

FUTBOLEROS

“Apasionados del futbol”

Autor: Leo Ripoll Vitini

Consultor: Albert Grau Perisé

Responsable asignatura: Santi Caballé Llobet

Grado de ingeniería informática

Área Java EE

Junio de 2019



(CC BY-NC-SA 4.0)

Esta obra se puede reutilizar siempre que se reconozca la autoría, no se emplee para fines comerciales y mantenga esta misma licencia. [Detalles](#)

Índice

1	Introducción	5
1.1	Ficha del trabajo	5
1.2	Enfoque y método seguido	5
2	Punto de vista de la empresa	6
2.1	Objetivos del proyecto	6
2.2	Análisis comparativo y justificación del proyecto	6
2.3	Descripción del proyecto	7
2.4	Análisis y definición de requisitos	7
2.4.1	Requisitos no funcionales	7
2.4.2	Requisitos funcionales	8
2.5	Casos de Uso	10
2.5.1	Actores	10
2.5.2	Diagrama de casos de uso	11
2.5.3	Especificación de casos de uso	12
2.6	Módulos / componentes principales del sistema	23
2.7	Calidad	23
2.8	Planificación temporal	23
2.9	Riesgos	25
2.10	Alcance del proyecto	25
3	Punto de vista de la información	27
4	Punto de vista de la computación	28
4.1	Diagrama de componentes	28
4.1.1	Primer refinamiento	29
4.1.2	Segundo refinamiento	30
4.1.3	Tercer refinamiento	31
4.2	Navegación y vistas	36
5	Punto de vista de la ingeniería	45
6	Punto de vista de la tecnología	46
7	Implementación	47
7.1	IDE	47
7.2	Control de Versiones	47
7.3	Comentarios sobre la implementación	48
8	Calidad del código	50
8.1	Fiabilidad del código:	51
8.2	Seguridad	53
8.3	Mantenibilidad del código	53
8.4	Duplicados	54
8.5	Líneas de código, número de métodos, comentarios...	56
8.6	Complejidad ciclomática	56
8.7	Conclusiones de calidad del código	57
9	Instalación en máquina virtual de Google	58
9.1	Producción	58
10	Pruebas de usabilidad IPO	59
10.1	Perfiles:	59
10.2	Actores:	59
10.3	Casos de uso:	59

10.4 Entrevistas	60
10.5 Conclusiones IPO	60
10.5.1 Incidencias críticas	60
10.5.2 Observaciones de usabilidad	60
10.5.3 Observaciones de diseño	62
10.5.4 Incidencias relacionadas con el navegador.....	62
11 Conclusiones del proyecto.....	63
12 Bibliografía	65
13 Glosario	66
14 Anejos	68
14.1 Plantillas de entrevistas.....	68
14.1.1 Instrucciones para usuario:.....	68
14.1.2 Anotaciones:	69
14.2 Entrevistas	69
14.2.1 Usuario 1	69
14.2.2 Usuario 2	70
14.2.3 Usuario 3	70
14.2.4 Usuario 4	71
14.2.5 Usuario 5	72
14.2.6 Usuario 6	73

Índice de ilustraciones

ILUSTRACIÓN 1 CASOS DE USO DEL SISTEMA.....	12
ILUSTRACIÓN 2 DIAGRAMA DE GANTT (I)	24
ILUSTRACIÓN 3 DIAGRAMA DE GANTT (II)	25
ILUSTRACIÓN 4 DIAGRAMA DE CLASES UML. PV INFORMACIÓN.....	27
ILUSTRACIÓN 5 DIAGRAMA DE COMPONENTES DE GRANO GRUESO.....	28
ILUSTRACIÓN 6 DEFINICIÓN DE LAS INTERFACES DEL SISTEMA	29
ILUSTRACIÓN 7 DIAGRAMA DE COMPONENTES. PRIMER REFINAMIENTO: MVC Y FACHADA.....	30
ILUSTRACIÓN 8 DIAGRAMA DE COMPONENTES. SEGUNDO REFINAMIENTO: FRONTCONTROLLER Y COMMAD.....	31
ILUSTRACIÓN 9 DIAGRAMA DE COMPONENTES DEL SUBSISTEMA DE ADMINISTRACIÓN. TERCER REFINAMIENTO: ORDENACIÓN DE COMPONENTES.....	32
ILUSTRACIÓN 10 DIAGRAMA DE COMPONENTES DEL SUBSISTEMA DE PROFILE. TERCER REFINAMIENTO: ORDENACIÓN DE COMPONENTES.....	33
ILUSTRACIÓN 11 DIAGRAMA DE COMPONENTES DEL SUBSISTEMA DE SOCIALMEDIA. TERCER REFINAMIENTO: ORDENACIÓN DE COMPONENTES.....	33
ILUSTRACIÓN 12 DIAGRAMA DE COMPONENTES DEL SUBSISTEMA DE GAME. TERCER REFINAMIENTO: ORDENACIÓN DE COMPONENTES.....	34
ILUSTRACIÓN 13 DIAGRAMA DE COMPONENTES DE LA CAPA DE INTEGRACIÓN DEL SUBSISTEMA ADMINISTRACIÓN.....	35
ILUSTRACIÓN 14 DIAGRAMA DE COMPONENTES DE LA CAPA DE INTEGRACIÓN DEL SUBSISTEMA PROFILE.....	35
ILUSTRACIÓN 15 DIAGRAMA DE COMPONENTES DE LA CAPA DE INTEGRACIÓN DEL SUBSISTEMA SOCIALMEDIA.....	35
ILUSTRACIÓN 16 DIAGRAMA DE COMPONENTES DE LA CAPA DE INTEGRACIÓN DEL SUBSISTEMA GAME.....	36
ILUSTRACIÓN 17 VISTA LOGIN	36
ILUSTRACIÓN 18 VISTA "REGISTRAR ADMINISTRADOR"	37
ILUSTRACIÓN 19 VISTA "AÑADIR EQUIPO"	37
ILUSTRACIÓN 20 VISTA "AÑADIR JUGADOR"	38
ILUSTRACIÓN 21 VISTA "MOSTRAR HISTORIAL DE ALINEACIONES".....	38
ILUSTRACIÓN 22 VISTA "CREAR ALINEACIÓN"	39
ILUSTRACIÓN 23 VISTA "REGISTRAR USUARIO"	39
ILUSTRACIÓN 24 VISTA "HISTORIAL DE USUARIO"	40
ILUSTRACIÓN 25 VISTA "BUSCAR USUARIO"	40
ILUSTRACIÓN 26 VISTA "MOSTRAR SEGUIDORES"	41
ILUSTRACIÓN 27 VISTA "MOSTRAR ENCUESTAS"	41
ILUSTRACIÓN 28 VISTA "INSERTAR ETRADA EN FORO"	42
ILUSTRACIÓN 29 VISTA "CREAR QUIZ"	42
ILUSTRACIÓN 30 VISTA "NUEVA PREGUNTA DE QUIZ"	43
ILUSTRACIÓN 31 VISTA "MOSTRAR EQUIPOS".....	43
ILUSTRACIÓN 32 VISTA "MOSTRAR ALINEACIÓN"	44
ILUSTRACIÓN 33 VISTA "CREAR ENCUESTA".....	44
ILUSTRACIÓN 34 PANTALLA DE CARGA DE ECLIPSE PHOTON	47
ILUSTRACIÓN 35 CONTROL DE VERSIONES GIT	47
ILUSTRACIÓN 36 HOSTING DE CÓDIGO PARA GIT: GITHUB.....	47
ILUSTRACIÓN 37 RESUMEN DE RESULTADOS DE SONARQUBE.....	50
ILUSTRACIÓN 38 RESUMEN DE RESULTADOS POR PAQUETES PRINCIPALES DEL SISTEMA.....	51
ILUSTRACIÓN 39 RESULTADOS PRINCIPALES DE FIABILIDAD DE CÓDIGO.....	51
ILUSTRACIÓN 40 GRÁFICA DE FIABILIDAD. NÚMERO DE LÍNEAS DE CÓDIGO - APROXIMACIÓN DEL COSTE DE ARREGLO.	52
ILUSTRACIÓN 41 BUG "EQUALS"	52
ILUSTRACIÓN 42 BUG DE ASIGNACIÓN.....	52
ILUSTRACIÓN 43 RECOMENDACIONES DE IMPLEMENTACIÓN	52

ILUSTRACIÓN 44 RESUMEN DE OBSERVACIONES DE SEGURIDAD	53
ILUSTRACIÓN 45 GRÁFICA DE SEGURIDAD. LINEAS DE CÓDIGO - TIEMPO DE ARREGLO.....	53
ILUSTRACIÓN 46 VULNERABILIDAD POR USO DE CREDENCIALES EN CRUDO	53
ILUSTRACIÓN 47 RESUMEN DE OBSERVACIONES DE MANTENIBILIDAD.....	54
ILUSTRACIÓN 48 GRÁFICA DE OBSERVACIONES DE MANTENIBILIDAD. LÍNEAS DE CÓDIGO - DEUDA TÉCNICA.....	54
ILUSTRACIÓN 49 RESUMEN DE OBSERVACIONES EN CUANTO A DUPLICADOS EN EL CÓDIGO.....	54
ILUSTRACIÓN 50 GRÁFICA DE DUPLICACIONES. LINEAS DE CÓDIGO - LÍNEAS DUPLICADAS	55
ILUSTRACIÓN 51 RECOMENDACIÓN PARA NO DUPLICAR CÓDIGO.....	55
ILUSTRACIÓN 52 RESUMEN DE OBSERVACIONES DE LÍNEAS DE CÓDIGO, COMENTARIOS, SENTENCIAS, ETC.....	56
ILUSTRACIÓN 53 GRÁFICA DE LINEAS DE CÓDIGO - LÍNEAS DUPLICADAS	56
ILUSTRACIÓN 54 RESUMEN DE COMPLEJIDAD CICLOMÁTICA	56
ILUSTRACIÓN 55 COMPLEJIDAD CICLOMÁTICA DE LOS PAQUETES DEL PRIMER NIVEL DEL PROYECTO.....	57
ILUSTRACIÓN 56 COMPLEJIDAD CICLOMÁTICA DEL PAQUETE <i>MANAGEDBEAN</i>	57
ILUSTRACIÓN 57 TIPO DE MÁQUINA INSTANCIADA EN <i>GOOGLE CLOUD COMPUTE ENGINE</i>	58

Índice de tablas

TABLA 1 FICHA DEL TRABAJO	5
TABLA 2 TRANSCURSO DEL EVENTO LOGIN	13
TABLA 3 TRANSCURSO DEL EVENTO "MOSTRAR ÚLTIMA ALINEACIÓN"	13
TABLA 4 TRANSCURSO DEL EVENTO "CREAR ALINEACIÓN PROBABLE"	14
TABLA 5 TRANSCURSO DEL EVENTO "ADMINISTRAR JUGADORES"	15
TABLA 6 TRANSCURSO DEL EVENTO "ADMINISTRAR EQUIPOS"	15
TABLA 7 TRANSCURSO DEL EVENTO "ELIMINAR ENTRADA DEL FORO"	16
TABLA 8 TRANSCURSO DEL EVENTO "REGISTRAR ADMINISTRADOR"	16
TABLA 9 TRANSCURSO DEL EVENTO "REGISTRAR USUARIO"	17
TABLA 10 TRANSCURSO DEL EVENTO "ACTUALIZAR DATOS DE USUARIO"	17
TABLA 11 TRANSCURSO DEL EVENTO "MOSTRAR HISTORIAL DE USUARIO"	18
TABLA 12 TRANSCURSO DEL EVENTO "BUSCAR USUARIO REGISTRADO"	18
TABLA 13 TRANSCURSO DEL EVENTO "SEGUIR A USUARIO"	19
TABLA 14 TRANSCURSO DEL EVENTO "CREAR ENCUESTA"	19
TABLA 15 TRANSCURSO DEL EVENTO "RELLENAR ENCUESTA"	20
TABLA 16 TRANSCURSO DEL EVENTO "ADMINISTRAR QUIZZ"	20
TABLA 17 TRANSCURSO DEL EVENTO "REALIZAR QUIZZ"	21
TABLA 18 TRANSCURSO DEL EVENTO "CREAR ENTRADA DE FORO"	22
TABLA 19 TRANSCURSO DEL EVENTO "RESPONDER A ENTRADA DE FORO"	22
TABLA 20 FICHA DE LA ENTREVISTA AL USUARIO 1	69
TABLA 21 FICHA DE LA ENTREVISTA AL USUARIO 2	70
TABLA 22 FICHA DE LA ENTREVISTA AL USUARIO 3	70
TABLA 23 FICHA DE LA ENTREVISTA AL USUARIO 4	71
TABLA 24 FICHA DE LA ENTREVISTA AL USUARIO 5	72
TABLA 25 FICHA DE LA ENTREVISTA AL USUARIO 6	73

1 Introducción

1.1 Ficha del trabajo

Tabla 1 Ficha del trabajo

Título del trabajo	Futboleros. Apasionados del fútbol
<i>Nombre del autor</i>	Leopoldo Ripoll Vitini
<i>Nombre consultor</i>	Albert Grau Perisé
<i>Nombre del Profesor responsable de asignatura</i>	Santi Caballé Llobet
<i>Fecha de entrega</i>	Junio 2019
<i>Titulación</i>	Grado de ingeniería informática
<i>Área de trabajo</i>	Java EE
<i>Idioma del trabajo</i>	Castellano / inglés (código)
<i>Palabras clave</i>	Red social, futbol, futbolero
<i>Resumen del trabajo</i>	
<p>El trabajo consiste en el desarrollo de un proyecto de software con el fin de crear desde cero una red social para amantes del fútbol accesible desde navegador. La idea del trabajo no es solo crear un producto de software sino reflejar gran parte de las etapas por las que pasa un proyecto de software. En concreto, mostrando los conocimientos adquiridos en las asignaturas cursadas, destacando las siguientes: Gestión de proyectos, programación web, diseño gráfico, Ingeniería de requisitos, Ingeniería de componentes de software, Interacción persona-ordenador, proyecto de desarrollo de software, y diseño de bases de datos.</p>	

1.2 Enfoque y método seguido

El proyecto se realiza siguiendo el *modelo de referencia para el procesamiento abierto y distribuido (RM-ODP)*. Este modelo describe un sistema, básicamente, a través de la especificación del mismo en 5 puntos de vista: empresa, información, computación, ingeniería y tecnología. Si bien el RM-ODP se utiliza para la descripción de sistemas y no de forma específica para la organización de un proyecto, el punto de vista de la empresa permite aglutinar los aspectos necesarios para la correcta organización del mismo. Por este motivo, el punto de vista de la empresa es más extenso, en comparación con otros puntos de vista.

El proyecto, por lo tanto, se describe entorno a estos 5 puntos de vista. Al principio de cada punto de vista, se describe brevemente qué aspectos se definen en el mismo con mayor concreción.

Las tecnologías que usará este proyecto están relacionadas con las especificaciones de Java EE, pero la elaboración del proyecto prevé, en las fases iniciales, un análisis independiente de la tecnología. Por ello, se presentarán las tecnologías utilizadas en el punto de vista de la tecnología, aunque la realidad sea que gran parte de la arquitectura estaba planteada ya en las fases iniciales.

La memoria del trabajo se cierra con las pruebas de calidad realizadas sobre el código, las pruebas IPO (interacción persona-ordenador) y las conclusiones finales.

2 Punto de vista de la empresa

El **punto de vista de la empresa** recoge aspectos relacionados con el negocio tales como objetivos del proyecto, justificación, objetivos, planificación, alcance, riesgos, requisitos, roles y casos de uso. Este punto de vista es el punto de partida a partir del cual se ha articulado y planificado todo el proyecto.

2.1 Objetivos del proyecto

Al ser un trabajo de final de carrera, se considera un objetivo fundamental **mostrar todas las fases de construcción de un proyecto de software** reflejando la aplicación de los conocimientos específicos adquiridos durante la carrera, destacando: *Gestión de proyectos, programación web, diseño gráfico, Ingeniería de requisitos, Ingeniería de componentes de software, Interacción persona-ordenador, proyecto de desarrollo de software y diseño de bases de datos.*

El objetivo principal del proyecto es **producir un producto de software** que consiste en una aplicación web creada desde cero. La aplicación se destina a aficionados del mundo del fútbol.

2.2 Análisis comparativo y justificación del proyecto

Se ha realizado un estudio analizando los diferentes portales web existentes alrededor del mundo del fútbol. Se han consultado:

- Periódicos deportivos digitales (*as.com, mundodeportivo.com, sport.es, marca.com*)
- Portales de juegos de fútbol como *comunio* o *futmondo*
- Portales de futbol “asociados” a los anteriores como *comuniazo.com* o *futbolFantasy.com*.

Tras el análisis se llega a las siguientes conclusiones:

1. No se han encontrado redes sociales para aficionados del fútbol, solo existen redes sociales para futbolistas y su promoción, no para “seguidores de futbol”. Es por ello que podría existir un hueco de mercado para el proyecto que se pretende emprender. Existen algunos proyectos indizados en google pero que no se han mantenido o materializado y no están subidos (*Fulbes.com, esFutbol.com* o *futbee.com*).
2. Una actividad muy solicitada y que muchos de los portales mencionados contienen, es la predicción de alineaciones probables. Los usuarios suelen buscar diferentes predicciones para apuestas deportivas o juegos. Ofrecer esta funcionalidad puede ser un buen punto de entrada al portal web Futboleros.
3. Otra actividad que se repite con frecuencia, es la de la realización de encuestas deportivas. Puede incluso ser interesante para ofrecer el resultado de las encuestas como servicio web.
4. Existe un gran mercado de datos deportivos en forma de servicios web tales como resultados, plantillas etc... de manera que no parece que valga la pena profundizar en esta área.
5. Los portales como *comunio* tienen una gran masa social, incluso portales asociados como *comuniazo*, que se nutre de servicios web del primero y se usa como herramienta para jugar a *comunio*. Futboleros podría entenderse como un complemento a *comunio* de diferente

tipología que *comunizo* u otros como *FutbolFantasy*. No parece buena idea ofrecer servicios parecidos a *comuniaz*/*FutbolFanasy* porque está muy desarrollado y además depende mucho de *comunio*.

Como conclusión, el proyecto *futboleros* es interesante por no estar explotado en el mercado. Es un portal que sirve de complemento a juegos y apuestas deportivas. Se detectan 2 puntos de entrada al portal interesantes, como la predicción de alineaciones probables, y la realización de encuestas deportivas. En cuanto a las encuestas deportivas, puede resultar interesante exponerlas como servicio web si la masa social es suficiente.

2.3 Descripción del proyecto

Título o nombre del proyecto: “**Futboleros. Apasionados del fútbol**”.

Futboleros es una red social destinada a aficionados del fútbol. Futboleros se puede entender como un punto de reunión donde se podrán crear y realizar encuestas, realizar quizzes, consultar alineaciones de equipos, seguir a otros usuarios, participar en un foro...

El ámbito futbolístico de la red social, se centrará en la primera división de la liga española. El idioma utilizado será el castellano para la presentación. Se usará inglés para todo lo relativo a código de programación.

Se consideran 3 tipos de usuarios para el portal: **administradores, usuarios registrados y usuarios sin registrar**. Tiene especial importancia la figura del administrador, encargado de tareas de mantenimiento continuas.

Aparte de todo lo relativo a registro, administración, seguimiento de usuarios e historial de usuarios, el sistema ofrecerá básicamente las siguientes funcionalidades:

1. Alineaciones probables. El administrador se encargará, cada jornada, de insertar una previsión de alineación para equipos de primera división. Ofrecer servicios web de alineaciones probables, también puede ser interesante, como opción adicional complementaria.
2. Encuestas. Sección destinada a encuestas de tipo futbolístico.
3. Quizz. Apartado que incluya juegos tipo “quizz” de fútbol.
4. Foro. Foro para discusiones futboleras.

Se considera interesante, también, poder crear un sistema de puntos para los usuarios, con el fin de clasificar tipos de usuario y crear el ambiente de competitividad propio del futbol. Así, se definen 5 tipos de usuarios: “Amateur”, “Canterano”, “Profesional”, “Veterano” y “Maestro futbolero”.

A continuación, se definen requisitos de forma más concreta.

2.4 Análisis y definición de requisitos

2.4.1 Requisitos no funcionales

1. La red social *Futboleros* debe ser accesible por navegador para todos los usuarios.

2. El sistema será accesible tanto desde ordenador de escritorio, como desde dispositivo móvil, por lo que deberá tener un diseño responsivo.
3. El sistema debe ser especialmente usable para usuarios, pudiendo rebajar para los administradores las exigencias de usabilidad.
4. El sistema debe ser extensible, flexible y mantenible.
5. El sistema debe ser ágil, no permitiendo esperas mayores a 1 segundo en navegaciones bajo condiciones aceptables de velocidad de conexión.
6. El sistema debe tener en cuenta la privacidad de los usuarios, el control de accesos y aspectos relacionados con la seguridad.

2.4.2 Requisitos funcionales

Usuarios no registrados

1. Los usuarios no registrados deberán poder:
 - Consultar alineaciones probables.
 - Ver resultados de encuestas.
 - Leer el foro
 - Ver Quizz
 - Registrarse en el sistema

Usuarios:

2. De los usuarios, se almacenará una id, un email, una contraseña, una puntuación y un nombre. Opcionalmente apellidos, fotografía de perfil y un equipo favorito.
3. Para registrarse, los usuarios deberán introducir un email real, se generará una contraseña y se enviará la misma a su cuenta de correo electrónico.
4. Para ingresar en el sistema, el usuario introducirá su email y una contraseña.
5. Los usuarios podrán modificar todos sus datos en su perfil.
6. La puntuación de los usuarios servirá para categorizar usuarios registrados. Los usuarios con menos de 50 puntos serán "Amateur", de 50 a 100 serán "Canterano", de 100 a 150 "Profesional", de 150 a 250 "Veterano" y los usuarios con más de 500 puntos serán "Maestro futbolero".
7. Los usuarios registrados, además de poder realizar lo mismo que los no-registrados, podrán:
 - Crear encuestas
 - Participar en Quizz
 - Abrir temas nuevos en el foro y responder a mensajes de otros usuarios.
 - Recibir puntos por crear encuestas, participar en foro o resolver quizz.
 - Buscar otros usuarios registrados a través de su nombre y/o apellidos.
 - Los usuarios podrán consultar datos de otros usuarios como puntuaciones, el resultado en los quizz y las encuestas en las que ha participado.
 - Seguir y ser seguidos por otros usuarios registrados.
 - Consultar su historial propio de encuestas / Quizz / foro.

Administradores

8. De los administradores se almacenará una id, un email, un nombre (opcional) y una contraseña.
9. Se podrán crear administradores. Solo administradores pueden crear otros administradores. Se deberá crear un administrador inicial en el sistema.
10. Los administradores podrán:
 - Crear y borrar jugadores.
 - Crear y borrar equipos.
 - Crear alineaciones probables de equipos. Solo podrán crear alineaciones cuya jornada sea posteriores o iguales a la última jornada registrada (si la hay).
 - Borrar encuestas.
 - Crear y borrar categorías o mensajes del foro.
 - Si se borra un mensaje, se debe sustituir por otro que indique el motivo de la eliminación y la fecha.
 - Añadir, modificar o eliminar quizz.
 - Al modificar un quiz, podrán añadir o borrar preguntas de un quiz.
 - Consultar las opciones de las preguntas de un quiz.

