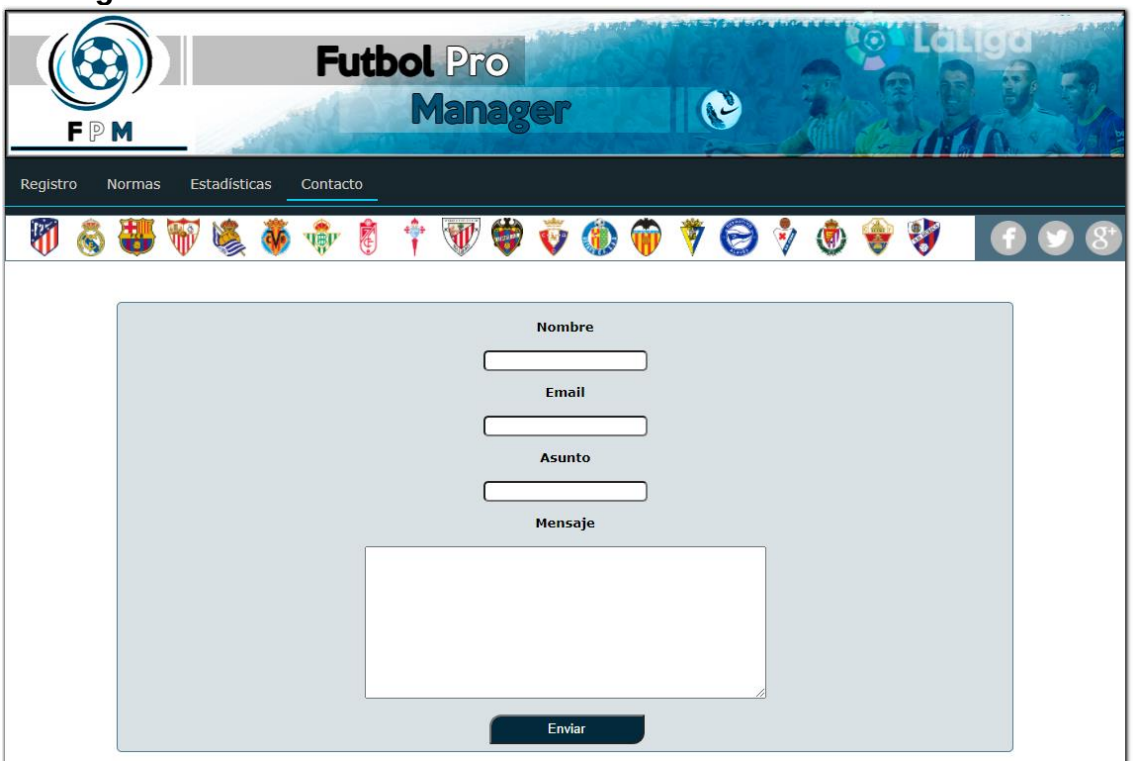


Ilustración 60. Página para contactar con el administrador del sitio web.

Esta sección pretende ser el rincón donde los usuarios puedan aportar 'feedback', además de realizar las consultas que se consideren necesarias, así como también notificar las problemáticas de la aplicación web.

- **Creación de un formulario.**
  Se busca diseñar un formulario sencillo, que pida lo mínimo para que el usuario no se sature rellenando datos. En esta ocasión se opta por solicitar un nombre de usuario, un e-mail con el fin de utilizarlo para responder a su consulta, el asunto del mensaje y finalmente un campo donde el usuario podrá redactar el texto. Es posible observar su diseño en la siguiente imagen.

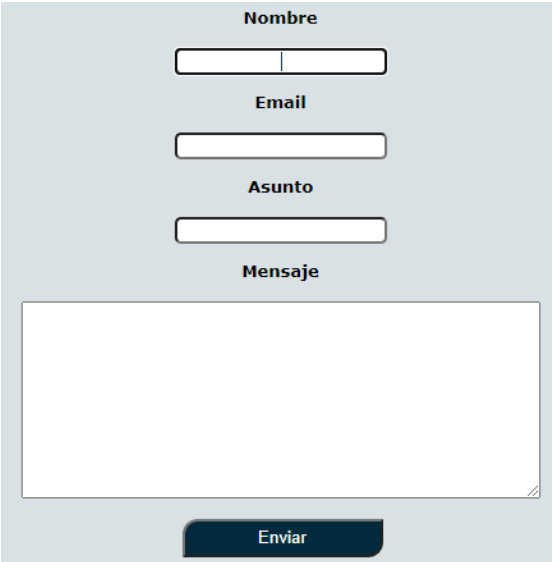


Ilustración 61. Formulario de contacto.

- **Validación de campos.**
  La validación de este formulario es sencilla. Todos los campos son obligatorios, por lo que, al pulsar el botón de envío de mensaje, se comprueba si hay contenido en todos ellos. De ser así, el mensaje se enviará (sin ningún tipo de comprobación de contenido o verificación de los datos), mientras que si no se cumple con esto, se devolverá un mensaje de error informativo.

- **Configuración para la recepción por correo.**
  Para poder realizar pruebas en local se ha optado por configurar un servidor de correo. Concretamente mediante el uso de Mercury, que se incluye en XAMPP, junto a la función mail() de php.