Alineaciones probables:

11. El sistema ofrecerá a usuarios tanto registrados, como no registrados, un listado de equipos y un enlace que muestre, para cada equipo, una alineación considerada “la alineación probable para la próxima jornada para el equipo referenciado”. La alineación presentada indicará la jornada que predice y una selección de 11 jugadores que previsiblemente vayan a ser titulares.
12. El sistema permitirá al administrador insertar las alineaciones probables de cada equipo. El sistema, permitirá flexibilidad para que el administrador decida cuándo insertar dicha información sin limitaciones.
13. Cada **jornada** contendrá una id, un número de jornada (dentro de una temporada, las jornadas se numeran de 1 a 42) y la fecha de la jornada (referente al día del primer partido de la jornada).
14. Las **alineaciones** tendrán una id, una fecha, 3 variables que indiquen la formación (p ej. 3-5-2) y que sumen 10. Las alineaciones estarán formadas por una selección de 11 jugadores de un mismo equipo. Se podrá escoger un portero La selección final deberá ser exactamente de 11 jugadores. Las alineaciones creadas se asociarán al administrador que la ha creado, al equipo al que pertenecen y a la jornada a la que hacen referencia.
15. De los **jugadores**, se deberá almacenar una id, el nombre, los apellidos, la posición y la posición numérica que ocupa en la última alineación que ha participado si hay alguna. Opcionalmente, una fotografía. Los jugadores se asociarán a un equipo.
16. La **posición** de un jugador podrá ser: portero, defensa, centrocampista y atacante.
17. De los **equipos** se deberá almacenar tan solo el nombre, una id y opcionalmente, una imagen (el escudo).

Encuestas:

18. Las **encuestas** estarán formadas por una id, una fecha, un título descriptivo (la pregunta de la encuesta), un listado de opciones de respuesta, un número de participantes y una fecha de creación. Se limita a 10 el número de opciones máximo y 2 el mínimo. No hay una fecha máxima de duración.

19. Un usuario solo podrá subir 2 encuestas por semana.
20. Las **opciones** de respuesta estarán formadas por una id, una descripción que defina la opción y el número de selecciones (cantidad de personas que han seleccionado esta opción de la encuesta).

Quiz:

21. Los **quizz** estarán formados por una id, una fecha de creación, una descripción (el tipo de quizz), un usuario (el creador) asociado, un conjunto de usuarios que han realizado el quizz y la puntuación máxima conseguida por cada usuario. Se considera que un usuario ha realizado un quizz cuando lo ha completado totalmente y obtenido una puntuación. Por último, se requiere un listado de preguntas para el quizz.
22. Las **preguntas** del quizz estarán formadas por una id, una descripción (el contenido de la pregunta), un listado de opciones.
23. Las opciones de las preguntas tendrán una id, podrán ser correctas o incorrectas y contendrán texto el texto de la pregunta.
24. Al responder a un quiz, a los usuarios se les mostrarán 10 preguntas de un quiz. Los quizz podrán contener más preguntas, pero solo se mostrarán al usuario 10 de ellas.
25. Al realizar un Quiz, tanto las preguntas como las opciones deberán aparecer ordenadas de forma aleatoria.
26. Los Quizz con menos de 10 preguntas no se les mostrarán a los usuarios.

Foro:

27. El foro se organizará por categorías. El administrador podrá añadir o quitar categorías. Las **categorías** estarán formadas por un título. Las categorías estarán asociadas al administrador que la creó o modificó.
28. Las categorías contendrán mensajes, que se presentarán ordenados por fecha.
29. Cada **mensaje** estará formado por una id, una descripción (contenido del mensaje), un titular (que no supere los 35 caracteres), una fecha de creación (*TimeDate*) y estará asociado al usuario que ha creado el mensaje. Un mensaje está asociado a una categoría y puede estar asociado a otro mensaje (cuando se trate de respuestas a un mensaje). No se permite la modificación de mensajes una vez creados.

Fotografías:

30. Las fotografías de jugadores, equipos o usuarios no podrán exceder los 300Kb y su extensión solo podrá ser *gif, png o jpg*.

2.5 Casos de Uso

2.5.1 Actores

- **Usuarios (U):** Forman parte de la red futboleros, participan en el sistema de puntos, pueden seguir otros usuarios registrados, crear encuestas, participar en el foro etc...
- **Usuarios sin registrar (UNR):** Para fomentar el uso del portal, el acceso a funcionalidades importantes no requiere registro. El hecho de tener que registrarse en el sistema es una "incomodidad" que provoca que algunos usuarios potenciales no visiten la página, por eso, el reclamo principal de la página web, que es obtener alineaciones probables para la

siguiente jornada, se puede realizar sin registro. De igual forma, se les permite leer el foro y leer las encuestas para fomentar que se registren si desean participar.

- **Administrador (A):** Actor fundamental para el funcionamiento de *Futboleros*. Se encarga de gestionar los datos, como jugadores, equipos, encuestas, quizz y foro, así como de la elaboración de las alineaciones probables. No pueden realizar quizz ni encuestas ni participar en el foro (solo gestionar: crear, modificar y eliminar categorías del foro y leer y eliminar mensajes inadecuados)

2.5.2 Diagrama de casos de uso

Se agrupan los casos de uso por paquetes en el diagrama para facilitar la elaboración de posteriores diagramas de componentes y ordenar funcionalidades. El listado completo de casos de uso es el siguiente:

1. Login (A, U)
2. Logout (A, U)
3. Show Line-up history (A)
4. Create Probable line-up (A)
5. Manage Players (A). Incluye crear, modificar y eliminar.
6. Manage Teams (A). Incluye crear, modificar y eliminar.
7. Register Admin (A)
8. Register User (U)
9. Update user data (U)
10. Unsubscribe/delete user (U)
11. Show user Activity history (U)
12. Search registered User (A, U)
13. Follow registered user (U)
14. Show following (U)
15. Answer forum entry (U)
16. Show polls (U, UNR)
17. Create fórum entry (U)
18. Create poll (U)
19. Fill poll (U, UNR)
20. Manage quizzes (A)
21. Do Quizz (U)
22. Participate in punctuation System (U)
23. Show probable line-up (A, U, UNR)
24. Read fórum (A, U, UNR)
25. Show teams (A, U, UNR)
26. Delete forum entry (A)
27. Manage forum category (A). Incluye crear, modificar y eliminar.

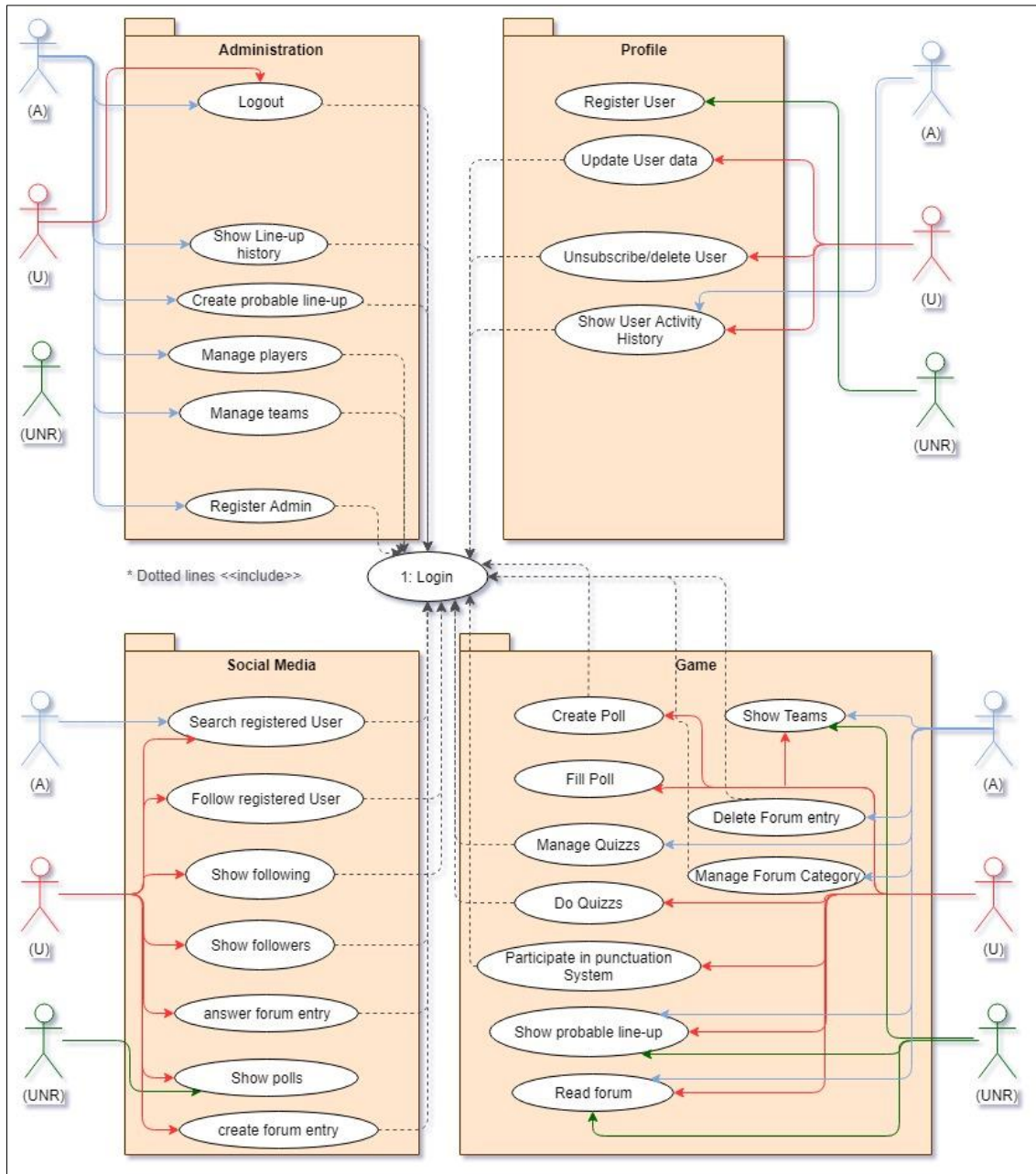


Ilustración 1 Casos de uso del sistema

2.5.3 Especificación de casos de uso

A continuación, se especifican los casos de uso más relevantes. Algunos casos de uso menos significativos se especifican con un resumen breve.

2.5.3.1 Login

Caso de uso: Ingresar al sistema, hacer *login*.

Actores: Usuarios registrados y Administradores.

Descripción: Se logeará ingresando email y contraseña, tanto usuarios como administradores.

Condición previa: El usuario o administrador debe haber sido registrado en el sistema.

Transcurso del evento:

Tabla 2 Transcurso del evento Login

Actor	Sistema
1. El actor hace clic en la opción Login	
	2. Muestra una pantalla para ingresar nombre de usuario y contraseña. Opción de recordar contraseña.
3. El usuario introduce las credenciales y hace clic en aceptar o login.	
	4. Muestra la página de inicio correspondiente al actor (admin o usuario)

Caso alternativo:

2a. El usuario accede a “recordar contraseña”

3a. El sistema informa que se ha enviado un email a su buzón con la contraseña.

3b. El usuario no introduce unas credenciales existentes en el sistema y el sistema se lo advierte. El sistema permite introducir unas nuevas.

2.5.3.2 Logout

Usuarios y administradores logados podrán hacer logout y el sistema le redirigirá a la pantalla de inicio de la aplicación.

2.5.3.3 Show Teams

Caso de uso: Mostrar equipos

Actores: Usuarios registrados, usuarios no registrados y administradores (Todos)

Descripción: Muestra un listado de equipos para conocer las alineaciones probables de estos

Condición previa: -

2.5.3.4 Show last line-up

Caso de uso: Mostrar ultima alineación probable de un equipo

Actores: Todos

Descripción: Muestra la alineación probable de un equipo

Condición previa: El equipo ha sido previamente seleccionado

Transcurso del evento:

Tabla 3 Transcurso del evento "Mostrar última alineación"

Actor	Sistema
1. Se selecciona el equipo del que se desea conocer la alineación probable prevista	
	2. El sistema muestra la alineación

Caso alternativo:

2a. El equipo no tiene una alineación probable registrada

3a. El sistema informa de esta situación y se muestra una alineación vacía.

2.5.3.5 Create Probable Line-up

Caso de uso: Crear Alineación probable

Actores: Administrador

Descripción: El sistema permitirá crear una alineación nueva o modificar la última. El sistema mostrará la última alineación registrada de un equipo (o una vacía si no hay) y permitirá al administrador cambiar los jugadores que desee. El sistema también mostrará la última jornada registrada (o cero (0) si no la hay) y permitirá modificarla.

Condición previa: El administrador debe haberse autenticado en el sistema.

Transcurso del evento:

Tabla 4 Transcurso del evento "Crear alineación probable"

Actor	Sistema
1. El administrador selecciona el equipo del que desea crear una alineación.	
	2. El sistema muestra la última alineación registrada de ese equipo.
3. El administrador hace clic en "modificar o crear nueva"	
	4. El sistema muestra la última alineación registrada del equipo y permite modificarla
5. Rellena la alineación , la jornada y formación	
	6. El sistema informa que la alineación ha sido guardada y redirige a "mostrar alineación" del equipo.

Caso alternativo:

5a. El administrador escoge un jugador que ya ha sido seleccionado.

6a. El sistema no se lo permite e informa a través de una advertencia.

5b. El administrador introduce una jornada anterior a la última registrada.

6b. El sistema no se lo permite e informa de que no se pueden modificar jornadas pasadas.

5c. El administrador introduce una formación que no suma 10

6c. El sistema el sistema se lo advierte y no permite guardar hasta subsanar la situación.

5d. El administrador deja alguna posición vacía en la alineación.

6d. El sistema advierte de que debe rellenar todas las posiciones y no hará nada, manteniendo la pantalla en espera hasta que el administrador rellene todas las posiciones.

2.5.3.6 Show line-up history

Muestra el historial de alineaciones de un equipo

2.5.3.7 Manage Players

Caso de uso: Añadir / quitar / modificar jugadores

Actores: Administrador

Descripción: Permite operaciones CRUD sobre jugadores en la capa de presentación

Condición previa: Administrador autenticado y seleccionado un equipo a editar

Transcurso del evento:

Tabla 5 Transcurso del evento "Administrar jugadores"

Actor	Sistema
1. Clic en editar jugadores	
	2. Lista a los jugadores y permite añadir, quitar o modificarlos
3. Edita jugadores (eliminación y modificación requieren confirmación)	
	4. Informa de que se ha guardado con éxito y redirige a última alineación.

Caso alternativo:

3a. Administrador añade o modifica un jugador que ya existe.

4a. El sistema no lo permite e informa del problema.

2.5.3.8 *Manage Teams*

Caso de uso: Añadir / modificar / eliminar equipos

Actores: Administrador

Descripción: operaciones CRUD sobre equipos en capa de presentación

Condición previa: Administrador autenticado y situado en "selección de equipos"

Transcurso del evento:

Tabla 6 Transcurso del evento "Administrar equipos"

Actor	Sistema
1. Clic en editar equipos	
	2. Lista los equipos, permitiendo eliminar, modificar o crear uno nuevo
3. Crea, elimina o modifica equipo	
	4. Informa que la operación se ha realizado con éxito

Caso alternativo:

3a. Crea un equipo existente o modifica a un equipo existente

4a. El sistema no lo permite e informa de ello.

2.5.3.9 *Delete Forum entry*

Caso de uso: Eliminar entrada del foro

Actores: Administrador

Descripción: Al visualizar el foro, el administrador tendrá la opción de eliminar el mensaje que desee e indicar un motivo.

Condición previa: El administrador ha accedido a una categoría del foro y ésta contiene algún mensaje.

Transcurso del evento:

Tabla 7 Transcurso del evento "Eliminar entrada del foro"

Actor	Sistema
1. Elimina una entrada del foro.	
	2. El sistema ofrece la posibilidad de indicar un motivo.
3. Indica un motivo si lo desea	
	4. Muestra la categoría del foro con un mensaje que sustituye el mensaje eliminado indicando "Mensaje eliminado" con el motivo si el administrador ha introducido alguno

2.5.3.10 Register Admin

Caso de uso: Registrar un administrador

Actores: Administrador

Descripción: Registro de un administrador en el sistema

Condición previa: Un administrador nuevo solo puede ser dado de alta por otro administrador autenticado. El sistema debe tener registrado de inicio, al menos, uno.

Transcurso del evento:

Tabla 8 Transcurso del evento "Registrar Administrador"

Actor	Sistema
1. Clic en Registrar administrador desde pantalla de inicio de administrador	
	2. Muestra un formulario donde es necesario indicar email y contraseña.
3. Ingresar los credenciales	
	4. Indica que la operación se ha realizado con éxito manteniéndose la sesión del administrador que ha registrado un nuevo administrador abierta.

Caso alternativo:

3a. El email ya existe

4a. El sistema informa que no puede usar el email introducido.

2.5.3.11 Register User

Caso de uso: Registrar un usuario

Actores: Usuario sin registrar

Descripción: Dar de alta un usuario en el sistema

Condición previa: -

Transcurso del evento:

Tabla 9 Transcurso del evento "Registrar usuario"

Actor	Sistema
1. Clic en registrar nuevo usuario	
	2. Muestra un formulario para que el usuario rellene sus datos
3. Rellena los datos, al menos los obligatorios. (email, contraseña)	
	4. El sistema informa que la operación se ha realizado con éxito y redirige a "Login"

Caso alternativo:

3a. El email ya existe

4a. El sistema informa del problema e indica que ese email ya existe.

2.5.3.12 *Update User Data*

Caso de uso: Actualizar datos de usuario

Actores: Usuarios registrados

Descripción: Permite actualizar los datos de registro del usuario

Condición previa: el usuario se ha autenticado en el sistema

Transcurso del evento:

Tabla 10 Transcurso del evento "Actualizar datos de usuario"

Actor	Sistema
1. Clic en "profile"	
	2. Muestra los datos del usuario en un formulario permitiendo modificarlos
3. Modifica los datos que desea	
	4. Informa que la operación se ha realizado con éxito y redirige a página principal.

Caso alternativo:

3a. El usuario introduce datos incompatibles (el email ya existe, por ejemplo)

4a. El sistema informa del problema y no hace nada más, esperando que el usuario modifique correctamente el formulario.

2.5.3.13 *Unsubscribe / delete User*

Caso de uso: Un usuario puede eliminar su cuenta clicando en eliminar cuenta una vez autenticado. El sistema confirma si el usuario está seguro antes de proceder. No se permite eliminar administradores.

2.5.3.14 *Show User Activity history*

Caso de uso: Mostrar el historial de actividad de un usuario

Actores: Usuarios registrados, administradores

Descripción: Al buscar usuarios y listar la búsqueda, se muestra el historial del usuario haciendo click sobre él.

Condición previa: Se ha realizado una búsqueda de usuario y se han obtenido resultados

Transcurso del evento:

Tabla 11 Transcurso del evento "Mostrar historial de usuario"

Actor	Sistema
1. Clic sobre "mostrar historial" junto a un usuario listado	
	2. Muestra el historial de ese usuario

Caso alternativo:

2.5.3.15 *Search Registered User*

Caso de uso: Buscar un usuario registrado

Actores: Administrador, usuario registrado

Descripción: Buscar un usuario registrado

Condición previa: El actor debe estar autenticado en el sistema

Transcurso del evento:

Tabla 12 Transcurso del evento "Buscar usuario registrado"

Actor	Sistema
1. Clic en "buscar usuario"	
	2. El sistema solicita que se introduzca un email o un nombre.
3. Rellena los datos	
	4. Realiza una búsqueda no estricta y muestra los resultados listados

Caso alternativo:

3a. El usuario no introduce valores de búsqueda

4a. El sistema muestra todos los usuarios ordenados por un criterio alfabético por nombre.

2.5.3.16 *Follow Registered User*

Caso de uso: Seguir a usuario registrado

Actores: Usuario registrado

Descripción: -

Condición previa: El usuario registrado se ha autenticado en el sistema y ha realizado una búsqueda obteniendo resultados.

Transcurso del evento:

Tabla 13 Transcurso del evento "Seguir a usuario"

Actor	Sistema
1. Clic sobre seguir usuario	
	2. El sistema sustituye "seguir" por "dejar de seguir" y registra el seguimiento.

Caso alternativo:

2.5.3.17 Show following

Caso de uso: Mostrar usuario que un usuario registrado sigue

Actores: Usuario registrado

Descripción: Muestra un listado de los usuarios registrados que un usuario registrado ha decidido seguir

2.5.3.18 Show Followers

Caso de uso: Mostrar quién sigue a un usuario registrado

Actores: Usuario registrado

Descripción: Muestra qué usuarios están siguiendo al usuario actualmente autenticado

2.5.3.19 Show polls

Caso de uso: Mostrar últimas encuestas creadas

Actores: Usuario registrado, usuario sin registrar

Descripción: Muestra las últimas encuestas creadas ordenadas por categoría y el resultado provisional.

2.5.3.20 Create Poll

Caso de uso: Crear una encuesta

Actores: Usuarios registrados

Descripción: Creación de encuestas

Condición previa: usuario registrado autenticado en el sistema y situado en mostrar encuestas.

Transcurso del evento:

Tabla 14 Transcurso del evento "Crear encuesta"

Actor	Sistema
1. Clic en crear encuestas	
	2. Muestra un formulario para rellenar encuesta, con título, descripción, opciones de respuesta y duración deseada (mínimo 1 mes).
3. Rellena la información	
	4. El sistema redirige a "mostrar encuestas" con la nueva encuesta añadida.

Caso alternativo:

2.5.3.21 *Fill Poll*

Caso de uso: Dar una respuesta a una encuesta

Actores: Usuarios registrados

Descripción: Permite seleccionar una de las opciones ofrecidas por el creador de la encuesta.

Condición previa: El actor ha accedido a “mostrar encuestas” y seleccionado una abierta.

Transcurso del evento:

Tabla 15 Transcurso del evento "Rellenar encuesta"

Actor	Sistema
	1. Muestra la encuesta.
2. Selecciona la opción que desea	
	3. El sistema actualiza la información relativa a contestaciones de la encuesta e informa que se ha guardado la respuesta

Caso alternativo:

2a. El usuario había rellenado con anterioridad la encuesta

3a. El sistema no le permite volver a votar e indica que ya ha votado.

2.5.3.22 *Manage Quizzes*

Caso de uso: Crear o actualizar un Quizz

Actores: Administrador

Descripción: Permite crear / actualizar o eliminar un Quizz. La eliminación será directa desde el listado de quizz previa confirmación del sistema. (No se describe la eliminación en el caso de uso)

Condición previa: administrador autenticado y ha accedido a quizz con el listado de quizz

Transcurso del evento:

Tabla 16 Transcurso del evento "Administrar Quizz"

Actor	Sistema
1. Clic en crear quiz (o modificar uno creado)	
	2. El sistema muestra el título del Quizz a modificar (o vacío si es nuevo) y el listado de preguntas del Quizz.
3. El administrador inserta el título, el número de opciones de respuesta y hace clic en añadir pregunta (o hace clic sobre una pregunta para modificarla).	
	4. Muestra una nueva pantalla para añadir pregunta (o modificarla), con título de pregunta, opciones de respuesta y un <i>checkbox</i> que indique la respuesta correcta.
5. Rellena los campos requeridos	

	6. El sistema muestra todas las preguntas añadidas o modificadas hasta el momento
7. Clic en guardar quizz	
	9. Muestra el listado de quizz con el nuevo Quizz añadido.

Caso alternativo:

8a. El administrador no ha añadido ninguna pregunta.

9a. El sistema informa que se debe introducir al menos 1 pregunta y aconseja introducir, al menos 10 preguntas.

8b. El administrador ha introducido menos de 10 preguntas y más de 1.

9b. El sistema advertirá que se aconseja introducir al menos 10 preguntas, y guarda el quizz.