  Para lograr el correcto funcionamiento ha sido necesario configurar el fichero php.ini de la siguiente manera:

  SMTP = localhost
  smtp_port = 25
  sendmail_from = postmaster@localhost

  Además de la función mail, que se ha implementado de la siguiente manera:

```
$headers="";
///////Configuración/////
$mail_destinatario = 'berferlo@futbolpromanager.es';
///////Fin configuración//
if (isset ($_POST['enviar'])) {
$headers .= "Email: ".$_POST['email']."
Mensaje:
".$_POST['mensaje']."";
if ( mail ($mail_destinatario, "Nombre: ".$_POST['nombre']." ".$_POST['apellido']."", $headers )) echo '
Su mensaje a sido enviado correctamente.
';
else echo '
Error al enviar el formulario.
'; }
echo '
```

Ilustración 62. Breve fragmento del uso de la función 'mail' de php.

  De esta manera, se ha configurado un escenario que permitirá testear el envío de mensajes vía web, sin la necesidad de un servidor web en activo.

- **Tests.**
  Las pruebas de esta sección han pasado por verificar si el envío de un mensaje desde la web, se recibe en el correo configurado. Los errores que han surgido se han podido resolver con búsquedas en internet al ser bastante comunes. En cualquier caso, este aspecto es puntual, ya que la intención del correo es que funcione una vez almacenado en un servidor web real, por lo que este punto seguramente sufrirá modificaciones que se comentarán más adelante.

9.8 Página panel de administración.

Ilustración 63. Vista del panel de administración/desarrolladores.

La sección de administradores permite ejecutar acciones o probar funcionalidades con el objetivo de mejorar o hacer funcionar la aplicación como es debido. Se trata de una sección que ha ido incorporando funcionalidades según las necesidades del desarrollador, principalmente para agilizar las pruebas realizadas.

Los subobjetivos en los que se ha desglosado este hito y que coinciden (casi en su totalidad) con el número de funcionalidades que se ven en la imagen 62 son los siguientes:

- **Restringir acceso a perfiles exclusivos.**
  Como es lógico, esta sección solamente está disponible para aquellos usuarios que se definan como administradores en la base de datos. En esta ocasión, el usuario que permite el acceso es ‘computer’ y la contraseña ‘computer’. De hecho, este usuario solamente tendrá acceso a esta sección.

- **Función para aceptar las pujas realizadas (Aceptar Pujas a Computer).**
  Tal y como se ha mencionado anteriormente en la sección ‘Mercado de Fichajes’, los usuarios podrán realizar sus ofertas a jugadores sin equipo que el ordenador ha ofrecido en el mercado de fichajes (recordad que cada 2 días debe renovar este mercado). Sin embargo, así como las ofertas que van de usuario a otro usuario, se gestionan manualmente por el individuo, las ofertas dirigidas a la ‘computer’, no hay nadie que las pueda aceptar o declinar. Así pues, surge la necesidad de crear un script que permita verificar las pujas en cada comunidad y que sea capaz de determinar a qué usuario se le asigna cada jugador en base a sus pujas realizadas.

  A nivel de desarrollo, este script ha permitido testear si las pujas funcionaban según lo esperado y corregir las desviaciones. Sin embargo, se espera de este script, ser ejecutado cada 24 horas de

forma automatizada una vez quede almacenado en el servidor web. De esta forma, no se dependerá de una intervención humana que esté pendiente de ejecutar esta función.

- **Función para actualizar el mercado (Actualizar Mercado).**
  Sobre esta necesidad ya se ha hablado en diferentes puntos. Se sabe que el mercado de fichajes deberá renovarse cada dos días (por definición del propio desarrollador). Así pues, para realizar pruebas y verificar su funcionamiento, ha sido necesario ejecutar esta función manualmente en incontables ocasiones. Sin embargo, se espera que este script se convierta en una tarea programada que se ejecute en el servidor automáticamente cada 48 horas.

- **Función para guardar las alineaciones (Guardar Alineaciones).**
  Las bases del juego definen que cada usuario compite con una alineación cada jornada, pero ¿qué alineación es la que compite si se tiene en cuenta que un usuario puede realizar cambios en cualquier momento? Esta funcionalidad busca resolver esta problemática. Se ha concretado que la alineación que se tendrá en cuenta es la que tiene cada usuario justo 5 minutos antes de empezar la jornada. Así pues, esta función permite ejecutarse en cualquier momento, para obtener las alineaciones de los usuarios y almacenarlas en una tabla temporal de la base de datos. De esta forma, las modificaciones posteriores (cambios en la alineación, compra/venta de jugadores, etc.) no afectarán al funcionamiento normal de la misma. En este caso, la tarea no se puede programar a una hora determinada, puesto que el inicio de las jornadas puede variar. Sin embargo, tras analizar el transcurso de esta temporada, la hora más común es las 21.00 horas cada viernes, por lo que se programaría a las 20.55 y en determinadas excepciones, se ejecutaría manualmente.

- **Función para reiniciar saldo del usuario (Reiniciar Dinero a Usuarios).**
  A la hora de realizar operaciones en el mercado de fichajes para testear la aplicación, se desajustaba el saldo de los usuarios. La forma más eficiente para restaurar el saldo sin tener que modificarlo manualmente en la base de datos ha sido crear un script que resetea a 20.000.000 (saldo por defecto al crear un usuario) el saldo de todos los usuarios. Evidentemente, se trata de una funcionalidad para uso del desarrollador, ya que ejecutarla durante el transcurso de la temporada afectaría a todas las comunidades desvirtuando la competición.

- **Panel para otorgar puntos a los jugadores (Dar Puntos).**
  Esta sección es crucial, ya que es la que se encarga de asignar los puntos obtenidos por cada jugador en su correspondiente partido en base a las aportaciones que ha realizado. Para la ocasión se ha optado por la inserción manual de puntos a partir de las estadísticas extraídas en 'FotMob' (una app móvil que desglosa las actuaciones de los jugadores). Así pues, tras finalizar cada partido, se podrá acceder al equipo en cuestión y otorgar a cada jugador los puntos logrados, que, de forma indirecta, se sumaran a los usuarios si cuentan con alguno de

estos jugadores en sus filas. Para poder visualizar este panel (que inicialmente está oculto), se debe pulsar el botón 'Dar Puntos' y se visualizará un panel con los equipos. Tras seleccionar cualquier de estos, se desplegará un nuevo panel con todos los jugadores de dicho equipo, momento en el que se podrán completar las casillas con los puntos obtenidos. La siguiente imagen permite ilustrar lo descrito.

Ilustración 64. Paneles desplegables para asignar puntuaciones a jugadores.

Se compone de un 1er panel donde se seleccionará el equipo al cual se quiere repartir puntos, mientras que en un 2º panel y mediante el uso de Ajax se carga de forma asíncrona y se muestra de forma dinámica la plantilla que lo compone. Seguidamente solamente quedará rellenar los campos con las puntuaciones que hayan participado y pulsar sobre el botón 'Puntuar' que actualizará las puntuaciones totales de los jugadores y de los usuarios. Un ejemplo se puede ver en la ilustración 63, donde se visualiza la plantilla al completo del Barcelona.

- **Otras funcionalidades no contempladas.**
  Espacio que se reservó inicialmente para implementar funcionalidades que en dicho momento no se contemplaban, de esta forma, se reserva un rango de tiempo que probablemente irá invertido en ellas. En esta ocasión, son dos nuevos subobjetivos los que aparecieron (Reiniciar Puntos Última Jornada y Actualiza Clasificación General) y a continuación se describen.

- **Reiniciar Puntos Última Jornada.**
  Esta función se debe ejecutar al finalizar la jornada, cuando ya se han otorgado los puntos a todos los equipos, ya que, de esta forma, las tablas de las bases de datos que contienen los puntos obtenidos en la última jornada quedarán a 0, por lo que cuando se presente la siguiente jornada, los jugadores partirán de una puntuación de 0 y no de la

obtenida la semana anterior (error que ocurría antes de implementar esta funcionalidad).

- **Actualiza Clasificación General.**
  Cabe recordar que cada usuario puede visualizar dos clasificaciones: la general (total) y la semanal (la de la jornada en vivo). Aunque la clasificación semanal se irá actualizando conforme se otorguen los puntos a los equipos, la clasificación general debe esperar a que la jornada finalice para ser actualizada. Esta función se crea con esta finalidad. Pretende ser una tarea automatizada, sin embargo, la variación de horarios en cada jornada impide intuir cual es la mejor hora.

- **Tests.**
  En parte, las funcionalidades que aparecen en esta sección surgen precisamente de la idea de simular acciones para testear. Así pues, se puede decir que este panel de administración es una especie de apartado que facilita la vida al desarrollador a la par que permite dotar de funcionalidades relevantes la aplicación.

  En general, la fase de pruebas en este apartado ha consistido en verificar que las operaciones realizadas en las bases de datos trabajaban de forma consistente. Ha sido sencillo detectar errores al modificar saldos, jugadores, puntos y subsanarlos, puesto que la interfaz permite visualizar de forma sencilla los cambios realizados. Es por este motivo que la resolución de las problemáticas ha sido rápida, pero algo repetitiva debido al elevado número de consultas y acciones realizadas en la base de datos y la cantidad de tablas implicadas para cada modificación (update, insert, delete, etc.).

  Se ha profundizado especialmente en comprobar dos hitos específicos: la funcionalidad que se encarga de guardar las alineaciones de cada usuario y la funcionalidad encargada de repartir los puntos y actualizar los puntajes en la clasificación semanal, ya que se tratan de aspectos fundamentales del juego.

10. Despliegue de la aplicación

Una vez elaborada la aplicación web, suele ser habitual desplegarla en un servidor para presentarla al público. Y aunque en esta ocasión se está trabajando con un desarrollo sin ánimo de lucro y con fines académicos, se considera interesante cumplir con este procedimiento con el objetivo de interactuar con un entorno real. En definitiva, se habla de un paso que prácticamente se convierte en esencial cuando se busca monetizar una idea.

Con este fin, que ya quedó planificado en un inicio, se procede a la cumplimentación de los subobjetivos fijados y que a continuación se describen.

- **Análisis de mercado.**
  Para comenzar, se ha optado por realizar búsquedas genéricas (sobre el despliegue web) en internet con el fin de conocer qué productos son los que sobresalen, es decir, aquellos que se presentan como imprescindibles y también, aquellos productos secundarios más específicos que tratan de cubrir las necesidades más concretas de los clientes potenciales.

  En este punto, es posible localizar dos conceptos clave, como son el hosting y el dominio web. El hosting, por su parte, permitirá alojar todo el contenido relativo a la aplicación web en un servidor a priori activo las 24 horas de tal forma que todo aquello que se aloje en él pueda ser consultado en cualquier momento. El objetivo del dominio web no es otro que identificar la dirección donde se ubica el contenido almacenado, que en este caso será la aplicación web.