2.5.3.23 *Do Quizzes*

Caso de uso: Realizar un quizz

Actores: Usuario registrado

Descripción: Permite realizar quizzes creador por un administrador

Condición previa: El usuario ha accedido a la sección de quizz y existen quizz creados.

Transcurso del evento:

Tabla 17 Transcurso del evento "Realizar Quizz"

Actor	Sistema
1. Clic sobre el Quizz que desea hacer	
	2. El sistema presenta la primera pregunta del Quizz (seleccionada de forma aleatoria)
3. El usuario selecciona la respuesta deseada y hace clic en aceptar.	
	4. El sistema muestra la segunda pregunta si la hay, y se repite el proceso un máximo de 10 veces
5. El usuario responde la última pregunta del quiz	
	6. Muestra el resultado del Quizz, contabilizando los aciertos.
7. Clic en aceptar	
	8. Redirige a "show user Activity history"

Caso alternativo:

3a. El usuario no selecciona ninguna respuesta.

4a. El sistema informa que se debe seleccionar una respuesta y espera a que el usuario lo haga para permitir avanzar a la siguiente pregunta.

2.5.3.24 *Participate in punctuation System*

Caso de uso: participar en sistema de puntuación

Actores: Usuario registrado

Descripción: Crear encuestas, participar en el foro y responder quizzes, suma puntos que acumulan los usuarios registrados para conseguir roles de futbolero.

Condición previa: estar autenticado en el sistema.

2.5.3.25 *Read Forum*

Caso de uso: Leer Foro

Actores: Administrador, usuario registrado y usuario sin registrar

Descripción: Actores pueden acceder al foro, seleccionar una categoría y visualizar los mensajes de esa categoría.

2.5.3.26 *Create Forum entry*

Caso de uso: Crear una entrada en el foro

Actores: Usuarios registrados

Descripción: Permite crear una entrada en un foro, en el interior de una categoría.

Condición previa: usuario autenticado en el sistema y seleccionado una categoría del foro

Transcurso del evento:

Tabla 18 Transcurso del evento "Crear entrada de foro"

Actor	Sistema
1. Clic en crear entrada	
	2. Muestra un formulario para rellenar título del mensaje y contenido
3. Rellena los datos	
	4. Muestra la nueva entrada en la categoría del foro.

Caso alternativo:

3a. El usuario no introduce ningún título (o contenido).

4a. El sistema informa de que se requiere un título (o contenido).

2.5.3.27 *Answer forum entry*

Caso de uso: Responder a una entrada del foro

Actores: Usuario registrado

Descripción: Los usuarios que seleccionen una entrada existente en el foro, podrán responder a esa entrada.

Condición previa: estar autenticado en el sistema, haber seleccionado una categoría del foro y un mensaje del mismo.

Transcurso del evento:

Tabla 19 Transcurso del evento "Responder a entrada de foro"

Actor	Sistema
1. Clic en responder	

	2. Muestra una pantalla como la de añadir entrada
3. Rellena los datos (obligatorio título y contenido)	
	4. Muestra el mensaje original seguido de la respuesta nueva introducida.

2.6 Módulos / componentes principales del sistema

El sistema se subdividirá en 4 componentes genéricos que contendrán subdivisiones más específicas:

- *Profile*: Contiene todo lo relativo a usuarios registrados: alta de usuarios, modificación de datos personales, eliminación de usuario, registro de mensajes subidos...
- *Social Media*: Referente a seguidores, seguidos, *kudos*, y sistemas de puntuación.
- *Administration*: Inserción de alineaciones probables, consulta y borrado de mensajes y categorías de foro, lógica de *login* y *logout*, alta de administrador, creación de *quizz*, añadir/eliminar jugadores, añadir /eliminar equipos.
- *Games*: Encuestas / Quizz / foro etc...

2.7 Calidad

Se deben realizar pruebas unitarias para comprobar las funcionalidades del sistema. Éstas podrán ser automatizadas o no.

Se realizará un estudio de la calidad del código producido mediante una herramienta de análisis como *JHawk* o *SonarQube*.

Se realizará en la fase final del proyecto pruebas de interacción persona-ordenador con usuarios reales de diferentes perfiles para comprobar el funcionamiento de la aplicación. Se procurará, siempre que sea posible, utilizar diferentes navegadores y dispositivos para las pruebas.

2.8 Planificación temporal

Se asigna le una cantidad de tiempo adicional a la fase de implementación, aunque se adaptará en función de los requerimientos de las pruebas de evaluación continuada. Se utiliza la herramienta *GanttProject* para realizar la planificación temporal. A continuación, se muestra la imagen exportada por *GanttProject*.

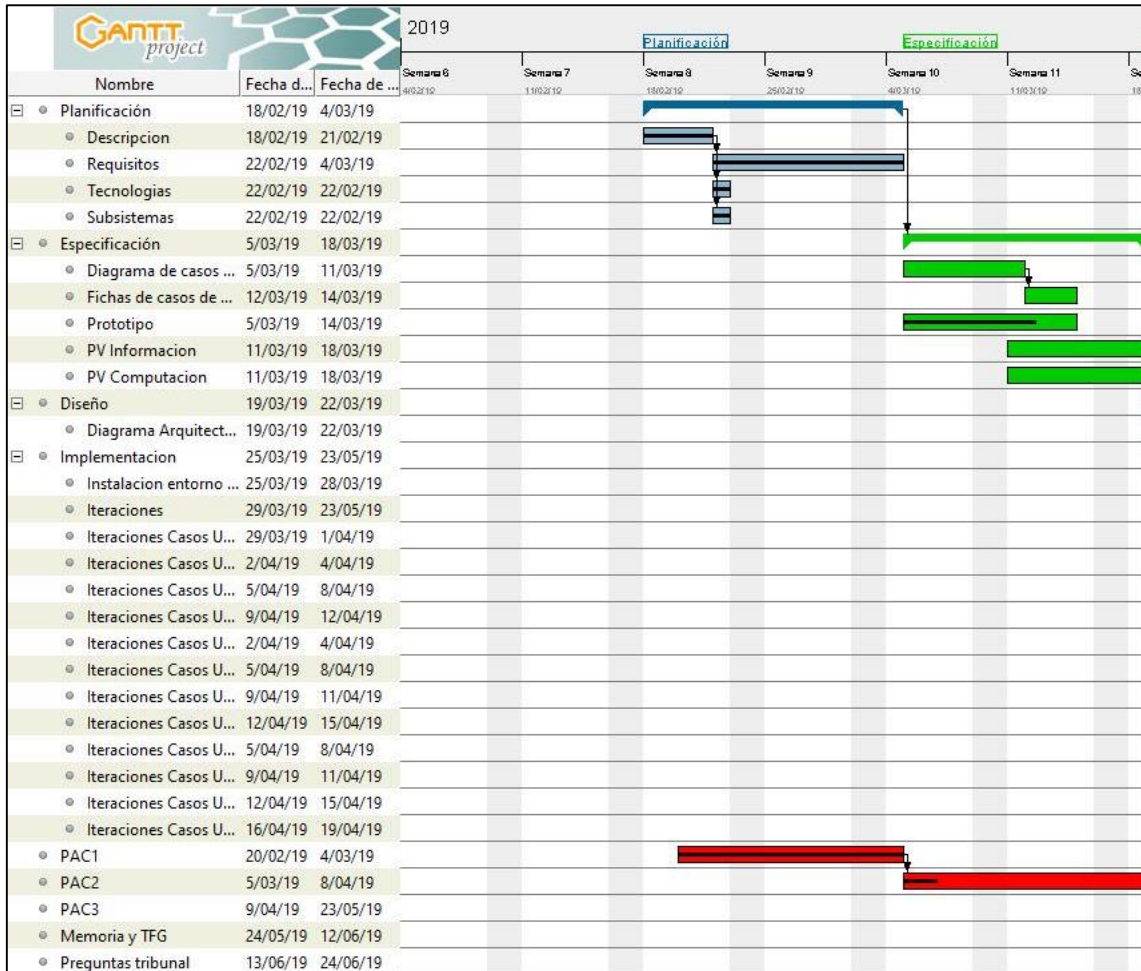


Ilustración 2 Diagrama de Gantt (I)

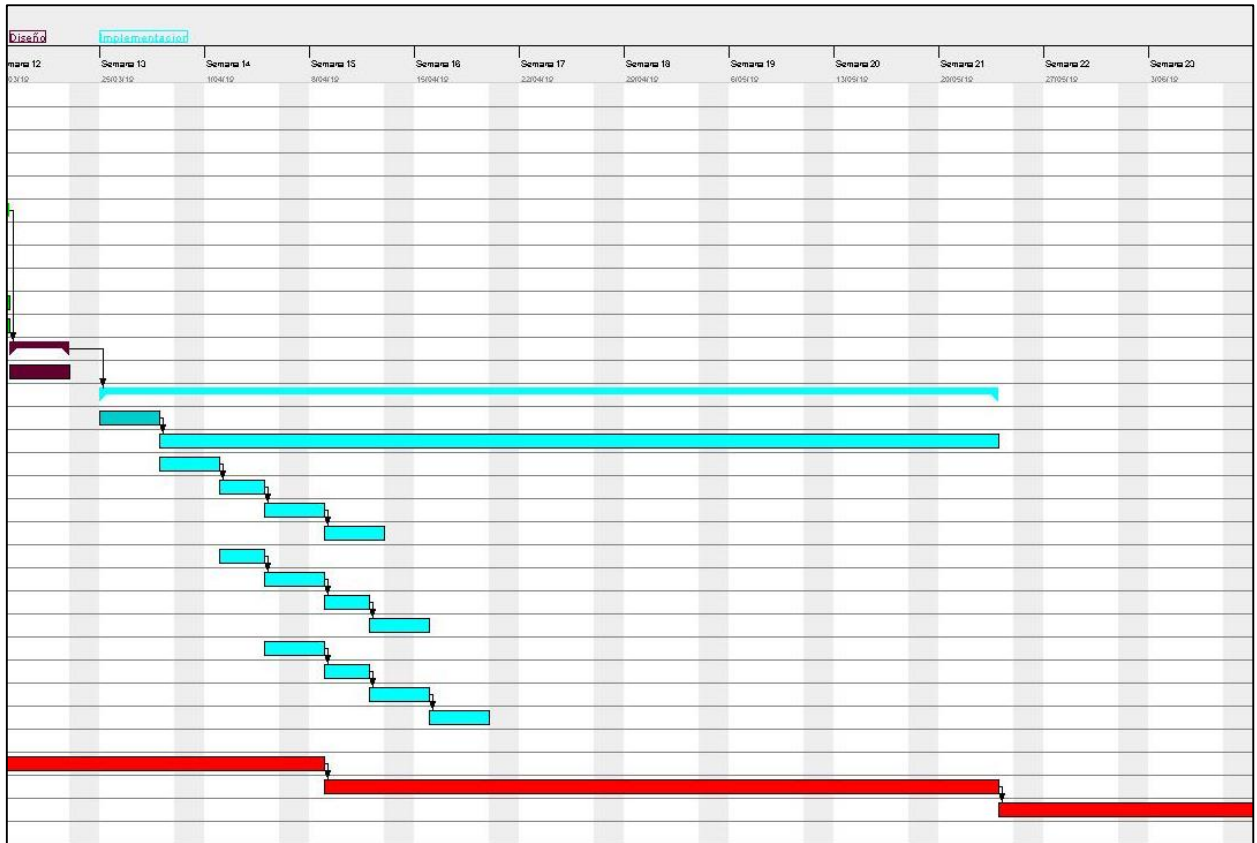


Ilustración 3 Diagrama de Gantt (II)

2.9 Riesgos

- Problemas legales derivados de los derechos de futbolistas o de LaLiga BBVA.
 - Para un uso no-comercial, no existen problemas legales.
- El uso de *relational-extension* SQL y herencias puede ser difícil de implementar.
 - Se evita diseñando sin herencias en los usuarios (administradores y usuarios serán entidades diferentes).
- Alcance excesivo del proyecto.
 - Se priorizan las secciones (como se verá en el próximo apartado) para minimizar este riesgo y se elimina una sección “Valoraciones de jugadores” previamente prevista.

2.10 Alcance del proyecto

Después del análisis de requisitos, la definición de los casos de uso, los riesgos y la planificación temporal, se considera oportuno **priorizar las secciones principales** del proyecto ante la elevada probabilidad no poder implementar todos los requisitos y funcionalidades con el tiempo disponible para la implementación:

1. Alineaciones probables.
2. Encuestas.

3. Quiz.
4. Foro.

Las funcionalidades más estructurales del proyecto y los casos de uso relacionados, no se priorizan ya que se consideran fundamentales y su implementación obligatoria (Registro, Perfil de usuario, Login, Logout, Buscar usuarios, Seguir y ser seguidos por usuario, ...)

3 Punto de vista de la información

El **punto de vista de la información** describe en detalle la información que tratará el sistema. En este documento se presenta un esquema invariante donde se especifican las estructuras de datos que gestiona el sistema a través de un diagrama de clases UML, que determina los tipos de datos, sus relaciones y posibles restricciones.

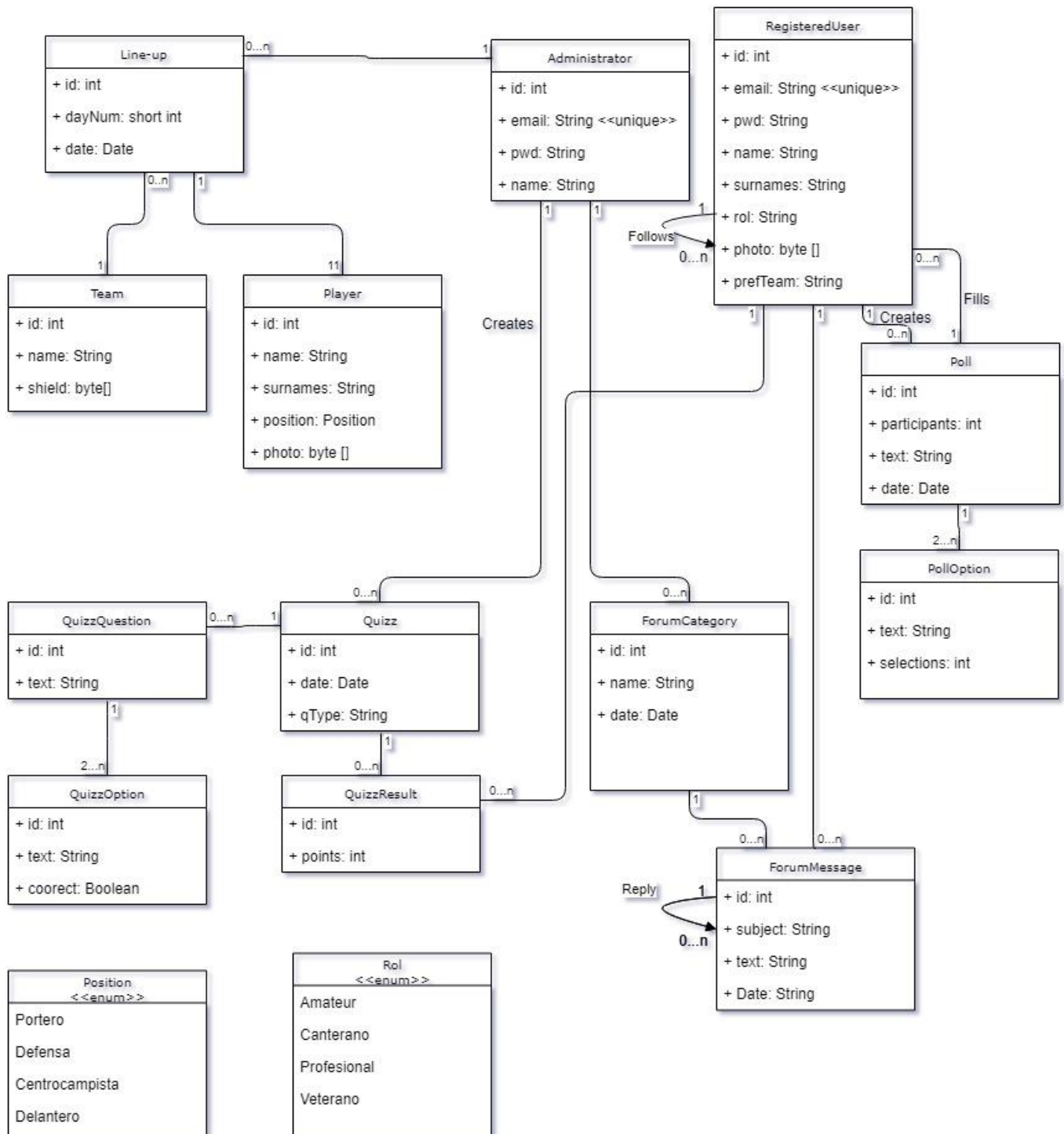


Ilustración 4 Diagrama de clases UML. PV Información

4 Punto de vista de la computación

El **punto de vista de la computación** contiene el diseño funcional que se realiza a partir de los puntos de vista previamente descritos, definiendo las interfaces y cómo se encapsulan en componentes arquitectónicos. En este documento se definen los estilos arquitectónicos que se van a utilizar (cliente-servidor, por capas y componentes distribuidos) partiendo de un diseño de grano grueso, para especificar más en sucesivos refinamientos.

A continuación, se muestra el progresivo refinamiento de los componentes del sistema y se define su arquitectura. Los estilos arquitectónicos aplicados son:

- **Arquitectura cliente-servidor**
- **Arquitectura de componentes distribuidos**
- **Arquitectura por capas**

Mediante esta mezcla de arquitecturas se consigue flexibilidad, escalabilidad, mitigar el acoplamiento, mejorar la tolerancia a fallos y mejorar la mantenibilidad. La distribución por capas favorece la escalabilidad y permite encapsular funcionalidades que hay que ofrecer en capas con interfaces muy definidas (por ejemplo, presentación, negocio y datos). Si estructuramos los elementos de cada capa siguiendo una arquitectura de objetos distribuidos, conseguimos un sistema fácilmente escalable.

Se podría mejorar el rendimiento añadiendo más servidores y replicando componentes, con componentes altamente cohesionados y más mantenibles.

4.1 Diagrama de componentes

Teniendo en cuenta los estilos arquitectónicos mencionados, se muestra un primer diagrama de componentes de grano grueso. Por evitar repeticiones innecesarias, solo se muestra la estructura del subsistema *Admin*. El resto de subsistemas (*Profile*, *SocialMedia* y *Game*) siguen la misma estructura y se incorporan en las mismas capas (presentación, negocio e integración). Con posterioridad, en el tercer refinamiento, se especifican todos los subsistemas ya que existen diferencias que vale la pena remarcar:

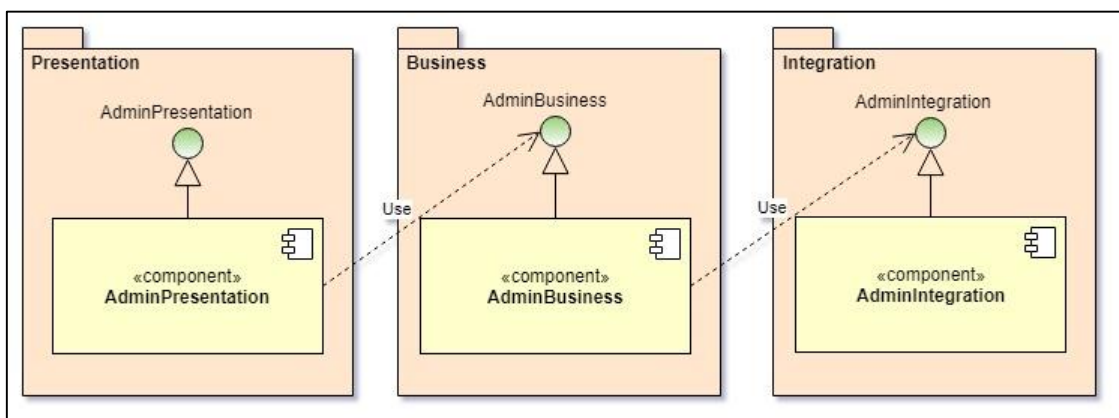


Ilustración 5 Diagrama de componentes de grano grueso

Las interfaces se especifican a continuación y se evita volver a reproducirlas en posteriores diagramas para facilitar la visualización de los diagramas. La distribución de funcionalidades, se procura

distribuir de forma equitativa evitando interfaces con muchas funcionalidades en comparación con el resto. Ha resultado inevitable cargar claramente más la interfaz de *Games*.

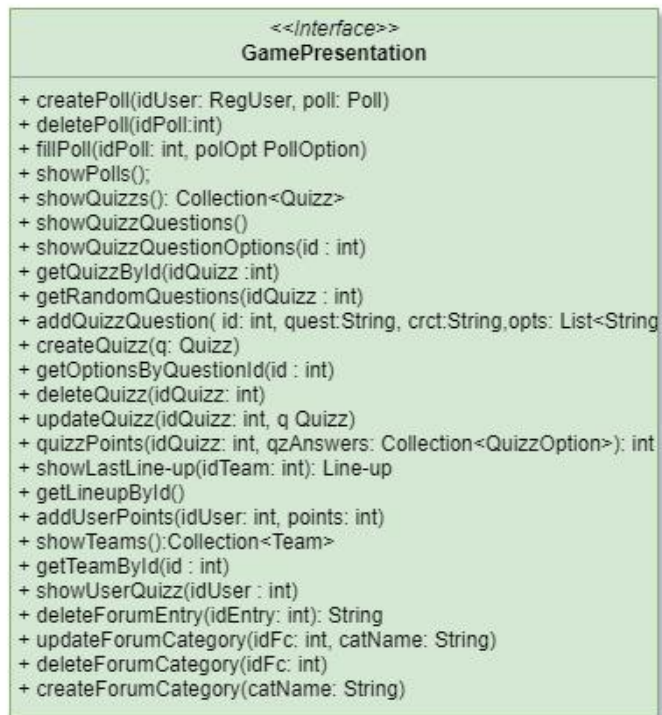
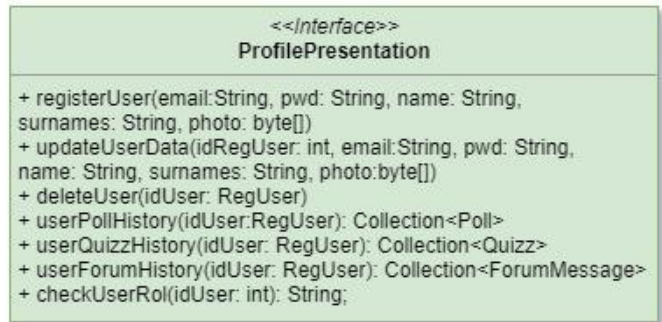
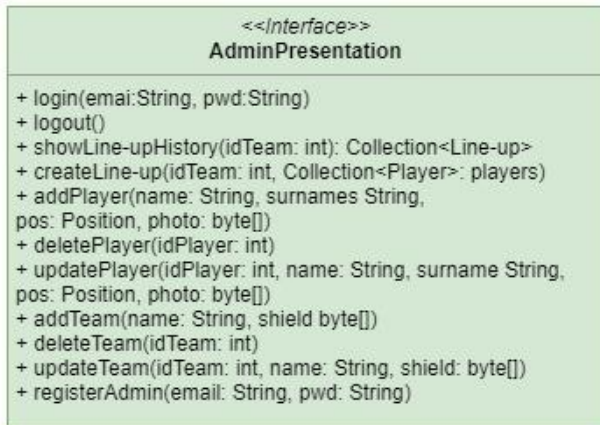


Ilustración 6 Definición de las interfaces del sistema

4.1.1 Primer refinamiento

Un primer refinamiento del diagrama de componentes, consiste en la aplicación de un patrón **MVC** (*Model – View – Controller*), tal y como se especifica en el siguiente diagrama. El modelo se define en la capa de negocio, el controlador y las vistas, en la capa de presentación. Con la aplicación de este patrón, se permitirá separar y/o delegar cada una de las distintas funcionalidades según la capa de la aplicación que le corresponda.