  Tras realizar varios análisis generales, es posible vislumbrar una gama de precios muy amplia que permite incluso iniciar el despliegue de forma gratuita, siempre teniendo en cuenta las fragilidades que tiene esta metodología. Sin embargo, si se habla de servicios que aseguran una mínima calidad, es fácil comprobar que los precios base de los productos aumentan paulatinamente según se le añaden herramientas suplementarias o que pueden aportar algún beneficio adicional. Así pues, este límite de precio, lo marcarán las necesidades y las características de la aplicación en cuestión.

- **Selección de opción adecuada.**
  En primera instancia se valora la posibilidad de realizar el despliegue de forma gratuita a través de opciones como LucusHost, 000webhost o freehosting, pero finalmente se localizan incompatibilidades con las características de este proyecto, como son, la imposibilidad de programar tareas, las limitaciones en las transferencias, o bien, las desconexiones programadas del servidor durante algunos minutos (en algunos casos horas). Por otra parte, se considera clave disponer de un servicio de atención técnica que permita resolver conflictos, un servicio que como es lógico requiere de un mínimo coste. En definitiva, se opta por una solución de pago.

De entre las posibles opciones de pago que permiten almacenar de forma amigable la aplicación se opta por analizar principalmente las más económicas, no por ser ahorrativos, sino porque tampoco tendría sentido aplicar un sobrecoste.

A la hora de decidir y con la experiencia obtenida tras reiteradas búsquedas, se opta por buscar una oferta que cubra los siguientes aspectos como mínimo:

- El coste anual que se está dispuesto a asumir es inferior a 100 euros.
- Incluir un hosting + dominio sobre SSD.
- Un dominio cuya extensión sea de las conocidas (.com, .es, etc.).
- Posibilidad de programar tareas cron.
- Compatibilidad con los lenguajes utilizados (html, css, javascript, php).
- Capacidad de almacenar mínimo 2 bases de datos (1 para otros usos).
- Posibilidad de añadir el certificado SSL.
- Almacenamiento y transferencia de datos muy superior a la que se necesita en la actualidad, con el fin de poder implementar mejoras futuras sin limitaciones.
- Soporte técnico.
- Accesos FTP.

Y con estas expectativas, se localiza el servicio Hostinet, que cumple con todo lo anteriormente especificado.

Ilustración 65. Logo Hostinet.

Así pues, se procede a la creación de una cuenta de usuario y a la compra de los productos, que aparecen inactivos inicialmente, impidiendo comenzar a trabajar con ellos. Tras 1 hora en este estado, se contacta con el servicio de asistencia (vía email) que responde con rapidez y procede a su activación, tal y como se puede ver en la siguiente imagen:

DOMINIO	TIPO	ESTADO	PRODUCTO	REGISTRO	CADUCIDAD	AUTORENOVAR
futbolpromanager.es	dominio	activo	es	18/05/2021	18/05/2022	SÍ
futbolpromanager.es	hosting	activo	SSD 3	18/05/2021	18/05/2022	SÍ

Ilustración 66. Vista de los productos contratados en hostinet. Hosting + dominio.

El siguiente paso es revisar la configuración del dominio, es decir, la información relativa al dominio (fecha inicio y fin, nombre, estado), los datos del titular (nombre, dirección, país, etc.) y la información de las dns asignadas. En este punto sería posible modificar la zona de las dns, sin embargo, no es necesario para los intereses que ocupan a la aplicación. El acceso a la web es https://futbolpromanager.es/.

- **Migración/Adaptación del proyecto de local al servidor web.**
  Esta sección se propuso inicialmente ante la perspectiva de localizar incompatibilidades y problemáticas comunes a la hora de migrar un proyecto testeado localmente hacia un servidor web real, donde las restricciones se incrementan. Seguidamente se presentan las casuísticas surgidas:

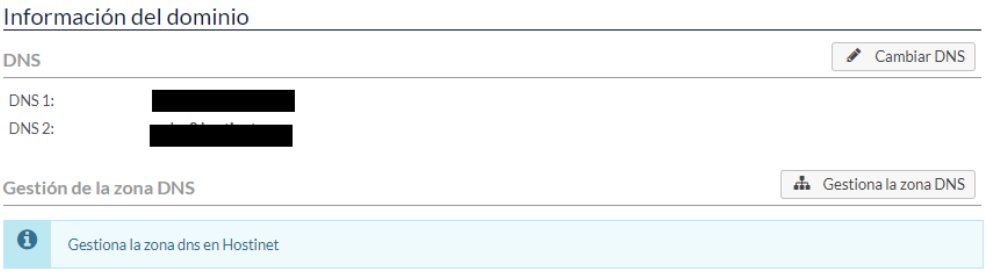


Ilustración 67. Información del dominio relativa a las dns.

1. **Instalación del certificado SSL**. Al acceder a la aplicación en este nuevo entorno se muestran errores y advertencias relativos a la falta de seguridad. El principal y más evidente es el que presenta Google Chrome y que queda ejemplificado con la siguiente imagen:

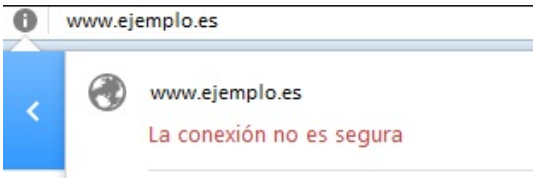


Ilustración 68. Mensaje informativo representativo de conexión no segura.

El hosting contratado ha permitido poner solución fácilmente para esta problemática, mediante la instalación del certificado SSL gratuito que proporcionan. De esta manera se logra aportar una conexión segura tal y como se visualiza en la siguiente imagen:

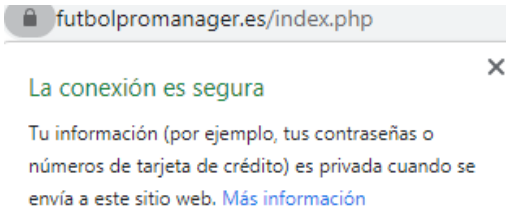


Ilustración 69. Mensaje de conexión segura obtenido tras instalar el certificado ssl.

2. **Configuración de favicon**. El primer mensaje que se muestra en la consola indica que en la ruta que se traza en el servidor web por defecto no se localiza un favicon. Así pues, se procede a la creación

del icono y a la incorporación de este en la ruta correcta, que no es otra que la propia raíz del proyecto */public_html/favicon.ico*. La verificación de que todo está correcto se puede observar en la siguiente imagen, donde se puede comprobar que ya aparece el icono de la aplicación en las pestañas del navegador:

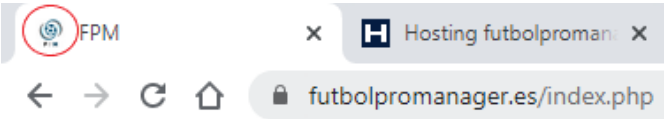


Ilustración 70. Favicon de la aplicación en una pestaña de google chrome.

3. **Creación de base de datos**. El hosting contratado permite la creación de la base de datos de forma manual o asistida. Se opta por la segunda opción que consiste en completar los campos que se solicitan. Para la ocasión, será necesario configurar el nombre de la misma y adicionalmente un usuario y los permisos que este tendrá sobre la base de datos. En ambos casos, se aplicará un prefijo, correspondiente al nombre de usuario del hosting, que permitirá que la nomenclatura sea única y no se pueda solapar con la de otros usuarios de hostinet. Finalmente, la configuración queda definida de la siguiente manera:

Base de datos	Tamaño	Usuarios con privilegio
jzscphpd_fpm	384 KB	jzscphpd_develop

Ilustración 71. Nombre de la base de datos y usuario con todos los privilegios.

4. **Exportación/Importación de la base de datos.** Se realiza la exportación de los datos contenidos en la base de datos local para importarlos en la base de datos recién creada en el servidor. Esta acción se completa sin incidencias. A continuación, se visualiza la estructura mediante phpmyadmin (del servidor):

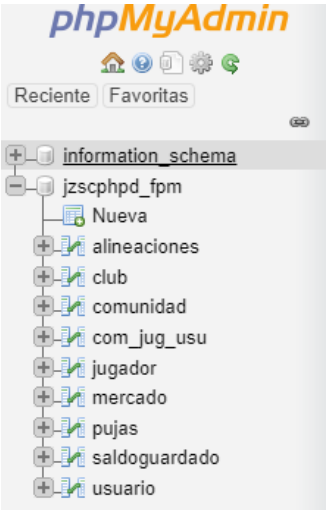


Ilustración 72. Base de datos cargada en hostinet.

5. **Configuración del servidor en Dreamweaver**. En este punto es necesario crear una copia del proyecto que existe en local sobre la cual se realizarán las modificaciones necesarias que permitan adaptarse a las necesidades de un nuevo contexto. Así pues, dentro de Adobe Dreamweaver se configura la conexión que permitirá la transferencia FTP (ver ilustración 72) de forma que, al realizar modificaciones sobre los ficheros, estos puedan colgarse en el servidor de forma sencilla. De esta manera, se agilizan las comprobaciones.