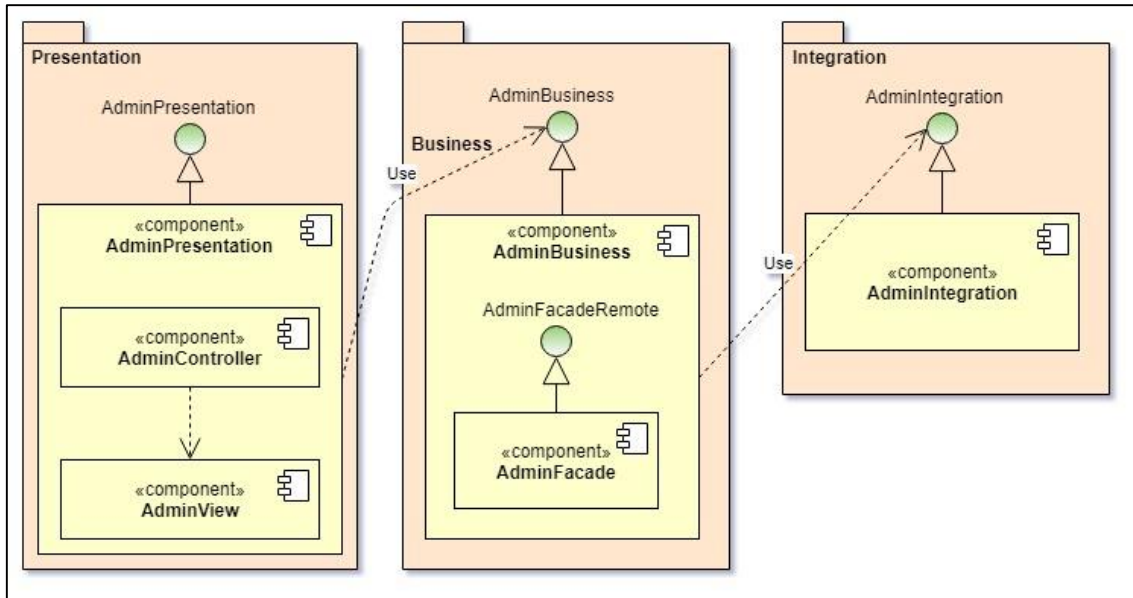


Ilustración 7 Diagrama de componentes. Primer refinamiento: MVC y fachada

4.1.2 Segundo refinamiento

A continuación, se realiza un segundo refinamiento con la finalidad de desacoplar cada una de las funcionalidades y/o acciones de la vista. Para ello, se utilizarán los patrones **FrontController** y **Command**. Este refinamiento, solo hace referencia a la capa de presentación.

El patrón *FrontController* permite tener un único punto de entrada para todas las peticiones que provengan de un navegador aportando flexibilidad, re-utilización de código y evitando la redundancia del mismo.

Las operaciones no se implementarán dentro del controlador, se utilizará un esquema basado en el patrón *Command*. El controlador será el responsable de coordinar todo el proceso, pero la implementación está desacoplada. Cada acción que el usuario puede realizar por medio de la interfaz de usuario (GUI) se mapeará con una vista y el controlador simplemente ejecuta estas acciones.

En resumen, el patrón *Command* se utiliza para desacoplar operaciones y acciones. Se define una acción por cada funcionalidad especificada en la interfaz, serán necesarias algunas vistas menos que acciones ya que pueden agruparse varias acciones en una vista. Dichas vistas se detallarán posteriormente en los prototipos de pantalla.

Nuevamente se muestra solo el subsistema *Administration* ya que, en el próximo refinamiento, se especificarán todos los subsistemas en detalle.

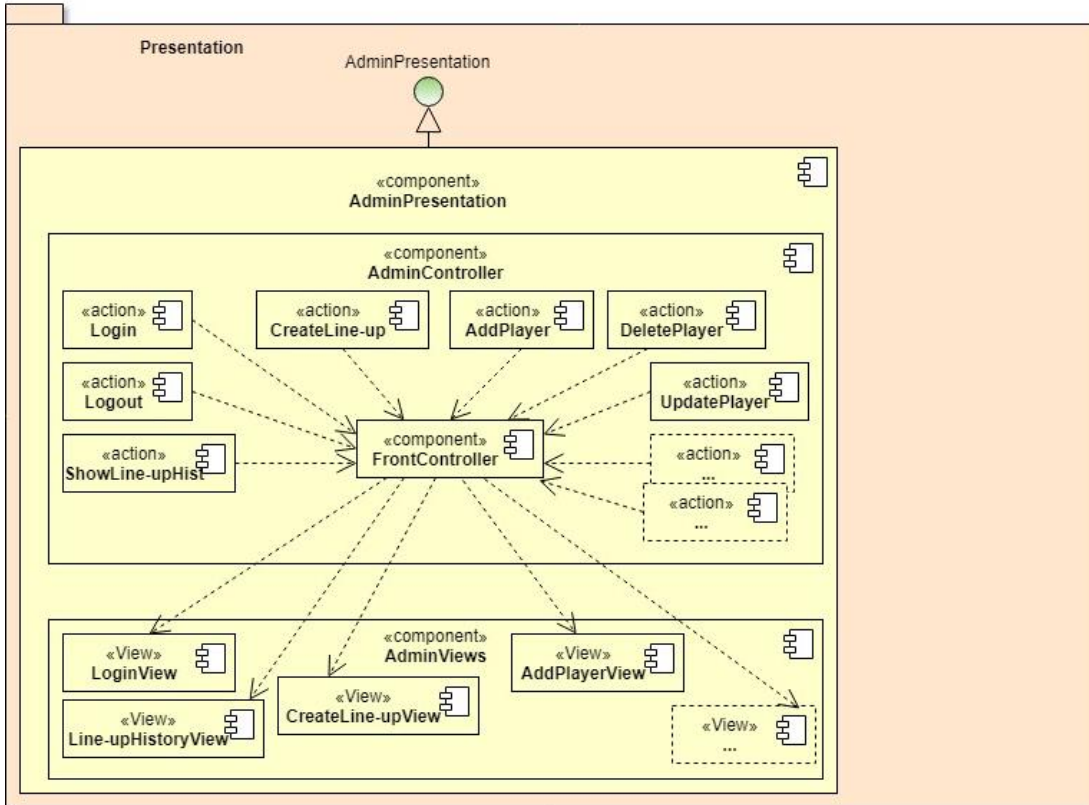


Ilustración 8 Diagrama de componentes. Segundo refinamiento: *FrontController* y *Commad*

4.1.3 Tercer refinamiento

Por último, se organizan los componentes para obtener una distribución bien agrupada que favorecerá la mantenibilidad del sistema. Se añade un control de errores transversal representado con un componente de tipo <<view>>.

La capa de negocio no requiere refinamientos adicionales de manera que es válido el diagrama descrito en el primer refinamiento, con tan solo un patrón **facade** aplicado. Por lo tanto, se muestra el refinamiento, previo a la aplicación de una tecnología concreta, para las capas de presentación e integración de todos los subsistemas:

4.1.3.1 Capa de presentación:

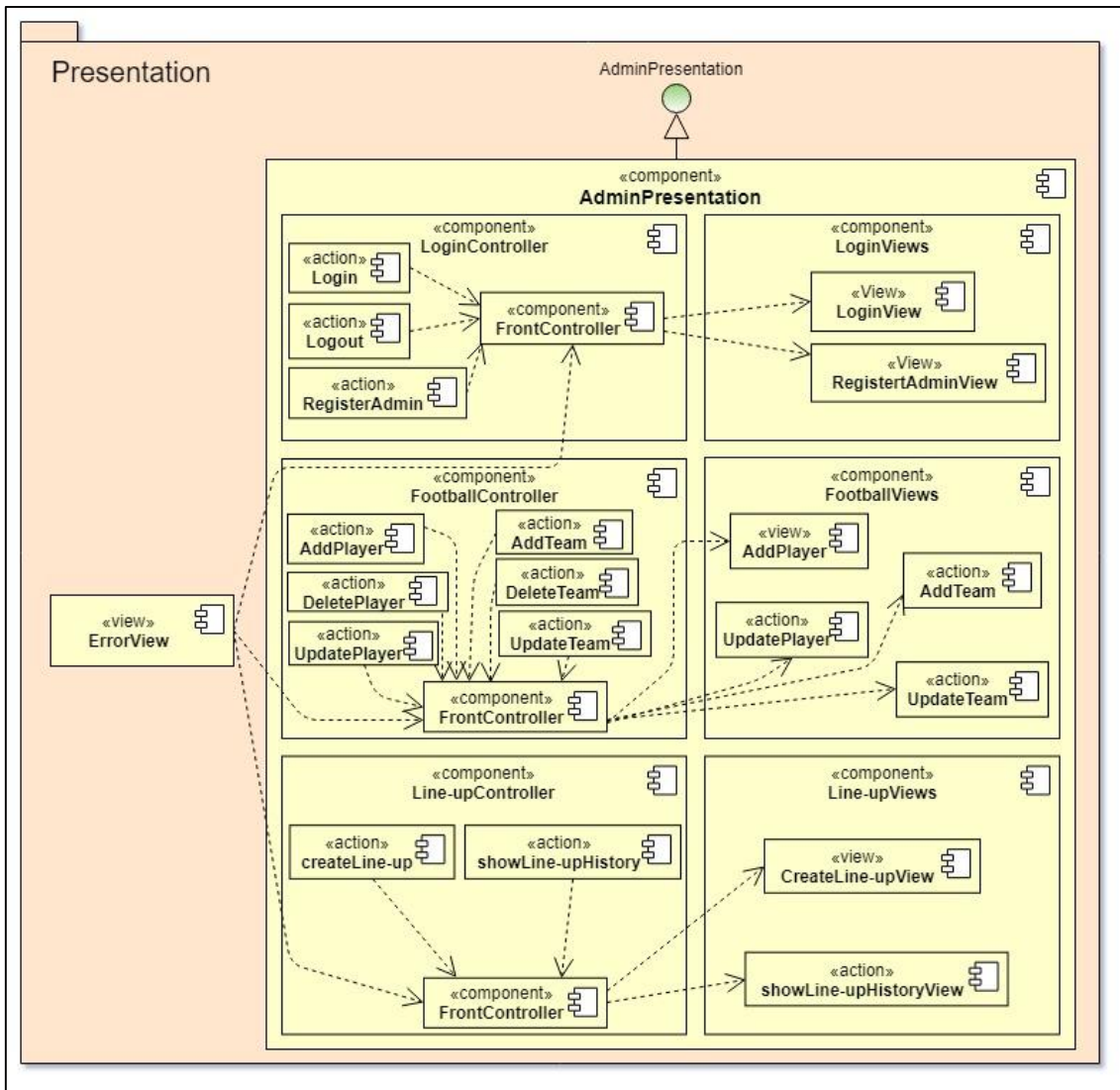


Ilustración 9 Diagrama de componentes del subsistema de *administration*. Tercer refinamiento: ordenación de componentes

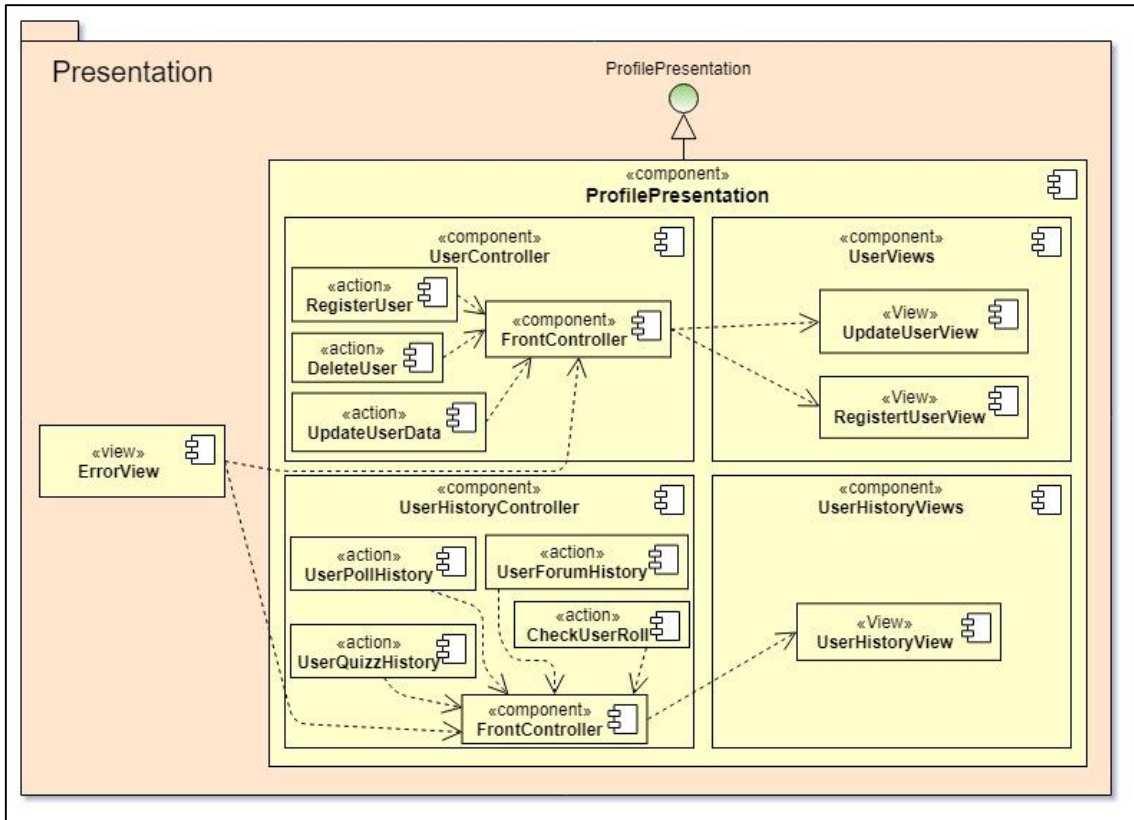


Ilustración 10 Diagrama de componentes del subsistema de *Profile*. Tercer refinamiento: ordenación de componentes

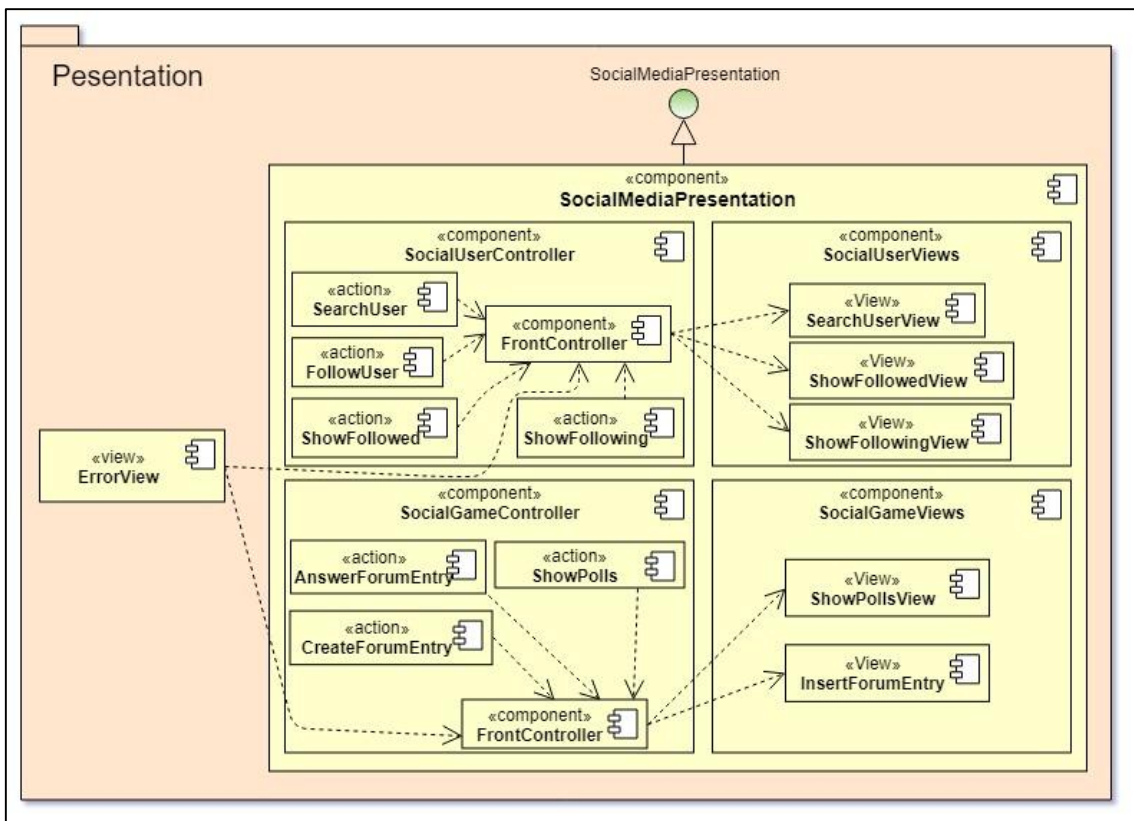


Ilustración 11 Diagrama de componentes del subsistema de *SocialMedia*. Tercer refinamiento: ordenación de componentes

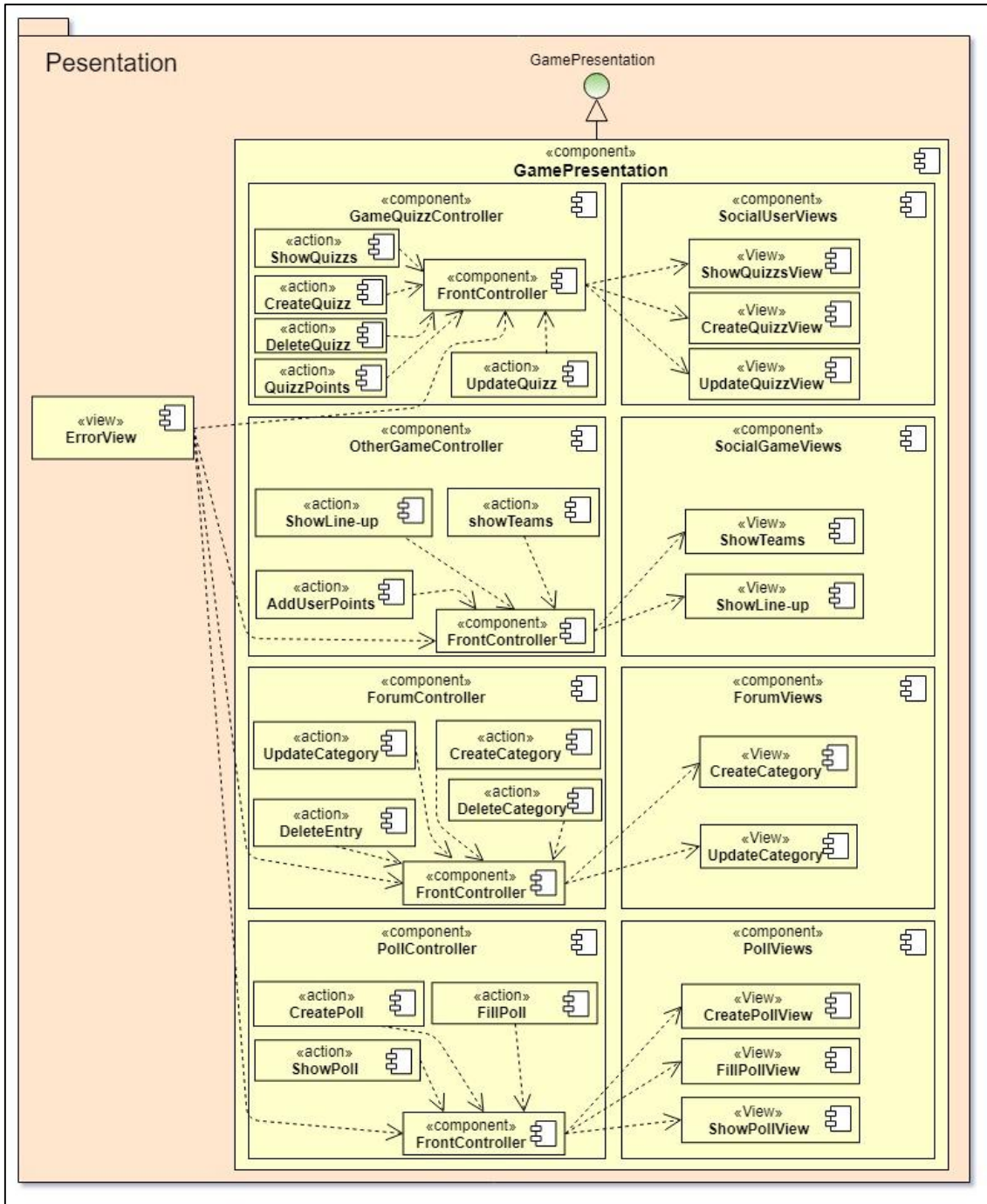


Ilustración 12 Diagrama de componentes del subsistema de *Game*. Tercer refinamiento: ordenación de componentes

Ha sido necesario añadir varias vistas a este diagrama, modificando lo inicialmente previsto:

1. Vista que muestra historial de alineaciones
2. Vista que muestra los resultados de una encuesta.
3. Vista que muestra resultados de un Quiz
4. Vista de presentación de Quiz

4.1.3.2 *Capa de negocio*

Como se ha comentado, no se requieren refinamientos adicionales en la capa de negocio, siendo suficiente el diagrama de componentes definido en el primer refinamiento.