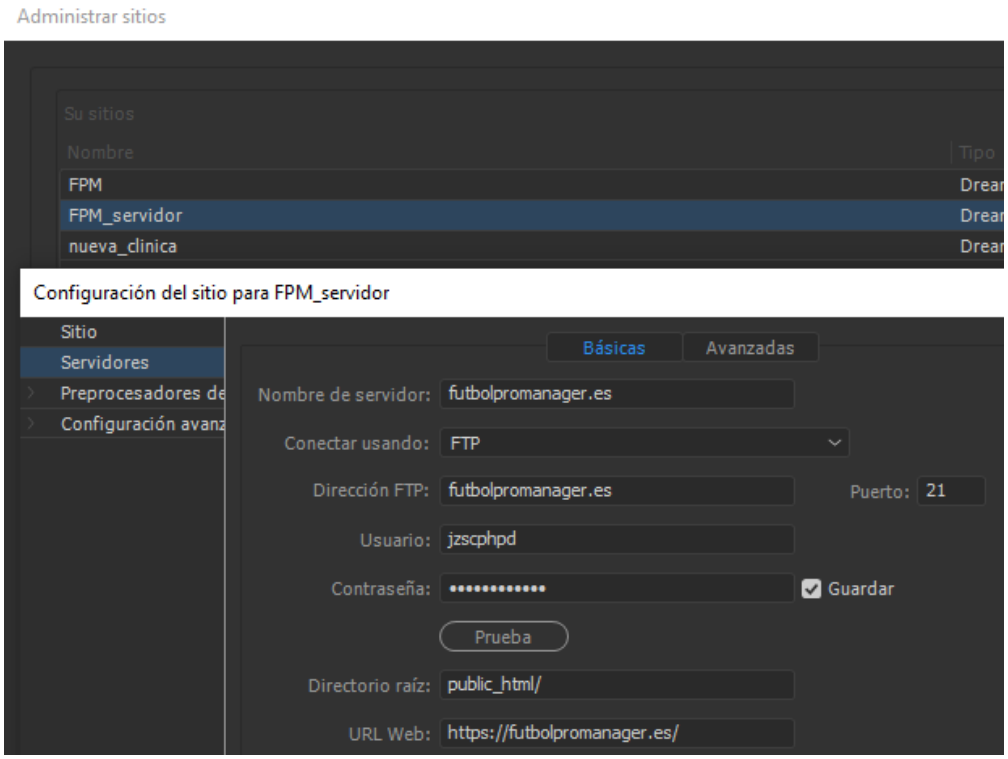


Ilustración 73. Configuración de sitio y del servidor en dreamweaver.

6. **Adaptación de código**. El cambio de contexto para el proyecto requiere de la adaptación de varias líneas del código desarrollado, especialmente si se habla de las referencias que se diseñan para funcionar localmente.

   El primer cambio se realiza en la conexión a la base de datos, que ahora deberá hacer referencia a los datos del nuevo servidor en lugar de a localhost, como se puede ver en la siguiente imagen.

```
$servidor="ehost6023.hostinet.com";
$usuario="jzscphpd_develop";
$pass="[redacted]
$db="jzscphpd_fpm";
$conn=new mysqli($servidor,$usuario,$pass,$db);
$acentos=$conn->query("SET NAMES 'utf8'");
```

Ilustración 74. Modificación de los datos de conexión.

Otra modificación se encuentra a la hora de realizar las consultas a la nueva base de datos, ya que, en local, el nombre de la base de datos es ‘lpm’, mientras que en el servidor pasa a ser ‘jzscphpd_fpm’.

También ha sido necesario adaptarse al nivel estricto de coherencia que se solicita en lo referente a la diferenciación entre mayúsculas y minúsculas. Mientras que el servidor local era más permisivo y no diferenciaba la mayoría de casos, el servidor web detecta diferencias que implican errores en el funcionamiento de la aplicación. Por este motivo, se ha requerido de una tarea de corrección y adaptación a esta nueva circunstancia que sin duda queda justificada.

7. **Modificación del widget ‘Resultados de la jornada’**. En la sección ‘Clasificación’ de la web, también es posible localizar un widget donde se muestran los resultados de la última (o la próxima) jornada de la liga y que hasta ahora dependía de la web ‘soccerstats247.com’. Sin embargo, el servidor web no acepta su funcionamiento, por lo que se opta por aplicar un widget compatible como es el que proporciona ‘El País’.

Ilustración 75. Widget en local.

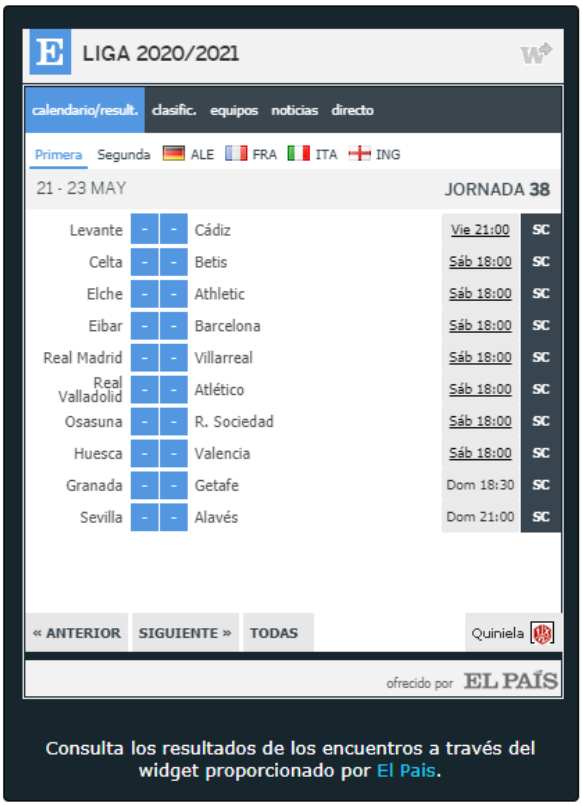


Ilustración 76. Widget en el servidor web.

8. **Formulario de contacto**. Cabe recordar en este punto que para lograr que funcionase el formulario de contacto en local, era necesaria una configuración adicional en el fichero de configuración de php (php.ini). Sin embargo, trabajar con un servidor web real aporta ciertas ventajas. Una de ellas es la posibilidad de crear una cuenta de correo asociada al dominio. Además de esto, no se necesitará de ninguna configuración adicional para lograr un funcionamiento básico del mismo mediante la funcionalidad ‘mail’ que proporciona php. Sin embargo, sí que ha sido importante diseñar el formato del mensaje a la hora de recibirlo y que finalmente ha quedado como muestra este mensaje resultante de una de las pruebas:

Error en jugador

From jzscphpd@ehost6023.hostinet.com on 2021-05-20 18:35

Detalles

```
Email del usuario: rita@prueba.com
Mensaje:

      Buenos días.

Acabo de darme cuenta que el jugador El Yamiq no muestra el logo de su equipo, mientras que el resto sí.

Espero que os sea de ayuda.

Un saludo.
```

Ilustración 77. Formato de correo tras envío desde el formulario de contacto en la aplicación.

9. **Tareas programadas (cron)**. Una vez se verifica que la aplicación se comporta razonablemente bien en el nuevo entorno llega el momento de implementar una facilidad que el servidor web proporciona, como es la de programar la ejecución de tareas (en este caso scripts php) para que la funcionalidad de la web no dependa exclusivamente del factor humano. Así, tareas tales como la ejecución del script que se encarga de actualizar el mercado de fichajes cada dos días, o el script que se encarga de resolver las ofertas que realizan los diferentes usuarios sobre jugadores libres que aporta 'la máquina' cada dos días, pasarán a ser automatizadas. Este es uno de los beneficios de disponer de un servidor funcional las 24 horas. A continuación, se muestra el listado de tareas programadas actualmente.

Trabajos de cron actuales

Minuto	Hora	Día	Mes	Día de la semana	Comando
55	6	*	*	*	/usr/local/bin/php /home/jzscphpd/public_html/cron/pujasComp.php
0	7	*/2	*	*	/usr/local/bin/php /home/jzscphpd/public_html/cron/actualizaMercados.php
55	20	21,28	*	*	/usr/local/bin/php /home/jzscphpd/public_html/cron/guardaAlineaciones.php

Ilustración 78. Listado de tareas cron que aplican automatismo a la aplicación.

10. **Redirección automática hacia https**. Aunque inicialmente se instaló el certificado SSL para aplicar seguridad a la conexión con la aplicación, se requiere de una redirección forzada hacia este protocolo, ya que de forma manual sería posible acceder a través de la conexión no segura (http). Para esto, ha sido necesario crear el fichero '.htaccess' con las siguientes líneas:

.htaccess (ASCII text, with CRLF line terminators)

```
RewriteEngine On
RewriteBase /
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://www.futbolpromanager.es/$1 [R,L]
```

Ilustración 79. Configuración que fuerza la redirección hacia HTTPS en todas las secciones de la web.

❖ **Tests.**

Ahora que la aplicación ya dispone de las funcionalidades previstas se invita al lector a echar un vistazo y probarla a través de *https://futbolpromanager.es/*.

Si bien, conviene indicar que la aplicación actualmente no es responsiva, siendo esta una de las mejoras previstas que más fuerza cobran a la hora de mejorar la aplicación en un futuro.

La aplicación se ha testeado en dos monitores, uno de 23.6” y otro de 21.5”, ambos con la resolución configurada a 1920 x 1080 y a través del navegador Google Chrome, Edge y Firefox. Se indica como referencia para disfrutar correctamente del uso de la aplicación.

Para probar, el lector puede crearse una nueva cuenta, o bien podrá identificarse con cualquiera de los usuarios existentes para realizar pruebas y cuyos datos son los siguiente: Usuario: *Test***x**, Contraseña: *Test***x**, donde **x** es igual a **un número -del 1 al 12**-. Mientras que para acceder al panel de administrador se utilizarán los siguientes datos: Usuario: *computer*, Clave: *computer*. Los datos se proporcionan con la intención de facilitar el acceso y realización de pruebas a quien lo desee, esperando que se le de buen uso y confiando en la buena fe del individuo.

11. Conclusiones

El desarrollo del proyecto ha permitido poner en práctica los conocimientos adquiridos dentro de un entorno real. En este contexto, ha sido necesario trabajar con herramientas profesionales (DreamWeaver, Photoshop, suite Xampp,etc.), definir el alcance de la aplicación y fijar unos objetivos que se ajustaran a un plazo de tiempo preestablecido.

Así pues, en esta situación, se aprende que cada paso debe ser trabajado meticulosamente con la intención de realizar el mínimo número de iteraciones sobre un 'sprint'. Para ello, es necesario visualizar diferentes casos de uso que permitan anticiparse a las diferentes casuísticas que surgen en un desarrollo en el que los usuarios podrán interactuar directamente con distintas partes de la aplicación.

Respecto al alcance definido se puede concluir que ha habido coherencia en todo momento con el planteamiento realizado inicialmente, ya que los objetivos se han podido completar en mayor o menor profundidad siempre dentro de las fechas establecidas. Posiblemente, la experiencia a nivel profesional ha permitido visualizar de forma más clara cómo se debía repartir la faena. Así pues, en este aspecto no ha habido incidencias reseñables.

Un aspecto importante para lograr los objetivos ha sido la implementación de la metodología Scrum. Su representación se ve reflejada en la división (y subdivisión) de los objetivos en pequeños proyectos interconectados. El hecho de pode regresar a cualquiera de estos proyectos para corregir problemáticas o desarrollarlos mediante la aplicación de otras alternativas que en su momento no se contemplaron ha facilitado el trabajo, especialmente debido a que la dependencia entre estos no era excesivamente alta, por lo que un error no se extendía demasiado a lo largo del proyecto, sino que se ubicaba en espacios muy concretos.

Sin embargo, el hecho de avanzar en un proyecto cuyos objetivos han sido fijados en un espacio de tiempo significativamente reducido, no ha permitido alcanzar un nivel de profundidad alto en algunos puntos. Esto ha provocado en algún momento frustración al no poder implementar funcionalidades que surgían espontáneamente, lo que llevaba al temor de no ser lo suficiente ambicioso.