4.1.3.3 Capa de integración

Se definen los componentes que se usarán en la capa de integración para cada subsistema, teniendo en cuenta el punto de vista de la información y las funcionalidades definidas en las interfaces:

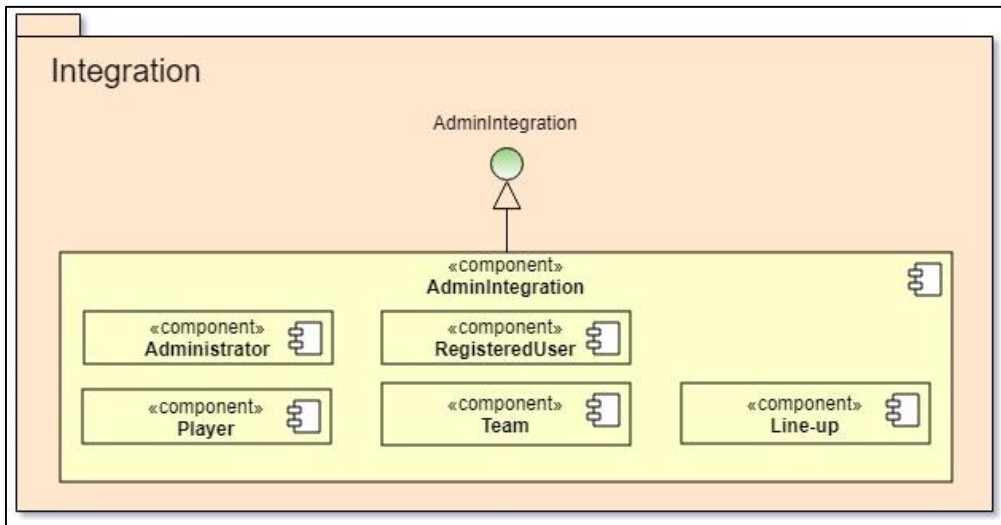


Ilustración 13 Diagrama de componentes de la capa de integración del subsistema *Administration*

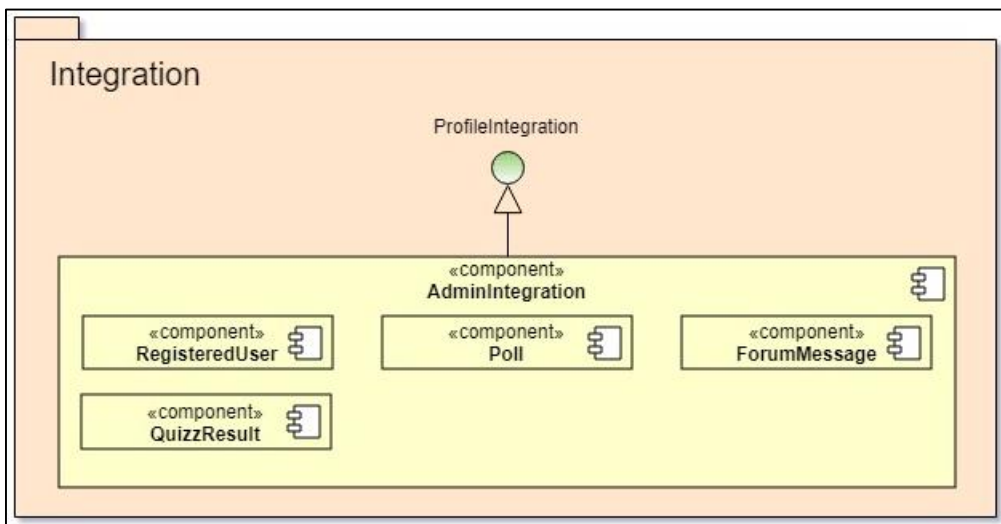


Ilustración 14 Diagrama de componentes de la capa de integración del subsistema *Profile*

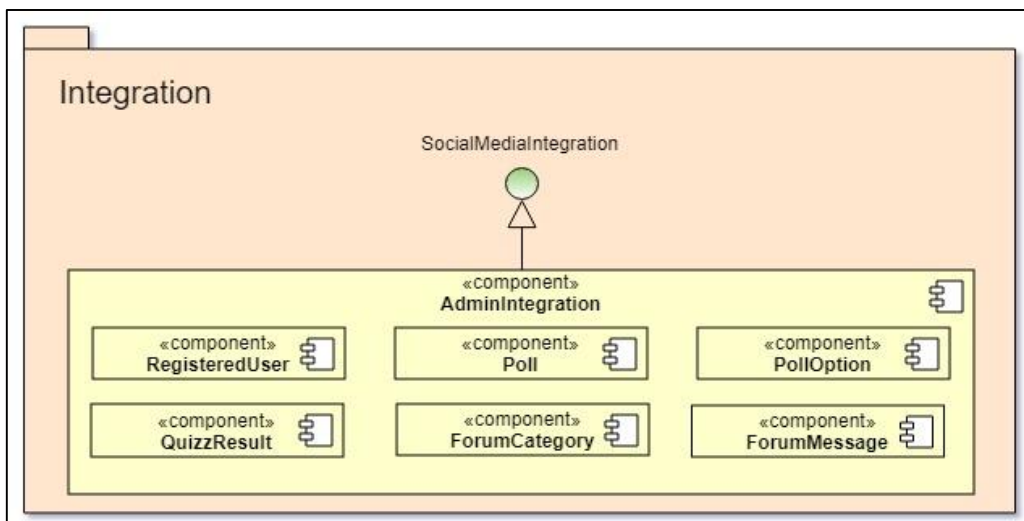


Ilustración 15 Diagrama de componentes de la capa de integración del subsistema *SocialMedia*

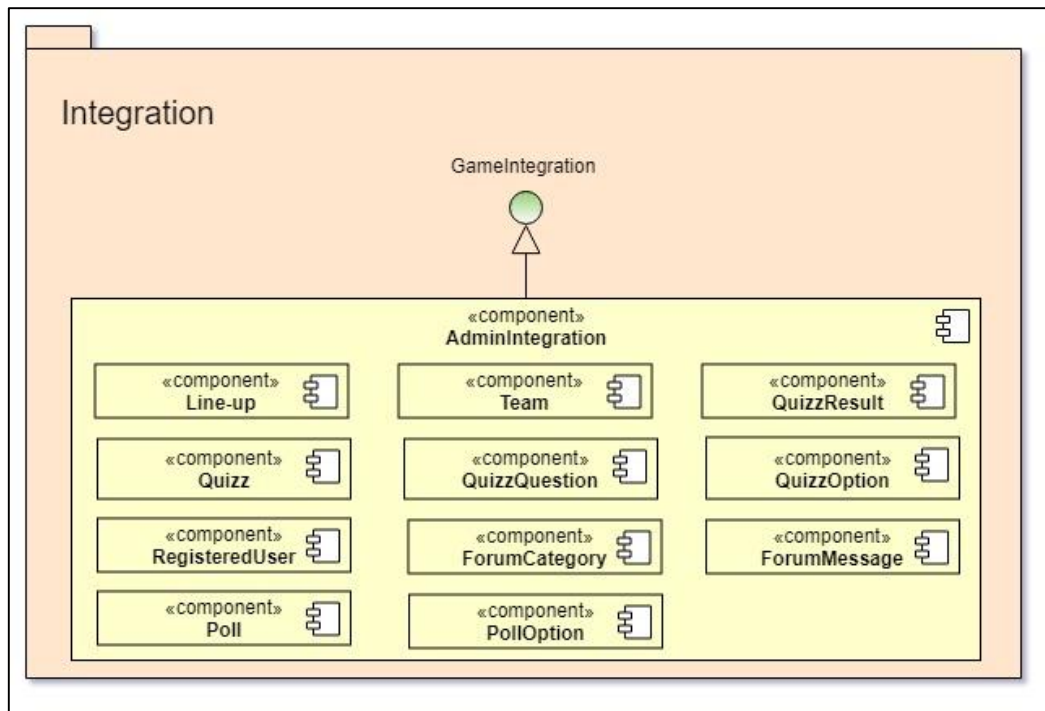


Ilustración 16 Diagrama de componentes de la capa de integración del subsistema *Game*

4.2 Navegación y vistas

4.2.1 LoginView



Futboleros
la red social de los amantes del fútbol

Alineaciones Probables
Encuestas
Quizzes
Foro
Administracion

Login

Email *:

Contraseña *:

[Recordar contraseña](#)

Ilustración 17 Vista Login

4.2.2 RegisterAdminView

The screenshot shows the 'Registrar Nuevo Administrador' form. At the top, there is a navigation bar with the Futboleros logo and the tagline 'la red social de los amantes del fútbol'. On the right, a user profile for 'admin@admin.com' is visible, with buttons for 'Editar datos' and 'Cerrar Sesión'. Below the navigation bar, there are five menu items: 'Alineaciones probables', 'Encuestas', 'Quizz', 'Foro', and 'Administracion'. The main content area is titled 'Registrar Nuevo Administrador' and contains three input fields: 'Email *:', 'Contraseña *:', and 'Contraseña (bis) *:'. The entire form area is overlaid with a large 'X'.

Ilustración 18 Vista "Registrar Administrador"

4.2.3 AddTeamView

*UpdateTeamView se asemejará mucho, por lo que no se considera necesario especificarla.

The screenshot shows the 'Añadir nuevo equipo' form. The navigation bar is identical to the previous view, but the 'Administracion' menu item is highlighted with a green dashed border. The main content area is titled 'Añadir nuevo equipo' and contains three input fields: 'Nombre *:', 'Escudo:', and 'Seleccionar Foto * 150x150 pixels approx'. Below the input fields, there is a blue button labeled 'Añadir'. The entire form area is overlaid with a large 'X'.

Ilustración 19 Vista "Añadir equipo"

4.2.4 AddPlayerView

*UpdatePlayerView se asemejará mucho, por lo que no se considera necesario especificarla

The screenshot shows the 'Añadir nuevo jugador' (Add new player) form. At the top, there is a navigation bar with the site logo 'Futboleros la red social de los amantes del fútbol', a user profile for 'admin@admin.com' (Sesión de administrador), and buttons for 'Editar datos' and 'Cerrar Sesión'. Below the navigation bar are menu items: 'Alineaciones probables', 'Encuestas', 'Quizz', 'Foro', and 'Administracion'. The main form area is titled 'Añadir nuevo jugador' and contains the following fields:

- Equipo *:** A dropdown menu with the text 'Selecciona equipo'.
- Nombre *:** A text input field.
- Apellidos *:** A text input field.
- Posición *:** A dropdown menu with the text 'Selecciona posición'.
- Escudo :** A text input field.
- Seleccionar Foto** * 150x150 pixels approx: A button to upload a player photo.

At the bottom left of the form is a blue 'Añadir' button.

Ilustración 20 Vista "Añadir jugador"

4.2.5 ShowLine-upHistoryView


The screenshot shows the 'Historial de alineaciones' (Line-up history) view. At the top, there is a navigation bar with the site logo 'Futboleros la red social de los amantes del fútbol', a user profile for 'admin@admin.com' (Sesión de administrador), and buttons for 'Editar datos' and 'Cerrar Sesión'. Below the navigation bar are menu items: 'Alineaciones probables', 'Encuestas', 'Quizz', 'Foro', and 'Administracion'. The main content area is titled 'Historial de alineaciones' and contains the following elements:

- Equipo:** *Equipo previamente seleccionado
- + Crear nueva alineacion o modificar última guardada**
- Table:** A table with columns 'Fecha', 'Jornada', and 'Ver'. The table lists several entries with dates and a 'Ver' button (eye icon).

Fecha ↓	Jornada	Ver
* fecha mas reciente	20	👁
fecha	19	👁
fecha	15	👁
fecha	14	👁
fecha temporada pasada	40	👁
fecha temporada pasada	38	👁
...

Ilustración 21 Vista "Mostrar historial de alineaciones"

4.2.6 CreateLine-upView



Futboleros
la red social de los amantes del fútbol

Foto genérica admin@admin.com Editar datos ²
Sesión de administrador Cerrar Sesión ¹

Alineaciones probables
Encuestas
Quizz
Foro
Administración

Crear alineación probable

Equipo: Equipo previamente seleccionado

Formación: (por ejemplo 4 4 2)

Jornada: (última jornada guardada) i Se muestra la ultima alineación guardada para el equipo. Inserta la proxima jornada o modifica la ultima guardada. ¡No se pueden guardar jornadas anteriores!

Formación (4-4-2)

Portero: Seleccionar portero

Defensas: Seleccionar jugador
Seleccionar jugador
Seleccionar jugador
Seleccionar jugador

Medios: Seleccionar jugador
Seleccionar jugador
Seleccionar jugador
Seleccionar jugador


Delanteros: Seleccionar jugador
Seleccionar jugador

Guardar

Ilustración 22 Vista "Crear alineación"

4.2.7 RegisterUserView

* Muy similar a updateUserView



Futboleros
la red social de los amantes del fútbol

Alineaciones Probables
Encuestas
Quizzes
Foro
Administración

Registrar Usuario

Email *:

Contraseña *:

Contraseña (bis) *:

Nombre *:

Apellidos:

Equipo preferido:

Fotografía: Seleccionar Foto * 300x400 pixels approx

Ilustración 23 Vista "Registrar usuario"

4.2.8 UserHistoryView

Futboleros
la red social de los amantes del fútbol

Nombre Usuario: categoria de futbolero

Encuestas

Puntuaciones
Imagen Rol: Categoría: Futbolero experto

Puntuacion	Clasificacion
45	112

Encuestas
5 últimas encuestas creadas

fecha	nº part	Titular encuesta
fecha	nº part	Titular encuesta
fecha	nº part	Titular encuesta

Resultados en Quizz

fecha	6/10	Nombre Quizz
fecha	6/10	Nombre Quizz
fecha	6/10	Nombre Quizz

Últimos mensajes en foro

fecha	Respondido!	Titular mensaje
fecha	Respondido!	Titular mensaje
fecha	Sin respuesta	Titular mensaje

Ilustración 24 Vista "Historial de usuario"

4.2.9 SearchUserView

Futboleros
la red social de los amantes del fútbol

Nombre Usuario: categoria de futbolero

Encuestas

Buscar Usuario

Email: Rellena solo 1 de los campos

Nombre:

Apellido/s:

Resultados

email	Nombre	Apellidos	Ver historial	Seguir
user@email.com	User	-		
useriam@email.com	useriam	Gamez Told		

Ilustración 25 Vista "Buscar usuario"

4.2.10 ShowFollowersView

* ShowFollowingView es muy semejante

Futboleros
la red social de los amantes del fútbol

Foto Nombre Usuario categoría de futbolero Editar datos 2 Cerrar Sesión 1

Alineaciones probables Encuestas Quizz Foro Administracion

Usuarios que sigues

email	Nombre	Apellidos ↓	Ver historial	Seguir
user@email.com	User	-		
useriam@email.com	useriam	Gamez Told		
joseri@gmeil.com	Jose	Riba Camps		
...

Ilustración 26 Vista "Mostrar seguidores"

4.2.11 ShowPollsView

Futboleros
la red social de los amantes del fútbol

Alineaciones Probables **Encuestas** Quizzes Foro Administración

Inicia Sesión de futbolero 1 Registrar Nuevo futbolero 2

i Inicia sesión para unirse a la red social y podrás comentar en el foro, rellenar tus encuestas, y realizar quizzes.

Encuestas

Título	Fecha ↓	Votos	Resultados
Encuesta 1	fecha reciente	102	
Encuesta 2	dd/mm/aaaa	250	
Encuesta 3	dd/mm/aaaa	190	
Encuesta 4	dd/mm/aaaa	300	
...

Ilustración 27 Vista "Mostrar encuestas"

4.2.12 InsertForumEntry

The screenshot shows the 'Añadir mensaje' form in the Futboleros forum. At the top, there is a navigation bar with the site logo and name 'Futboleros la red social de los amantes del fútbol'. On the right, there is a user profile section with a photo placeholder, the username 'Nombre Usuario', and the category 'categoria de futbolero'. Below this are buttons for 'Editar datos' and 'Cerrar Sesión'. A secondary navigation bar contains buttons for 'Alineaciones probables', 'Encuestas', 'Quizz', 'Foro', and 'Administracion'. A third bar contains buttons for 'Seguidores', 'Buscar jugadores', 'Mi Historial', 'Eliminar cuenta', and 'Crear encuesta'. The main form area is titled 'Añadir mensaje' and contains the following fields:

- Categoría:** A dropdown menu with 'Categoria seleccionada' selected.
- Asunto:** A text input field containing '(si es respuesta) RE: Asunto mensaje respondido'.
- Contenido:** A large text area for the message content.
- Enviar:** A button to submit the message.

Ilustración 28 Vista "Insertar entrada en foro"

4.2.13 CreateQuizView

The screenshot shows the 'Crear Quiz' form in the Futboleros forum. At the top, there is a navigation bar with the site logo and name 'Futboleros la red social de los amantes del fútbol'. On the right, there is a user profile section for an administrator with the email 'admin@admin.com' and the role 'Sesión de administrador'. Below this are buttons for 'Editar datos' and 'Cerrar Sesión'. A secondary navigation bar contains buttons for 'Alineaciones probables', 'Encuestas', 'Quizz', 'Foro', and 'Administracion', with 'Administracion' highlighted in green. The main form area is titled 'Crear Quiz' and contains the following elements:

- Info icon:** A warning icon with a text box stating: 'Se recomienda añadir un mínimo de 10 preguntas. Si se introducen mas de 10, se hará una selección aleatoria de las mismas a la hora de realizar el quizz. Solo puede haber 1 respuesta correcta exactamente.'
- Título:** A text input field containing 'Acierta los máximos goleadores de mundiales'.
- Tipo:** A dropdown menu with 'Fútbol internacional' selected and '(Fútbol internacional/ genérico/ etc...)' as a hint.
- Pregunta 1:** A text input field containing 'Quien fué el máximo goleador del mundial de Korea en el '96'. Below it are five radio button options: 'Di Stefano', 'Amavisca', 'Pirri', 'Klose', and 'Di Baggio'.
- + Añadir nueva pregunta:** A button to add a new question.
- Finalizar y guardar:** A blue button to save the quiz.

Ilustración 29 Vista "Crear Quiz"

4.2.14 NewQuizQuestionView



Futboleros
la red social de los amantes del fútbol



admin@admin.com
Sesión de administrador

Editar datos 2

Cerrar Sesión 1

Alineaciones probables
Encuestas
Quiz
Foro
Administración

Nueva Pregunta

Título: Acierta los máximos goleadores de mundiales

Tipo: Fútbol internacional

Pregunta 2: Quien fué el máximo goleador del mundial de Alemania en el 2003

Opciones:

Klose

Beckenbauer

Ronaldo Nazario

+ Añadir opción


i Selecciona la respuesta correcta! Añade un mínimo de 2 opciones

Finalizar y guardar


Ilustración 30 Vista "Nueva pregunta de Quiz"

4.2.15 ShowTeamsView

* Vista de administrador



Futboleros
la red social de los amantes del fútbol



admin@admin.com
Sesión de administrador

Editar datos 2

Cerrar Sesión 1

Alineaciones probables
Encuestas
Quiz
Foro
Administración

i Selecciona un equipo para ver su historial de alineaciones y modificar la ultima alineacion o crear una nueva.

Equipos

								Nombre
								Real Madrid CF
								FC Barcelona
								Valencia CF
								Real Betis BP
								Leganés
								etc...

+ Añadir nuevo equipo

Ilustración 31 Vista "Mostrar equipos"

4.2.16 ShowLine-upView



Futboleros
la red social de los amantes del fútbol

Alineaciones Probables
Encuestas
Quizzes
Foro
Administración

Inicia Sesión de futbolero ¹
Registrar Nuevo futbolero ²



Inicia sesión para unirse a la red social y podrás comentar en el foro, rellenar tus encuestas, y realizar quizzes.


Alineación probable

Equipo: Leganes



Ilustración 32 Vista "Mostrar alineación"

4.2.17 CreatePollView



Futboleros
la red social de los amantes del fútbol

Foto Editar datos ²
categoria de futbolero Cerrar Sesión ¹

Alineaciones probables
Encuestas
Quizz
Foro
Administracion

Crear Encuesta

Título:

Tipo: Fútbol internacional


Opciones:

Diego Costa

Morata

Iago Aspas

+ Añadir opción

 Añade un mínimo de 2 personas! Las encuestas creadas no se pueden modificar

Finalizar y guardar

Ilustración 33 Vista "Crear encuesta"

5 Punto de vista de la ingeniería

El **punto de vista de la ingeniería** define aspectos de distribución, replicación y localización de los objetos del punto de vista de la computación.

Se toman las siguientes decisiones:

- Los usuarios finales utilizarán ordenadores personales o dispositivos móviles con navegadores.
- La base de datos será relacional
- La capa de presentación se ubicará entre el cliente y el servidor. Las capas de negocio e integración estarán ubicadas en un servidor remoto. Esto considera que en el siguiente punto de vista se usará una especificación Java EE.
- Los componentes de la capa de negocio (de los distintos subsistemas) se ejecutarán de manera concurrente en el servidor.
- El acceso a los componentes de la capa de negocio por parte de los componentes de presentación debe estar preparado para ser remoto, por lo que las dos capas se ejecutarán en máquinas virtuales diferentes.
- Los componentes de la capa de negocio accederán a los de integración de forma local, es decir, deberán ejecutarse en la misma máquina virtual.
- No se considera replicación de datos.

6 Punto de vista de la tecnología

El **punto de vista de la tecnología** define los requisitos de la tecnología que se usará para implementar el sistema (tipo de ordenadores, servidores, sistemas operativos, etc...). Los requisitos del punto de vista de la tecnología permiten poder utilizar distintas plataformas de componentes que, en este caso, serán las que sigan las especificaciones de **Java EE**.

Se definen las restricciones siguientes:

- El cliente debe poder acceder al sistema desde cualquier sistema operativo.
- Las comunicaciones entre cliente y servidor se hacen a través de internet.
- La invocación e interacción del cliente se hará a través de un navegador (*Chrome, Mozilla, internet Explorer*)
- Como contenedor de EJB se utilizará **Wildfly**
- Como gestor de bases de datos, se usará **PostgreSQL**

La tecnología de implementación será Java EE, de manera que, partiendo del diseño de componentes previamente especificado, se aplicará el perfil Java EE.

Las decisiones tomadas para aplicar el perfil Java EE en la **capa de presentación** son:

1. Se utiliza el **framework JSF** (*Java Server Faces*).
2. Las vistas se implementan mediante **Facelets**.
3. El controlador será el *Servlet FacesServlet* que viene incorporado en el *framework JSF*, por lo que no es necesario implementarlo.
4. Las acciones que se corresponden con los *commands* se definirán con **Named CDI Beans**.

Las decisiones relativas a la **capa de negocio**, son:

1. Se permitirá el acceso tanto local como remoto a los componentes de la capa de negocio.
2. Por otro lado, se implementará la interfaz de la capa de negocio con un **EJB** de sesión sin estado. Así, el contenedor Java EE resuelve la complejidad inherente a las comunicaciones remotas, proporciona comportamiento transaccional y gestiona la seguridad. Cuando un usuario realice una petición, ésta será recogida por un *servlet*, el cual localizará el EJB que tiene que ejecutar a través del método *lookup*, ya que toda operación que se comunique con la base de datos tiene que ser tratada por un EJB.

Las decisiones relativas a la **capa de integración** son:

1. Todos los datos de la aplicación se almacenarán en una base de datos relacional y el acceso a los datos desde la capa de negocio, será local.
2. Se gestionará la persistencia mediante entidades **JPA**, permitiendo así mapear los datos de las tablas relacionales a objetos manejables en Java.

7 Implementación

7.1 IDE

Para el desarrollo de código, se ha instalado y usado *Eclipse 4.8.0 (Photon)* y los plug-ins de JBoss y GIT.

Eclipse Java EE IDE for Web Developers.

Version: 2018-09 (4.9.0)

Build id: 20180917-1800



Ilustración 34 Pantalla de carga de Eclipse Photon

7.2 Control de Versiones

Se ha utilizado *GIT* como software de control de versiones y se han ido respaldando continuamente los cambios realizados en un repositorio remoto (con una cuenta gratuita en *GitHub*). De esta manera, se asegura el código haciendo prácticamente imposible perderlo por fallo de un dispositivo.



Ilustración 35 Control de versiones GIT



Ilustración 36 Hosting de código para GIT: GitHub

Además, mediante el control de versiones *GIT* se facilita el acceso a versiones anteriores de la aplicación, permitiendo comparar versiones del código o directamente usar versiones anteriores.

Adicionalmente, gracias a subir el código a *GitHub*, se facilita el acceso al código por parte de los consultores y la posibilidad de revisar y controlar todos los *commits* realizados durante la fase de implementación.

La dirección del repositorio es: <https://github.com/LeoWorks/futboleros>

7.3 Comentarios sobre la implementación

Durante la fase de implementación se han plasmado todos los puntos de vista anteriormente descritos. En cualquier caso, es importante señalar algunos aspectos técnicos, modificaciones sobre lo planificado y comentarios sobre la implementación del proyecto:

1. Algunas **imágenes** se guardan en la base de datos. Para mostrar las imágenes en el navegador, por lo tanto, ha sido necesario implementar *servlets* que intercepten las peticiones http que solicitan la imagen almacenada en la base de datos. Se guardan los escudos de los clubes, las fotografías de jugadores y las fotos de perfil de los usuarios. Cada tipo de imagen tiene un *servlet* dedicado para evitar problemas de “cruces” experimentados en la presentación. De cara a la escalabilidad del proyecto y consciente de la importancia del rendimiento de la aplicación, será necesario comprobar que ésta es una implementación adecuada. En caso de problemas de rendimiento, se debe plantear guardar las imágenes en el sistema de archivos del servidor, en vez de la base de datos.
2. **Todos los *backing beans* extienden la clase *BaseBusiness***, que contiene la lógica de *lookup* de los EJBs. Se encapsula así mejor la información y se evita duplicar código. Se ha incorporado esta clase abstracta (*BaseBusiness*) al paquete *utils.session*. Aunque tendría sentido agruparlo en un paquete diferente, se ha preferido, por motivos de mantenibilidad, no crear un paquete que contuviera tan sólo esta clase.
3. El **sistema de autenticación** se ha implementado mediante filtros. Para mejorar la seguridad e impedir que los usuarios puedan navegar hacia atrás y ver información sensible de sesiones anteriores, se ha añadido un filtro que obliga a reenviar información que normalmente se almacena en el caché del navegador (*utils.Filter.NoCacheFilter.java*).
4. Se ha usado la Api de **Java Mail** para el envío de emails durante el proceso de registro de usuarios utilizando una cuenta gratuita de google. Cuando un usuario se registra, recibe en su cuenta de email una contraseña generada de forma aleatoria para acceder al sistema. No se ha hecho así con los administradores, los cuales pueden usar cualquier email, exista o no (solo debe tener un formato correcto y no existir en la base de datos). Tampoco se ha incorporado, por límite de tiempo, la opción de recuperar contraseña, opción relativamente importante y que debe ser considerada en un futuro cercano.
5. Los usuarios que se registren y nunca accedan al portal, ya sea porque no existe su email o porque nunca han accedido, quedaran marcados en la base de datos. **Cada vez que un usuario se autentica en el sistema, se actualiza un atributo de fecha llamado “último acceso”**. Esta variable es nula si nunca ha accedido, permitiendo limpiar la base datos y/o obtener información interesante como usuarios que introducen emails que no existen o usuarios que nunca han llegado a ingresar al sistema por otros motivos.
6. El paquete *managedBean.Game* ha requerido muchas más clases de las inicialmente previstas, especialmente, *managedBean.Game.QuizzController*. Su implementación ha resultado más compleja y extensa de lo previsto, como se puede comprobar en el siguiente apartado, mostrando una **elevada complejidad ciclomática**. Es el único *backing bean* (*createQuizz.java*) que ha requerido un *scope* de sesión (aparte del de *Login*). En aras de no aumentar la complejidad de este paquete, se han minimizado controles de seguridad tales como los problemas que pueden surgir cuando un usuario navega hacia atrás durante el

Quiz. Este punto podría mejorarse, tal vez si en la puesta en producción surgen problemas derivados de ello.

7. Se han usado pocos *beans* validadores **@FacesValidator**, ya que la sintaxis es comprobable en muchos casos desde JSF. Además, muchas de las validaciones son a nivel de negocio y tiene más sentido comprobarlas a ese nivel utilizando *FacesMessages* para advertir al usuario.
8. El control de errores se ha realizado añadiendo etiquetas de **<error-page>** en *web.xml*, donde se redirigen a páginas de error todos los errores 404, 500, excepciones java *Throwable* y *timedOutConection*.
9. Se ha intentado evitar en todo momento el uso de estrategias de **fetch EAGER** en las asociaciones de las entidades JPA. Mediante la estrategia *Join-fetch*, a la hora de recoger de la base de datos las entidades que contienen relaciones *@OneToMany*, se evita usar un *fetch EAGER* que tiene un impacto negativo en el rendimiento.
10. Debido a la poca información que contiene un jugador y/o un equipo, los administradores pueden crear o eliminar dichas entidades, no modificarlas. Si en el futuro se añaden muchos más atributos a estas entidades, el coste de implementar la actualización no es elevado.
11. En la capa de presentación se ha hecho uso de **JQuery** para dinamizarla. Los botones *responsive* o el *popup* de información del usuario, por ejemplo, se implementan con *JQuery*. La validación de la introducción de fotografías, también es se implementa con *JQuery*. De esta manera, se puede comprobar desde el cliente si la fotografía que se quiere introducir tiene un tamaño y extensión adecuados, sin tener que subir la imagen al servidor para comprobarlo, representando un claro problema de rendimiento si el archivo es grande.
12. La presentación se ha construido mediante **templates**, facilitando la modificación de elementos comunes y evitando duplicidades.
13. Se ha procurado utilizar un **HTML semántico** utilizando el doctype de HTML5 e incluyendo tags semánticos como secciones, navs, footers, cabeceras etc... Aunque el uso de *jsf* dificulta producir un código html semántico y el código es aun mejorable en este aspecto.

8 Calidad del código

Se decide utilizar, finalmente, la herramienta **sonarQube** para analizar la calidad del código por ser una herramienta gratuita de buena calidad para analizar software. La herramienta requiere modificar el fichero *build.xml* de *Ant*, y añadir la librería de escaneo de sonarQube para *Ant*, para que *sonarQube*, una vez iniciado, pueda analizar el código mediante el comando “ant sonar”. Posteriormente, se pueden visualizar los resultados y realizar otras operaciones a través del puerto 9000 desde un navegador: <http://localhost:9000/projects>.

El resumen genérico de resultados es el siguiente:

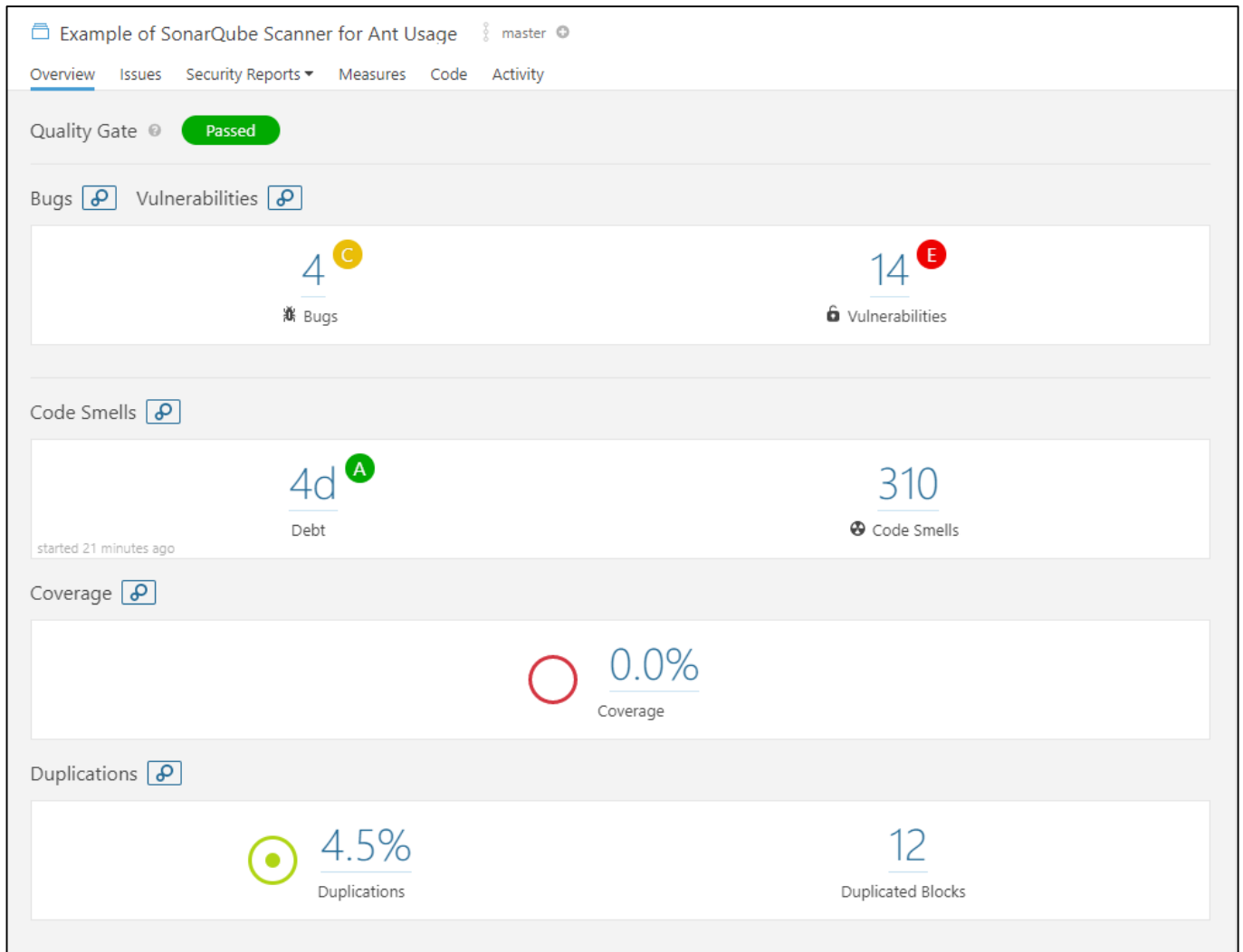


Ilustración 37 Resumen de resultados de SonarQube

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Coverage	Duplications
src	4,174	4	14	310	0.0%	4.5%
ejb	773	2	1	53	0.0%	0.0%
jpa	915	0	0	23	0.0%	7.6%
managedbean	1,938	2	8	193	0.0%	4.8%
META-INF	27	0	0	0	0.0%	0.0%
utils	521	0	5	41	0.0%	6.9%

Ilustración 38 Resumen de resultados por paquetes principales del sistema

Como se verá a continuación *SonarQube* penaliza y llama la atención principalmente sobre el uso no-codificado de una contraseña en el código en el apartado de seguridad. El paquete *managedbean* es el más complejo, con más líneas de código y con más vulnerabilidades, de manera que es un candidato a revisión. *SonarQube* advierte otros problemas menores que también se mencionan a continuación.

8.1 Fiabilidad del código:

Los problemas subrayados en este apartado, indican un posible comportamiento distinto al previsto. Se destaca la clase *ProfileFacadeBean.java*, apreciando una comparación de *strings* con “==” en vez de *equals*, y algunos comentarios necesarios.

Reliability ? Overview	
Overview	
Overall	
Bugs	4
Rating	
Remediation Effort	18min

Ilustración 39 Resultados principales de fiabilidad de código

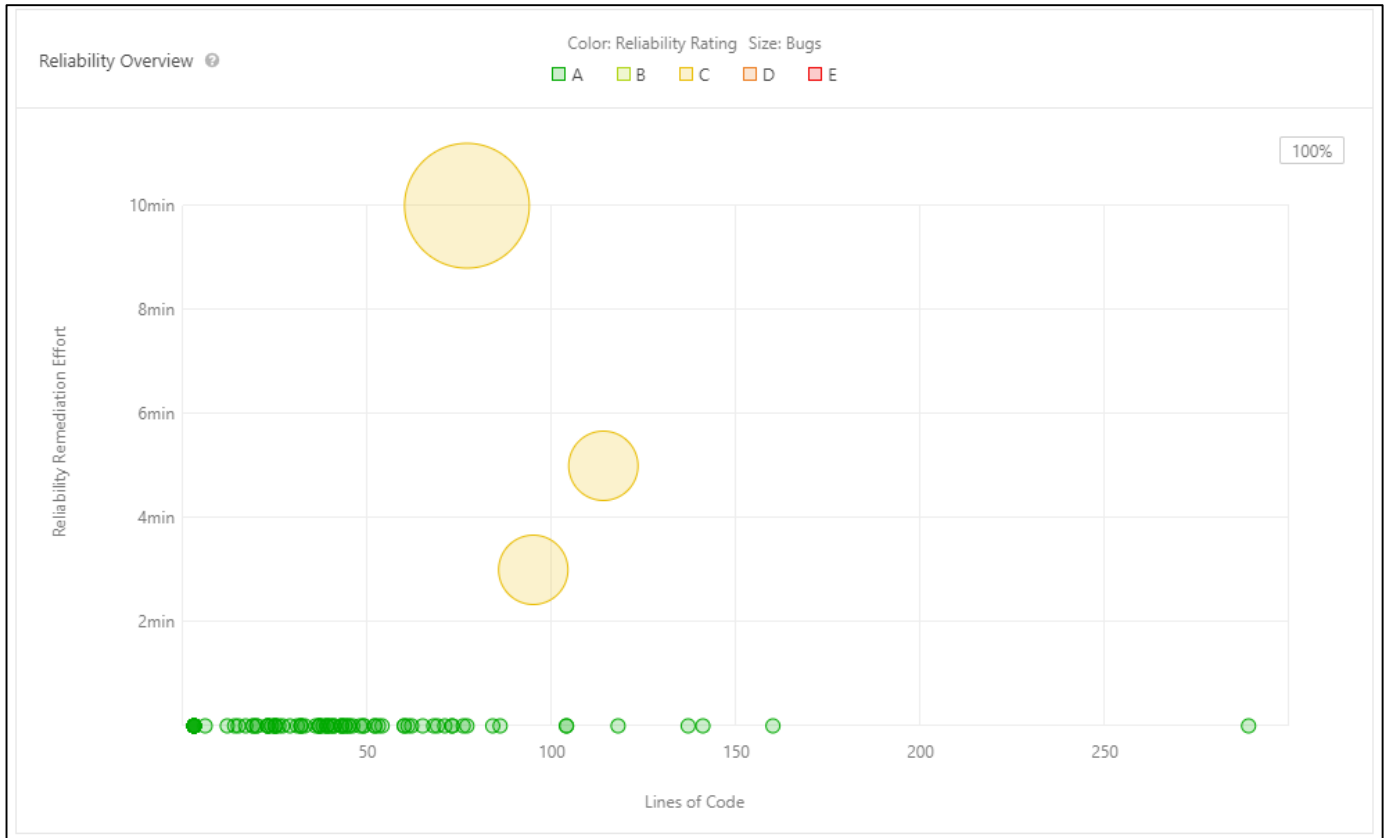


Ilustración 40 Gráfica de fiabilidad. Número de líneas de código - aproximación del coste de arreglo.

```
public Collection<UserJPA> searchUsers(long idUser, String paramName, String paramSurname){
    Query jpaQuery;

    String name = paramName.toLowerCase();//.replaceAll("\\s+", "") removes whitespaces but confuses composed names ('Juan Ca
    String surname = paramSurname.toLowerCase();

    //System.out.println("name = "+name+ " surname= "+surname);

    if(name == "") {

```

Use the "equals" method if value comparison was intended. ... last month L37

🐛 Bug 🚨 Major 🔵 Open Not assigned 5min effort cert, cwe

Ilustración 41 Bug "equals"

```
public void setRol(String rol) {
    rol = rol;

```

Remove or correct this useless self-assignment. ... last month L101

🐛 Bug 🚨 Major 🔵 Open Not assigned 3min effort cert

Ilustración 42 Bug de asignación

```
121 ... public void updatePlayer(int idPlayer, String name, int idTeam, String position, byte [] photo){

```

Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation. ... last month L121

🧑‍🔬 Code Smell 🚨 Critical 🔵 Open Not assigned 5min effort suspicious

Ilustración 43 Recomendaciones de implementación

8.2 Seguridad

Se destaca un único problema considerado grave: la detección de la presencia de una contraseña en el código sin cifrar, en la clase *profileFacadeBean*.

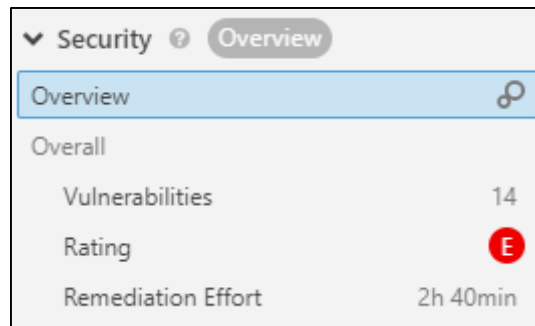


Ilustración 44 Resumen de observaciones de Seguridad

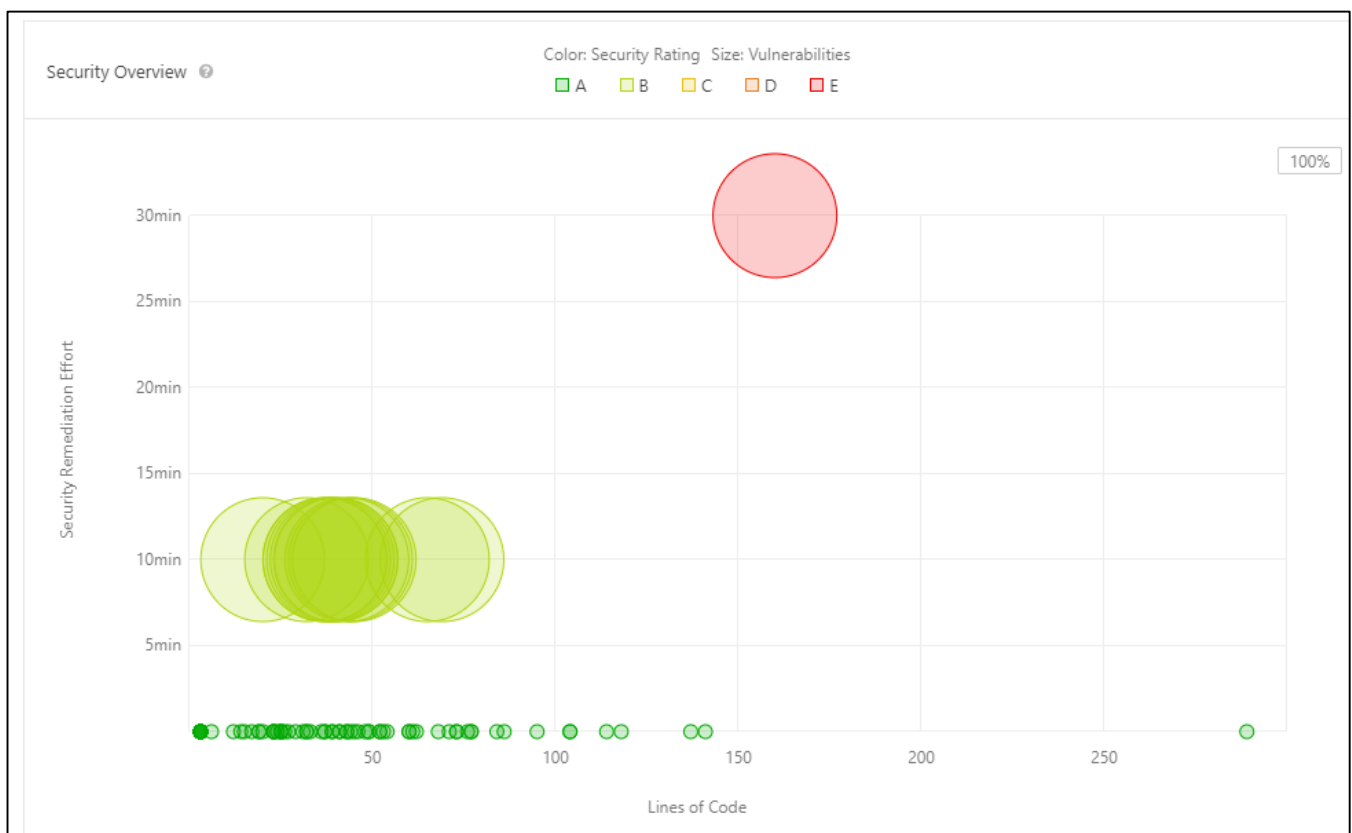


Ilustración 45 Gráfica de seguridad. Líneas de código - tiempo de arreglo

```

116 --      public void sendRegisterMail(String email, String generatedPassword){
117 --          /*SendMail TLS*/
118 --
119 --          String username = "futbolerosWebPortal@gmail.com";
120 --          String password = "password";
    
```

'password' detected in this expression, review this potentially hard-coded credential. 9 days ago ▾ L120 🔗

Vulnerability Blocker Open Not assigned 30min effort cert, cwe, owasp-a2, sans-top25-porous

Ilustración 46 Vulnerabilidad por uso de credenciales en crudo

8.3 Mantenibilidad del código

No se destacan problemas significativos de mantenibilidad.

▼ Maintainability ⓘ	
Overview	🔗
Overall	
Code Smells	310
Debt	4d 1h
Debt Ratio	1.6%
Rating	A
Effort to Reach A	0

Ilustración 47 Resumen de observaciones de mantenibilidad

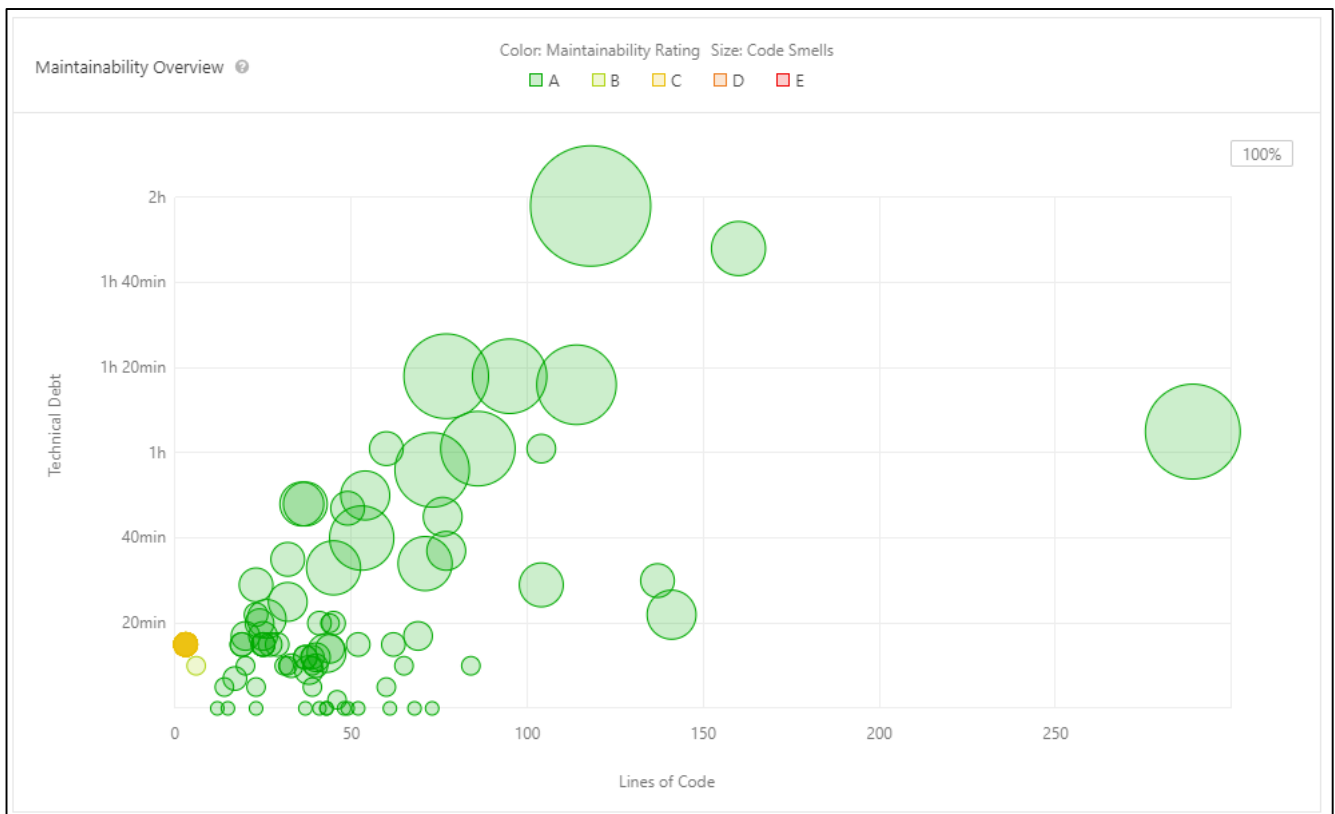


Ilustración 48 Gráfica de observaciones de mantenibilidad. Líneas de código - deuda técnica