No obstante, este factor también ha aportado cosas positivas, puesto que las ideas que surgían se anotaban como un depósito de posibilidades futuras que permitan ampliar y dar valor a la aplicación. Entre ellas, se pueden destacar las siguientes:

- Diseño 'responsive'. Esta es la espina que se queda clavada a la hora de realizar la entrega del trabajo. La dedicación para adaptar el diseño a las dimensiones de los dispositivos más comunes excedía el

tiempo que siendo realistas podía invertirse. Tras realizar una evaluación de los aspectos que se consideraban más adecuados a lo aprendido en la carrera, se prefiere añadir funcionalidad antes que diseño. Pese a todo, este es el primer aspecto a mejorar en el desarrollo futuro de la aplicación al ser consciente que un factor determinante puede ser precisamente la capacidad de adaptación (flexibilidad).

- Creación de redes sociales. La creación en paralelo de redes sociales (Instagram, Facebook o twitter) que permitan promocionar la aplicación.

- Fichas de clubs y jugadores. Una funcionalidad surgida de forma espontánea es la de pulsar sobre un club o un jugador de fútbol y que se redireccione a una sección con datos ampliados de estos. Por ejemplo, para consultar los puntos que ha realizado un jugador en cada una de las jornadas e incluso navegar entre distintas temporadas, logrando algo similar a un histórico.

- Panel de comunicación entre los usuarios de una comunidad. Crear un tablón donde se puedan comunicar (chat asíncrono) los usuarios dentro de sus comunidades.

- Panel de personalización de reglas de la comunidad. La posibilidad de que el administrador de cada comunidad pueda configurar una serie de reglas particulares para su comunidad (repartir premios, frecuencia de actualización de mercado de fichajes, etc.).

12. Glosario

- **Sprint**: término incluido en la metodología Scrum que refiere a los ciclos de trabajo cuya temporalidad está prefijada.

- **Jugador**: hace referencia al deportista que pertenece a un club de la liga de futbol. No confundir con el término usuario.

- **Usuario**: persona que utiliza la aplicación.

- **Comunidad**: espacio que recoge un grupo de usuarios que comparten características comunes, como es, un mismo mercado de fichajes.

- **Wireframe**: boceto que trata de representar la estructura de una aplicación web.

- **FPM**: acrónimo de Futbol Pro Manager.

- **Fantasy**: se denominan de esta forma a aquellos juegos en los que se otorgan puntos (originariamente denominados 'fantasy') a los jugadores reales de una competición. Así pues, se pueden encontrar 'fantasy' de fútbol, baloncesto, rugby, etc.

- **Back-end / Lado del servidor**: Refiere al conjunto de tecnologías que trabajan en el interior de las aplicaciones que viven en el servidor.

- **Front-end / Lado del cliente**: Refiere al conjunto de tecnologías que conforman las aplicaciones que interactúan con los usuarios, es decir, en el lado del cliente.

- **Brainstorming**: Metodología que consiste en la aportación de diversas ideas con el fin de aportar nuevas soluciones o resolver problemáticas.

- **Modelo entidad-relacion**: Herramienta que permite la representación de las entidades de una base de datos.

- **Banner**: Diseño que combina imágenes, texto y elementos interactivos con el fin de visibilizar la aplicación.

- **Computer/máquina/ordenador**: Usuario imaginario creado con el fin de aportar coherencia a la web. Por ejemplo, todo jugador que no pertenece a un usuario será propiedad de la máquina/computer. En definitiva, todas las gestiones las concluye la máquina (resolución de negociaciones en el mercado de fichajes, actualización del mercado, etc.).

13. Bibliografía

[1] *El tenis, la F1 y el baloncesto, los deportes más seguidos en España después del fútbol*. (2021, 5 de Marzo). Kantar. *https://www.kantar.com/es/inspiracion/consumidor/el-tenis-la-f1-y-el-baloncesto-los-deportes-mas-seguidos-en-espana* [25/05/2021]

[2] *Los jugadores de videojuegos en España superan ya los 15 millones*. (2020, 14 de Abril). 20minutos. https://www.20minutos.es/videojuegos/noticia/4226370/0/los-jugadores-de-videojuegos-en-espana-superan-ya-los-15-millones/ [25/05/2021]

[3] *Home*. (s. f.). Scrum.org. https://www.scrum.org/ [28/05/2021]

[4] *JavaScript | MDN*. (s. f.). MDN Web Docs. https://developer.mozilla.org/es/docs/Web/JavaScript

[5] *AJAX – Guía de Desarrollo Web | MDN*. (s. f.). MDN Web Docs. https://developer.mozilla.org/es/docs/Web/Guide/AJAX

[6] *jQuery*. (s. f.). JQuery. https://jquery.com/

[7] *¿Qué es PHP? – Manual*. (s. f.). Php. https://www.php.net/manual/es/intro-whatis.php

[8] *MySQL Documentation*. (s. f.). MySQL. https://dev.mysql.com/doc/

[9] *CSS Tutorial*. (s. f.). W3schools. https://www.w3schools.com/css/default.asp

[10] *Dreamweaver User Guide*. (s.f.). Adobe. https://helpx.adobe.com/dreamweaver/user-guide.html

[11] *Guía del usuario de Photoshop*. (s.f.). Adobe. https://helpx.adobe.com/es/photoshop/user-guide.html

[12] *Manual Datatables*. (s. f.). DataTables. https://datatables.net/manual/

[13] Apache Friends XAMPP Community and Forums. (s. f.). Apache Friends. https://www.apachefriends.org/es/community.html

Foros del web, comunidad para aprender web. (s. f.). Forosdelweb. https://www.forosdelweb.com

Desarrollo web. (s. f.). Desarrollo web. https://desarrolloweb.com/