8.4 Duplicados

Señala algunos elementos duplicados, destacando la clase *CreatePoll.java*:

▼ Duplications Overview	
Overview	🔗
Overall	
Density	4.5%
Duplicated Lines	320
Duplicated Blocks	12
Duplicated Files	11

Ilustración 49 Resumen de observaciones en cuanto a duplicados en el código

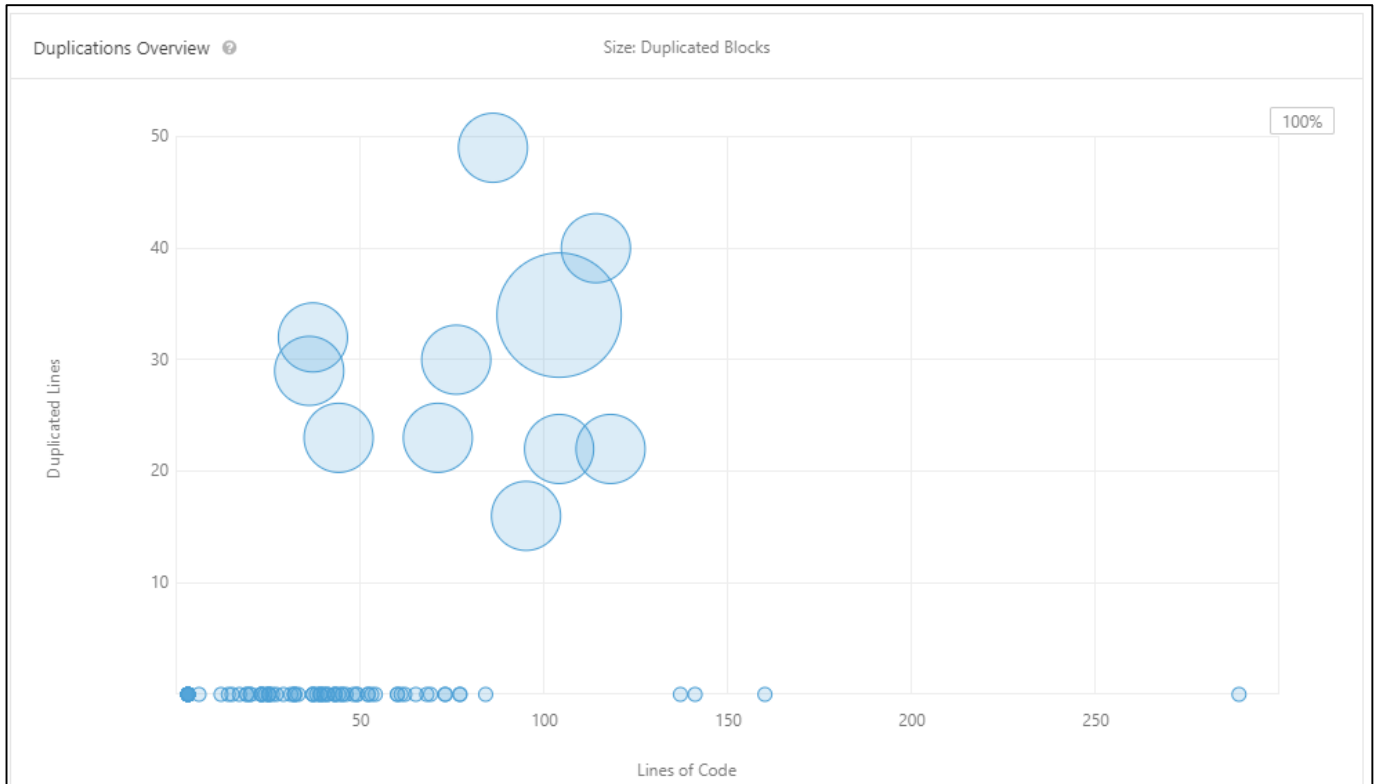


Ilustración 50 Gráfica de duplicaciones. Líneas de código - líneas duplicadas

```

36
37     if(name == "") {
38 ...        .jpaQuery = entman.createQuery("FROM UserJPA u WHERE LOWER(u.surname) LIKE :surname AND u.id <> :idUser ORDER BY u
39 ...        .jpaQuery.setParameter("name", '%'+name+'').setParameter("surname", '%'+surname+'').setParameter( 1 "idUser", i

```

Define a constant instead of duplicating this literal "idUser" 3 times. ... last month ▾ L39 🔗

⊕ Code Smell 🚨 Critical 🔵 Open Not assigned 8min effort 🗑️ design

Define a constant instead of duplicating this literal "surname" 3 times. ... last month ▾ L39 🔗

⊕ Code Smell 🚨 Critical 🔵 Open Not assigned 8min effort 🗑️ design

```

40 ...     }else if (surname == "") {
41 ...        .jpaQuery = entman.createQuery("FROM UserJPA u WHERE LOWER(u.name) LIKE :name AND u.id <> :idUser ORDER BY u.name"
42 ...        .jpaQuery.setParameter("name", '%'+name+'').setParameter("surname", '%'+surname+'').setParameter( 2 "idUser", i
43 ...     }else {
44 ...        .jpaQuery = entman.createQuery("FROM UserJPA u WHERE LOWER(u.name) LIKE :name AND LOWER(u.surname) LIKE :surname A
45 ...        .jpaQuery.setParameter("name", '%'+name+'').setParameter("surname", '%'+surname+'').setParameter( 3 "idUser", i
46 ...     }
47

```

Ilustración 51 Recomendación para no duplicar código

8.5 Líneas de código, número de métodos, comentarios...

▼ Size	
Lines of Code	4,174
Lines	7,040
Statements	1,314
Functions	542
Classes	89
Files	91
Comment Lines	516
Comments (%)	11.0%

Ilustración 52 Resumen de observaciones de Líneas de código, comentarios, sentencias, etc...

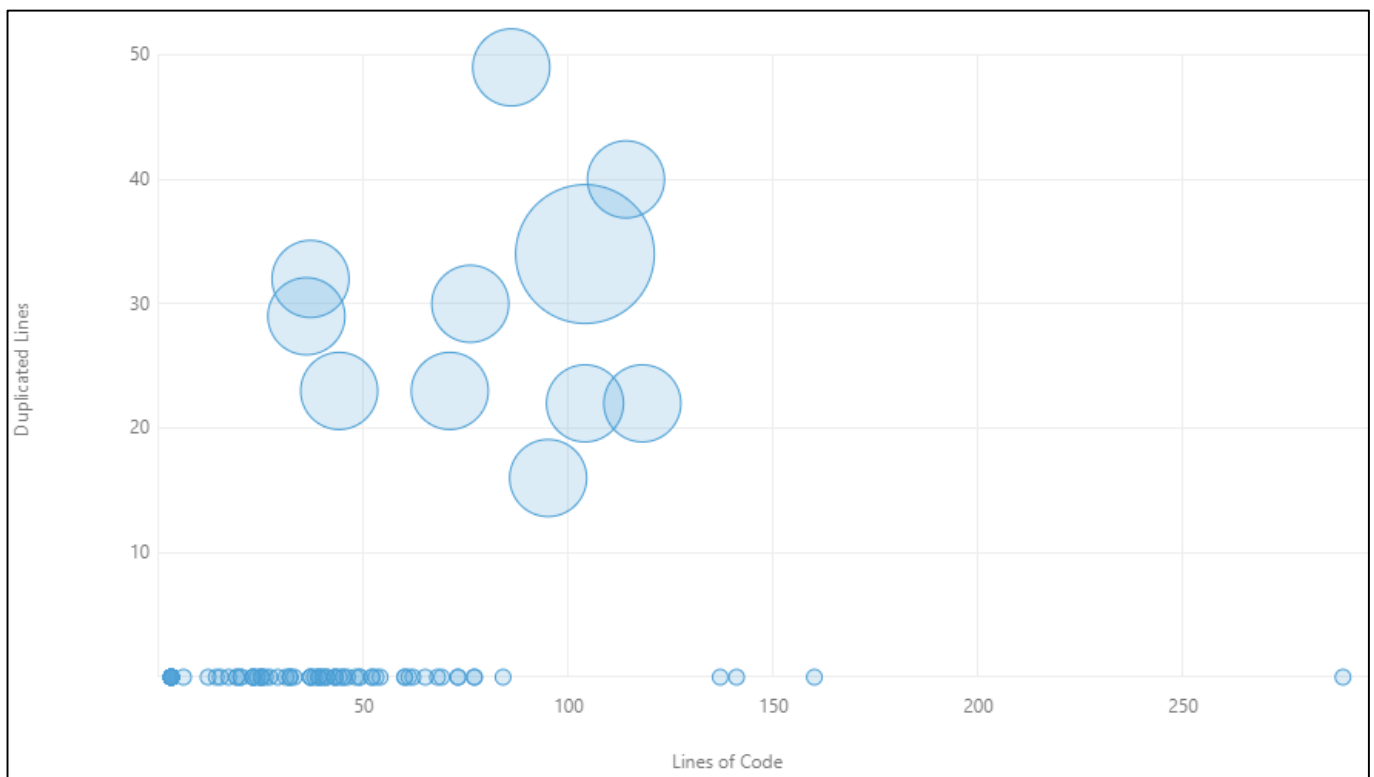


Ilustración 53 Gráfica de líneas de código - líneas duplicadas

8.6 Complejidad ciclométrica

Se destaca el paquete *managedbean.Game* en cuanto a complejidad ciclométrica y es, efectivamente, el paquete claramente más complejo, especialmente el *QuizzController*.

▼ Complexity ? Cyclomatic Complexi...	
Cyclomatic Complexity	627
Cognitive Complexity	241

Ilustración 54 Resumen de complejidad ciclométrica

Cyclomatic Complexity 627

ejb	98
jpa	155
managedbean	296
META-INF	-
utils	78

5 of 5 shown

Ilustración 55 Complejidad ciclomática de los paquetes del primer nivel del proyecto

Cyclomatic Complexity 296

Admin	75
ErrorController	9
Game	118
Profile	62
SocialMedia	32

5 of 5 shown

Ilustración 56 Complejidad ciclomática del paquete *managedBean*

8.7 Conclusiones de calidad del código

Tras el análisis de calidad realizado, se puede concluir que es un código mantenible, no excesivamente complicado, suficientemente comentado y con poca duplicación de código. Se detecta una vulnerabilidad de seguridad y varios problemas menos importantes, que hacen mejorable la fiabilidad del código. No se tienen medidas claras sobre acoplamiento y cohesión del código, métricas mejor especificadas en otras herramientas como *JHawk* y sería interesante valorarlo.

Es recomendable revisar todos los *issues* y *bugs* detectados, destacando especialmente:

1. Ocultar en el código el *password* de Gmail, al menos encriptándolo en el código para dificultar un poco su lectura. Es una cuenta gratuita creada para este proyecto, pero es un problema crítico, especialmente considerando que el código se encuentra disponible al público en GitHub.com. Si se retira el acceso público al código también se mejora la seguridad.
2. Utilizar *equals* para comparar strings en todos los lugares indicados en vez del comparador "==".
3. Revisar la clase *CreatePoll.java* y eliminar elementos duplicados.

9 Instalación en máquina virtual de Google

De forma provisional, se ha desplegado el proyecto en una máquina virtual gratuita de *google*, y se ha creado una cuenta gratuita de correo también en *google* para simular la puesta en producción y realizar las pruebas de usuario.

Se trata de una instancia en *google compute engine* con Windows Server 2012. Se han instalado los componentes necesarios y se han configurado (*Wildfly*, *postgresql* y *java*).

Tipo de máquina n1-standard-1 (1 vCPU, 3,75 GB de memoria)
Plataforma de CPU Intel Haswell

Ilustración 57 Tipo de máquina instanciada en *google cloud compute engine*

A continuación, se ha desplegado el proyecto en la máquina virtual, creado la base de datos, el usuario local de la misma y habilitado el acceso por el puerto 80 (tanto a la máquina virtual como al servidor *Wildfly*, que escucha por defecto el puerto 8080).

De esta manera, se puede acceder al proyecto, sin tener que instalar todo el entorno y desde cualquier dispositivo con acceso a internet, desde la dirección:

<http://35.187.189.135/futboleros/>

9.1 Producción

Para la **puesta en producción** final, será necesario configurar *Wildfly* y contratar servicios de hosting donde poder desplegar el proyecto. Será importante considerar que la base de datos utilizada es *postgresql*. De igual manera, se tendrá que contratar un dominio y disponer de cuentas de correo electrónico. Es aconsejable instalar un certificado SSL para la web. En cuanto a dominios disponibles, el dominio *futboleros.com* y *futboleros.es* están ocupados, por lo que habría que escoger otro nombre o extensión. Por ejemplo, *futboleros.eu* o *futbolerospasion.com*.

Una buena opción de *hosting* son los servicios de *Java Cloud Hosting de Jelastic*, una solución PAAS muy escalable y de alto rendimiento, que incluye certificados SSL gratuitos, es de pago por uso, permite incorporar dominios con facilidad, es compatible con *Wildfly* y *postgresql* y permite el uso opcional de *containers* (<https://jelastic.com/java-cloud-hosting/>).

10 Pruebas de usabilidad IPO

Se adjuntan en los anejos del final de la memoria, las plantillas utilizadas para realizar las entrevistas. A continuación, se muestran los perfiles analizados, los actores probados y los dispositivos de software y hardware analizados.

10.1 Perfiles:

Los perfiles seleccionados deben manifestar tener interés por el mundo del fútbol. Se dividirá por edades en 3 categorías:

- **Joven:** 15-20 años
- **Adulto:** 20-40 años
- **Mayor:** 41-60 años

10.2 Actores:

- Usuario General
- Usuario registrado
- Administrador

10.3 Casos de uso:

Para evitar que las entrevistas sean excesivamente largas, se prescinde de probar la totalidad de casos de uso, seleccionando los más importantes. Se considera innecesario probar las validaciones de los campos, aunque éstas pueden surgir de forma natural en algunos casos.

1. **Usuario general**
 1. Consultar alineación
 2. Registrarse
2. **Usuario registrado**
 3. Login
 4. Logout
 5. Modificación Perfil
 6. Participar en una encuesta
 7. Crear encuesta
 8. Ver resultados de una encuesta
 9. Hacer un quiz
 10. Buscar usuario
 11. Seguir usuario
 12. Ver el historial del usuario
 13. Seguidores / seguidos
 14. Historial
 15. Eliminar cuenta
1. **Administrador**
 16. Crear equipo/Borrar equipo
 17. Consultar alineaciones de equipo
 18. Consultar plantilla equipo
 19. Crear alineación
 20. Modificar alineación

21. Crear nuevo Quiz
22. Añadir pregunta
23. Eliminar pregunta
24. Eliminar encuesta
25. Registrar un administrador

10.4 Entrevistas

Se han realizado un total de 6 pruebas de interacción con 2 usuarios de cada perfil. Todos los usuarios entrevistados manifestaron ser aficionados al fútbol.

Entrevistas añadidas al anejo final (ver punto 14.2).

10.5 Conclusiones IPO

Las entrevistas han resultado muy útiles para detectar puntos de mejora de la aplicación. Se han dividido las observaciones en incidencias críticas, de usabilidad, de diseño y de navegador. La incidencia crítica se ha tratado de inmediato. Las de usabilidad, de diseño y navegador se ordenan por importancia y se anotan para una futura mejora de la aplicación.

Por norma general, la web ha resultado intuitiva y usable. Los usuarios han podido realizar todos los casos de uso probados, aunque con diferencias en cuanto a la facilidad y el tiempo necesario para realizar los mismos. Los usuarios jóvenes y adultos, aparentemente, se manejan con más agilidad en el portal. Los usuarios con más afición a juegos de fútbol también interactúan de forma mucho más intuitiva al crear alineaciones, equipos y demás aspectos concretos de fútbol.

10.5.1 Incidencias críticas

1. Se ha detectado un **bug** cuando se creaba un equipo, se añadía un jugador y se borraba el equipo posteriormente. Este borrado ocasionaba que no se visualizara ninguna de las alineaciones guardadas. Tras un análisis posterior, se pudo comprobar que la tabla *lineup-player* (que contiene la relación jugador-alineación) se borraba por completo en este caso, debido a un error en la sentencia SQL de borrado y se ha solucionado de inmediato.

10.5.2 Observaciones de usabilidad

Usuario sin registrar y usuario registrado

1. Las **indicaciones del sistema** “jugador creado” / “encuesta respondida” /etc... duran poco tiempo y son muy poco perceptibles. Especialmente, en el caso de uso “Registrar un usuario nuevo” donde varios usuarios no advierten que deben revisar el correo para obtener la contraseña.
2. Prácticamente todos los usuarios encuentran **poco intuitivo el botón de crear alineaciones/Quiz/encuestas/etc...** Es un botón que se confunde con la cabecera de la tabla. Sin embargo, todos lo aprenden rápidamente y no resulta un problema después de identificarlo por primera vez.
3. Un usuario advierte que las **encuestas están ordenadas de más antigua a más nueva**, cuando es más lógico que sea al revés.

4. Más de un usuario encuentra inútil que se muestre el botón de “Login” cuando ya se ha registrado. Con menor frecuencia, advierten de la misma circunstancia con el botón “registrarse”, una vez logeado.
5. El caso de uso **logout** ha sido poco intuitivo para varios usuarios, especialmente los usuarios con menos experiencia con redes sociales y portales con Login/logout. La imagen/metáfora de logout no es muy intuitiva.
6. Sería conveniente informar al usuario de **la cantidad de preguntas que tiene un quiz**, ya que ignorarlo genera desorientación e inseguridad y puede conducir a que el usuario abandone el test.
7. Un usuario ha tardado algo más de un minuto en recibir el email de registro. Los usuarios de Gmail con varias cuentas asociadas pueden experimentar este retraso.
8. Las flechas que indican la ordenación de una columna en las tablas, **dan a entender a los usuarios que se pueden ordenar todas las columnas** y al observar que no está implementada esta funcionalidad, se genera frustración. Se debe implementar la funcionalidad o es preferible eliminar la flecha.
9. Un usuario advierte que el botón de añadir opción en una encuesta, debería indicar “añadir opción” y no “añadir pregunta”.
10. El perfil mayor, al buscar a un usuario seguido, primero entra en “mis seguidores” en vez de “mis seguidos”. Sería conveniente poner “mis seguidos” a la izquierda de “mis seguidores”.
11. Se detecta más de un término escrito en inglés cuando debería estar en castellano.
12. Llama un poco la atención la diferencia entre Quiz y Quizz (singular y plural “slang” de Quiz en inglés).

Administrador

1. La mayoría de usuarios ha experimentado **confusión con la opción correcta del Quizz** al añadir una pregunta. El perfil adulto y mayor, especialmente, han entendido que primero se debe especificar la opción correcta, y luego todas las opciones, volviendo a incluir la opción correcta. Bastaría con indicar “opción correcta” y “opciones incorrectas”. Además, resultaría conveniente informar que el sistema las ordenará de forma aleatoria.
2. Algunos usuarios sienten confusión sobre qué link conduce a crear una alineación o a añadir equipos/jugadores. **El link “futbol” no resulta suficientemente intuitivo**. Aun y así, entre las opciones disponibles, todos han navegado correctamente, aunque con inseguridad. Tal vez “equipos” sea más intuitivo.
3. En algunos casos, coincidiendo con los menos aficionados a juegos de futbol, **las alineaciones no se conciben como previsiones de alineación** sino como alineaciones sin más. Es decir, no se conciben como una predicción, sino como una descripción de la alineación que realmente tuvo lugar en la jornada especificada. Sería conveniente, en la página principal, hacer una descripción visual de los contenidos del portal, con atención especial a la predicción de alineaciones. De igual forma, el link “onces” les resulta poco intuitivo para los menos “futboleros”.

10.5.3 Observaciones de diseño

1. El **fondo blanco da un aspecto de inacabado**, sería conveniente romper el blanco con otro color que diferenciase los laterales de la parte central de contenido.
2. Las **indicaciones del sistema de información o error, deberían ser más visibles**. Se percibirían mejor, si durasen más tiempo y si aparecieran en la parte superior central o superior derecha de la pantalla. En cualquier caso, deberían aparecer siempre en el mismo sitio.
3. La atención de los usuarios siempre se centra en las imágenes de escudos o jugadores en primer lugar. Sería conveniente **acercar los links (de creación/borrado/actualización) a esas imágenes y utilizar otras imágenes** para estas funcionalidades, en vez de texto. Además de las imágenes, en las tablas, la atención se centra en el título de la encuesta/Quiz/nombre de equipo situados a la izquierda. Además de acercar links a imágenes, convendría acercarlos a la izquierda de las tablas, más cerca de los títulos de cada fila.
4. La imagen grande que se muestra en la página inicial de la aplicación podría centrarse mejor en el eje vertical y aportar alguna información genérica sobre lo que se puede hacer en el portal.
5. No todas las imágenes de jugadores, escudos de club o fotos de perfil tienen los **bordes redondeados**, sería conveniente unificar criterios y redondearlos todos.

10.5.4 Incidencias relacionadas con el navegador

1. El perfil mayor ha experimentado cierta incomodidad al no poder navegar hacia atrás con el navegador. Curiosamente, el resto de perfiles no ha utilizado el botón atrás en ningún momento.
2. En Microsoft edge, existe la opción de no volver a mostrar el cuadro de diálogo de confirmación y si se activa, se impide poder borrar elementos al no poder confirmar si se desea eliminar el elemento (porque no aparece el cuadro una vez activada la opción). Por este motivo, sería conveniente recomendar el uso de Firefox, Safari o Chrome para el portal, o diseñar un cuadro de diálogo usando otra implementación.

11 Conclusiones del proyecto

La realización de este proyecto ha permitido valorar la importancia de una planificación previa, con estudios de mercado, definición de objetivos, requisitos, casos de uso y demás aspectos anteriores a la implementación. **El proceso de implementación se dirige mejor y resulta más efectivo gracias a las definiciones previas**, permitiendo ordenar mejor el código y trabajar de una forma más estructurada y sistemática.

El alcance del trabajo, con la priorización de secciones y la planificación temporal, también ha permitido trabajar de forma secuencial y teniendo siempre presente el tiempo disponible. Se han tomado decisiones en este sentido en la fase final del proyecto, dando preferencia a las pruebas de usabilidad y de calidad de código, a la implementación del foro, por ejemplo. Sin embargo, para facilitar su futura implementación, el foro tiene definidas en las interfaces y sus implementaciones, las firmas de las funciones necesarias, las entidades JPA, los requisitos y casos de uso, los diagramas de componentes etc...

Se considera importante que la aplicación contenga la sección de “foro” en un futuro, así como un sistema de puntuaciones para los usuarios. Ambas funcionalidades enriquecerán la red social y resulta muy recomendable añadirlas para dinamizar la aplicación. Su implementación ha sido descartada debido a las restricciones de tiempo en la fase de implementación.

Existen otros elementos que se han implementado, pero que requieren ser perfeccionados. En primer lugar, el diseño responsivo de las tablas de encuestas y quizz para los usuarios. Se requiere eliminar algunas columnas para una mejor visualización en dispositivos móviles. Se ha implementado el diseño responsivo en las tablas de los administradores, por lo tanto, la estrategia a seguir está implementada, así que el coste en tiempo requerido para aplicarlo a los usuarios, será bajo.

En segundo lugar, se ha utilizado la Api de Java para enviar correos electrónicos (Java Mail), generando una contraseña para los usuarios cuando se registran. Sería conveniente usar la misma Api para recuperar contraseña si un usuario la olvida. De igual forma, el coste de implementación será bajo, porque ya hay un modelo a seguir.

Durante la fase de implementación, se ha podido constatar como la planificación y definición previa de los Quiz ha sido muy alterada. Las estructuras de datos necesarias, las funciones de negocio, así como las vistas y controladores utilizados, han sido significativamente más numerosos y complejos de lo previsto. Ello ha ocasionado que el subsistema *Game* acabara excesivamente cargado. Hubiera sido conveniente, subdividir este subsistema en otros dos subsistemas diferentes.

La aplicación está en condiciones para la puesta en producción. Sin embargo, antes de la puesta en producción se debe configurar *Wildfly*, ya que está en modo desarrollo y atender a las siguientes indicaciones:

- Corregir las incidencias detectadas y no tratadas en las pruebas de usuario IPO.
- Atender a las recomendaciones de calidad de código más importantes detectadas con la herramienta SonarQube.
- Implementar “recuperar contraseña” para usuarios.
- Codificar el diseño responsivo de las tablas para los usuarios.

Se calcula que serían suficientes 5 días a jornada completa (del programador que ha desarrollado la aplicación) para resolver estos puntos.

De cara a las **líneas de trabajo futuro**, se distingue entre modificaciones a corto plazo, y modificaciones a largo plazo. A corto plazo, se contempla:

1. Implementar el sistema de puntuaciones para usuarios.
2. Implementar el foro e introducir modificaciones necesarias en el historial del usuario incluyendo participaciones en el foro.
3. Permitir a usuarios no registrados participar en encuestas.
4. Realizar pruebas de usuario con nuevas funcionalidades y con diseño responsivo.

Se calcula que serían necesarios aproximadamente 8-10 días (del programador que ha desarrollado la aplicación) para implementar estos puntos.

A largo plazo y observando la evolución de la aplicación web, se señalan las siguientes líneas de trabajo:

1. Incluir una sección de “valoración de jugadores”, donde los usuarios puedan poner nota a los jugadores en función de su criterio.
2. Incluir una sección “porra de la semana” para que los usuarios puedan participar en una porra para cada jornada, gestionada por el administrador.
3. Valorar exponer como servicio web las encuestas y sus resultados.
4. Implementar una capa adicional de seguridad JAAS. Para implementar JAAS se debe modificar la estructura de datos que almacena usuarios y administradores, por lo que se requiere modificar el modelo de datos de la aplicación.
5. En las tablas, incluir filtros de búsqueda, ordenación por columnas y agrupar resultados por grupos de filas.
6. Introducir la posibilidad de mostrar los logros conseguidos en *futboleros* en otras redes sociales como Facebook o Twitter.
7. Realizar pruebas de usuario con nuevas funcionalidades y con diseño responsivo.

Se calcula que serían necesarios aproximadamente 18-22 días (del programador que ha desarrollado la aplicación) para implementar estos puntos.

Finalmente, es importante destacar la figura del **administrador**, encargado de gestionar el portal. La aplicación está preparada para ser gestionada de forma sencilla, por lo que el administrador no requiere tener una formación técnica para el manejo de la misma. En un futuro, podría valorarse contratar servicios web de datos deportivos para que el administrador pueda desentenderse de gestionar los equipos y jugadores, dedicándose, solo a la predicción de alineaciones en el apartado de fútbol. De esta manera se podría extender la aplicación a más ligas de fútbol.

12 Bibliografía

SonarQube. *Analyzing with SonarQube Scanner for Ant*:

<https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner+for+Ant>

JBoss Developer. *Running JBoss on Port 80 or 443*:

<https://developer.jboss.org/wiki/RunningJBossOnPort80Or443>

Google. *Google compute Engine*:

<https://cloud.google.com/compute/>

Digital Ocean. *How to Import PostgreSQL Databases with pg_dump*:

<https://www.digitalocean.com/docs/databases/how-to/postgresql/import-databases/>

Kingsware. Johannes Neubauer. *Ordering Collections with JPA*:

<https://www.kingsware.de/2011/03/15/ordering-collections-with-jpa/>

GeeksforGeeks. *HTML | Viewport meta tag for Responsive Web Design*:

<https://www.geeksforgeeks.org/html-viewport-meta-tag-for-responsive-web-design/>

JavaWorld. Steven Haines. *Java persistence with JPA and Hibernate, Part 1: Entities and relationships*:

<https://www.javaworld.com/article/3373652/java-persistence-with-jpa-and-hibernate-part-1-entities-and-relationships.html>

JBoss. Documentación de JBoss:

<https://docs.jboss.org>

Oracle. Documentación de Oracle:

<https://docs.oracle.com/en/>

Coderanch. Alan Smith. *@Named vs @ManagedBean*:

<https://coderanch.com/t/596701/java/Named-ManagedBean>

Reddit. *Migrate your JSF ManagedBean to CDI Named annotations*:

https://www.reddit.com/r/java/comments/1pls2i/migrate_your_jsf_managedbean_to_cdi_named/

Jelastic. *Jelastic Java Cloud Hosting*:

<https://jelastic.com/java-cloud-hosting/>

Arsys. Dominios Arsys:

<https://www.arsys.es/dominios/buscar?dom=futbolerospasion.com>

13 Glosario

Actor

Se le llama actor a toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos, además de entidades abstractas, como el tiempo. 10

Arquitectura cliente-servidor

La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras. 28

Arquitectura de componentes distribuidos

Un sistema distribuido se define como una colección de computadoras separadas físicamente y conectadas entre sí por una red de comunicaciones 28

Arquitectura por capas

La programación por capas es un modelo de desarrollo *software* en el que el objetivo primordial es la separación (desacoplamiento) de las partes que componen un sistema *software* 28

backing beans

Clases java especiales a las cuales se les asocia un componente de interfaz de usuario 48

Command

Este patrón permite solicitar una operación a un objeto sin conocer realmente el contenido de esta operación, ni el receptor real de la misma. Para ello se encapsula la petición como un objeto, con lo que además facilita la parametrización de los métodos. 30

complejidad ciclomática

La Complejidad Ciclomática (en inglés, Cyclomatic Complexity) es una métrica del software en ingeniería del software que proporciona una medición cuantitativa de la complejidad lógica de un programa..... 48

diagrama de clases UML

En ingeniería de software, un diagrama de clases en Lenguaje Unificado de Modelado (UML) es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos. 27

diseño responsivo

El diseño web adaptable (también diseño web adaptativo o responsivo 8

EJB

Las Enterprise JavaBeans (también conocidas por sus siglas EJB) son una de las interfaces de programación de aplicaciones (API) que forman parte del estándar de construcción de aplicaciones empresariales J2EE (ahora JEE) de Oracle Corporation (inicialmente desarrollado por Sun Microsystems). 46

entidades JPA

Java Persistence API, más conocida por sus siglas JPA, es la API de persistencia desarrollada para la plataforma Java EE. Es un framework del lenguaje de programación Java que maneja datos relacionales en aplicaciones usando la Plataforma Java en sus ediciones Standard (Java SE) y Enterprise (Java EE). 46

fetch EAGER

La carga diferida (en inglés lazy loading), es un patrón de diseño comúnmente usado en la programación informática que consiste en retrasar la carga o inicialización de un objeto hasta el mismo momento de su utilización. Esto contribuye a la eficiencia de los programas, evitando la precarga de objetos que podrían no llegar a utilizarse. El opuesto de la carga diferida es la carga previa, precarga o eager loading..... 49

framework JSF

JavaServer Faces es el framework oficial de Java Enterprise para el desarrollo de interfaces de usuario avanzadas en aplicaciones web..... 46

FrontController

Es un patrón de diseño que se basa en usar un controlador como punto inicial para la gestión de las peticiones 30

GanttProject

anttProject es un programa de código abierto con licencia GPL escrito en Java con la biblioteca Swing, su objetivo es la administración de proyectos usando el diagrama de Gantt 23

google compute engine

Google Compute Engine es un servicio de google que ofrece máquinas virtuales que se ejecutan en los centros de datos de Google y están conectadas a través de una red de fibra a nivel mundial. Sus herramientas y su flujo de trabajo permiten escalar desde instancias individuales hasta un entorno de cloud computing global con balanceo de carga. 58

IPO

es la disciplina que estudia el intercambio de información entre las personas y los ordenadores. Su objetivo es que este intercambio sea más eficiente
 minimiza los errores, incrementa la satisfacción, disminuye la frustración y en definitiva, hace más productivas las tareas que envuelven a las personas y los ordenadores. 5

JAAS

Java Authentication and Authorization Service, o JAAS 64

Java EE

Java Platform, Enterprise Edition o Java EE (anteriormente conocido como Java 2 Platform, Enterprise Edition o J2EE hasta la versión 1.4 5

jQuery

jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. 49

metáfora

consiste en expresar una idea valiéndose de otra, con la que mantienen analogía o semejanza . También se entienden como la aplicación de una palabra o de una expresión a un objeto o a un concepto, con el fin de sugerir una comparación 61

Named CDI Beans

CDI es el framework de inyección de dependencia y administración de beans, lanzado como parte del Java EE 6 que incluye una completa, integral y fácil administración de beans. Los beans CDI son mas avanzados y flexibles que los beans JSF, principalmente por los interceptores y ámbitos de conversación(conversation), eventos, decoradores y mas 46

patrón facade

es un tipo de patrón de diseño estructural. Viene motivado por la necesidad de estructurar un entorno de programación y reducir su complejidad con la división en subsistemas, minimizando las comunicaciones y dependencias entre estos 31

patrón MVC

es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario 29

RM-ODP

el modelo de referencia de procesamiento abierto y distribuido RMODP proporciona un marco de coordinación para la normalización del desarrollo de aplicaciones abiertas y distribuidas, creando una arquitectura capaz de soportar de forma integrada aspectos tales como la distribución, interoperabilidad o portabilidad de los sistemas, objetos y componentes 5

servlet

El servlet es una clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor. Aunque los servlets pueden responder a cualquier tipo de solicitudes, estos son utilizados comúnmente para extender las aplicaciones alojadas por servidores web, de tal manera que pueden ser vistos como applets de Java que se ejecutan en servidores en vez de navegadores web 46

14Anejos

14.1 Plantillas de entrevistas

14.1.1 Instrucciones para usuario:

1. **Preguntas genéricas:**
 - a. ¿Visitas páginas web deportivas?
 - b. ¿Has jugado a *comunio* o *futmondo* o similar?
 - c. ¿Sabes lo que es *comuniazo* o *futbolFantasy*?
2. **Test futboleros:**

USUARIO Y USUARIO REGISTRADO

1. Accede a: <http://35.187.189.135/futboleros/>
2. Consulta la próxima alineación del FC Barcelona.
3. Regístrate en el sistema con tu email, si lo prefieres podemos usar el siguiente: "futboleros+minombre@gmail.com". Consulta correo (con indicaciones).
4. Auténticate en el sistema, haz login
5. Haz logout
6. Cambia tu nombre por Antonio
7. Participa en la encuesta "¿Es Messi una deidad?"
8. Crea una encuesta del mundo del futbol con 3 opciones cualesquiera
9. Realiza el Quiz "estadísticas de la liga BBVA (18/19)"
10. Busca al amigo "Alexis"
11. Seguir a Alexis y volver a pantalla principal
12. Dejar de seguir a Alexis
13. Ver Historial de Alexis
14. Eliminar cuenta

ADMINISTRADOR

1. Ingresa al sistema como administrador, con usuario "admin@admin.com" y contraseña "admin"
2. Crea un equipo "MiNombre FC"
3. Añade jugador "Minombre"
4. Crea una previsión de alineación para el FC Barcelona
5. Borra equipo "MiNombre FC"
6. Consulta alineaciones de Real Madrid
7. Modifica última alineación del FC Barcelona
8. Crea un quiz titulado "futboleros" y haz clic en guardar y añadir pregunta
9. Añadir pregunta "¿Quién es el máximo goleador de la liga inglesa?" Con opciones: Kane, Agüero y Messi. La respuesta correcta es Kane.
10. Ahora eres administrador. Elimina la encuesta que creaste antes como usuario.
11. Elimina el quizz que creaste hace un momento.
12. Registra un nuevo administrador "minombre@admin.com"

14.1.2 Anotaciones:

Anotaciones introductorias: Perfil, nombre, edad, navegador, descripción del dispositivo, resolución de pantalla.

Describir red social de forma muy genérica y:

- Pedir espíritu crítico y no sentir que está siendo examinado, todo lo contrario.
- Informar de que no se va a ayudar ni dar indicaciones para realizar casos de uso, es necesario observar la interacción.

Leer instrucciones del test al usuario y anotar observaciones.

Preguntas finales:

- ¿Qué le ha parecido la presentación en términos generales?
- ¿Le ha resultado usable el sistema en términos generales?

Agradecer su colaboración.

14.2 Entrevistas

14.2.1 Usuario 1

Tabla 20 Ficha de la entrevista al usuario 1

Medio de entrevista	Presencial
Nombre del usuario	Daniel Ripoll Vitini
Perfil	Adulto
Profesión	Psicólogo
Edad	34
Dispositivo	PC. Linux Debian 9.0
Navegador	Firefox Quantum 60
Resolución de pantalla	1920 x 1080
Visita prensa deportiva digital	No
Ha jugado a comunio / conoce comuniaz	Si / Si

Observaciones de interacción con futboleros

Como Usuario y usuario sin registrar:

1. Le **parece inseguro recibir una contraseña por correo** y cree que sería bueno recomendar de forma más clara que hay que cambiar la contraseña y/o borrar el email.
2. El usuario advierte **incomodidad con el botón de crear** (encuesta/Quizz/equipo etc.): parece la cabecera de la tabla y no un botón de crear. Sin embargo, indica que, al ser igual en todos lados, se acostumbra rápido.
3. Detecta que, al añadir una opción en una encuesta, el botón tiene el texto "añadir pregunta" y **debería decir "añadir opción"**.
4. Le **cuesta encontrar la forma de hacer logout**, le resulta poco intuitiva.

Como Administrador:

5. **Detecta un bug:** ¡al crear un equipo, crear un jugador y borrar el equipo posteriormente, se dejan de mostrar todas las alineaciones guardadas!
6. Entiende bien y rápido cómo crear una alineación, pero no lo asocia con el concepto de crear una previsión de alineación.
7. Al crear un Quizz. Primero introduce la correcta, y luego todas las opciones volviendo a introducir la correcta.

Observaciones generales de usabilidad:

8. Le resulta, en general, una web intuitiva, no ha habido ningún caso de uso que no supiera hacer o que le requiriera un tiempo excesivo.

14.2.2 Usuario 2

Tabla 21 Ficha de la entrevista al usuario 2

Medio entrevista	Virtual – Skype con escritorio remoto
Nombre del usuario	Álvaro Úbeda Ripoll
Perfil	Joven
Profesión	Estudiante de Ingeniería Industrial
Edad	21
Dispositivo	PC. Windows 10 Pro
Navegador	Google Chrome
Resolución de pantalla	1920 x 1080
Visita prensa deportiva digital	No
Ha jugado a comunio / conoce comuniaz	No (pero lo conoce) / No

Observaciones de interacción con futboleros

Como Usuario y usuario sin registrar:

1. Al registrarse como nuevo usuario no advierte la indicación de que la contraseña se ha enviado a su email.
2. El usuario advierte mucha **incomodidad con el botón de crear** (encuesta/Quizz/equipo etc.): parece la cabecera de la tabla y no un botón de crear. Cuando lo identifica, no resulta un problema en adelante.
3. Le **cuesta encontrar la forma de hacer logout**, le resulta poco intuitiva.

Como Administrador:

4. Para crear un equipo, experimenta cierta desorientación. **No resulta intuitivo el link “fútbol”** donde se encuentra la opción de crear equipo.
5. Entiende bien y rápido cómo crear una alineación, pero no lo asocia con el concepto de crear una previsión de alineación.

Observaciones generales de usabilidad:

6. Le resulta, en general, una web intuitiva, no ha habido ningún caso de uso que no supiera hacer o que le requiriera un tiempo excesivo.
7. En las tablas, las opciones de borrar/añadir/modificar se encuentran demasiado lejos del nombre haciéndolo incómodo. Observa que podría mejorar si se acercan las opciones al nombre que se busca en la tabla.

14.2.3 Usuario 3

Tabla 22 Ficha de la entrevista al usuario 3

Medio entrevista	Virtual – Skype con escritorio remoto
Nombre del usuario	Albert Winoc García
Perfil	Adulto
Profesión	Diseñador gráfico
Edad	32
Dispositivo	MacBook Pro (macOS Mojave 10.14.5)
Navegador	Google Chrome
Resolución de pantalla	2560 x 1600 (13,3’’)

Visita prensa deportiva digital
Ha jugado a comunio / conoce comuniaz

Si
Si / Si Juega en futmondo, utiliza futbolFantasy

Observaciones de interacción con futboleros

Como Usuario y usuario sin registrar:

1. Al registrarse como nuevo usuario advierte que debe consultar el email, pero le resulta incómodo que la advertencia desaparezca tan rápido.
2. El email tarda algo más de 1 minuto en llegar. Los usuarios que tienen varias cuentas asociadas en google tendrán esta problemática.
3. El usuario reconoce el **botón de crear** (encuesta/Quizz/equipo etc..) pero indica que se confunde con la cabecera de la tabla.
4. El usuario encuentra bastante rápido la forma de hacer logout, a diferencia de la mayoría.
5. El usuario advierte más de un término en inglés cuando debería estar en castellano.
6. Al realizar el quiz, el usuario indica que sería conveniente **informar del número de preguntas** que contiene el quiz.
7. Las flechas que indican la ordenación de las tablas molestan al usuario porque le hacen entender que se puede ordenar por categorías, cuando esa función no está implementada, causando frustración. Opina que es preferible ordenar por el criterio que sea, pero sin flecha.
8. El usuario advierte que las encuestas están ordenadas en sentido inverso al que resulta más útil. (es más útil mostrar las encuestas más recientes primero)
9. Le sorprende que sea tan intuitivo y fácil eliminar una cuenta, no es habitual y podría ser perjudicial a nivel de negocio.

Como Administrador:

10. Para crear un equipo, experimenta cierta desorientación. No resulta intuitivo el link "futbol" donde se encuentra la opción de crear equipo.
11. Al crear un quiz se le solicita poner 3 opciones. Escribe la respuesta correcta, y después 3 opciones más, entre las que vuelve a incluir la respuesta correcta, repitiendo la misma y obteniendo un error.

Observaciones generales de usabilidad:

12. Le resulta, en general, una web intuitiva, no ha habido ningún caso de uso que no supiera hacer o que le requiriera un tiempo excesivo, salvo la recepción del email de registro.
13. Una vez logeado en el sistema, el usuario quitaría las opciones de registrar y Login, le parece que no tienen sentido.

Observaciones generales de diseño gráfico:

14. El fondo blanco o casi blanco da un aspecto que parece inacabado o de Word, recomienda romper el fondo blanco con algún color que lo diferencie de la parte central, donde se muestra el contenido.
15. Destaca que los errores/confirmaciones del sistema deberían mostrarse más tiempo y de forma más visible (en la parte superior centrados o en la parte superior derecha)

14.2.4 Usuario 4

Tabla 23 Ficha de la entrevista al usuario 4

Medio entrevista	Virtual – Skype con escritorio remoto
Nombre del usuario	Inés Rodríguez Castillo
Perfil	Joven
Profesión	Estudiante de Biotecnología
Edad	19

Dispositivo	MacBook Pro (desconoce modelo)
Navegador	Safari
Resolución de pantalla	1920 x 1080
Visita prensa deportiva digital	No
Ha jugado a comunio / conoce comuniaz	No / No

Observaciones de interacción con futboleros

Como Usuario y usuario sin registrar:

1. Al registrarse como nuevo usuario no advierte la indicación de que la contraseña se ha enviado a su email.
2. Le cuesta un poco encontrar el botón de logout, pero menos que la media.
3. Al seguir la orden "buscar amigo Alexis" tarda más tiempo de lo normal en entrar en "Buscar Usuario" ya que deduce primero que debe estar en mis seguidos.

Como Administrador:

4. Tarda un tiempo muy superior a lo normal en encontrar la forma de crear una nueva alineación, no le resulta intuitivo el link "onces". Influye el hecho de ser poco aficionada al futbol.
5. Al seguir la instrucción "borrar equipo", borra el jugador en vez del equipo. Se le ha avisado de ello y ha encontrado rápido la opción de borrar.
6. Al añadir una pregunta al Quizz, añade la respuesta correcta y luego introduce el resto de opciones volviendo a introducir la respuesta correcta.

Observaciones generales de usabilidad:

7. El usuario ha acusado el hecho de no ser muy aficionado al futbol, resultándole poco intuitivo algunos aspectos relacionados con el fútbol. Otros aspectos como hacer Login y logout, sí le resultan intuitivos por afición a redes sociales.
8. Como la mayoría de usuarios, centra mucho su atención primero en imágenes, luego en texto.

14.2.5 Usuario 5

Tabla 24 Ficha de la entrevista al usuario 5

Medio de entrevista	Presencial
Nombre del usuario	Francisco Javier García-Mansilla
Perfil	Mayor
Profesión	Ingeniero de caminos
Edad	59
Dispositivo	PC. Windows 10 Pro
Navegador	Microsoft Edge
Resolución de pantalla	1280 x 1024
Visita prensa deportiva digital	No
Ha jugado a comunio / conoce comuniaz	No / No

Observaciones de interacción con futboleros

Como Usuario y usuario sin registrar:

9. A diferencia de otros usuarios, identifica rápido el botón de "crear" alineaciones/encuestas/Quizz y advierte bien que debe buscar en el correo la contraseña al registrar usuario.
10. Le cuesta encontrar el botón de logout, le resulta poco intuitivo.

11. Para encontrar el usuario que ha seguido, primero accede a mis seguidores, en vez de mis seguidos, puede que valga la pena poner “mis seguidos” a la izquierda.
12. Al modificar perfil, tarda un poco en hacer clic en modificar, ya que hay que el botón no aparece en la pantalla hasta que se mueve el scroll hacia abajo.
13. Cuando se borran elementos, aparece un cuadro de diálogo de confirmación. Edge da la opción de no volver a mostrar dicho cuadro de diálogo. ¡Si se activa dicha opción, no se pueden borrar elementos en el futuro, ya que no se puede confirmar!

Como Administrador:

14. Administra sorprendentemente bien el portal sin incidencias de usabilidad destacables.

Observaciones generales de usabilidad:

15. Le ha resultado agradable interactuar con el sistema, salvo el logout, ha realizado todos los casos de uso rápido, resultándole, en general, todo bastante intuitivo.
16. En general, se puede observar, como siempre centra mucho su atención en las imágenes, por un lado, y el título descriptivo de cada fila en las tablas, situado a la izquierda. Se acentúa mucho, por lo tanto, la dirección occidental de percibir/leer un documento (de arriba abajo y de izquierda a derecha).
17. Utiliza, en alguna ocasión el botón del navegador “atrás” causándole cierta desorientación e incomodidad al no poder usarlo.

14.2.6 Usuario 6

Tabla 25 Ficha de la entrevista al usuario 6

Medio de entrevista	Presencial
Nombre del usuario	Trinidad Díez Lasheras
Perfil	Mayor
Profesión	Administrativa
Edad	57
Dispositivo	PC. Windows 10 Pro
Navegador	Chrome
Resolución de pantalla	1920 x 1080
Visita prensa deportiva digital	No
Ha jugado a comunio / conoce comuniaz	No / No

Observaciones de interacción con futboleros

Como Usuario y usuario sin registrar:

1. No percibe, a través de la advertencia del sistema que debe ir a buscar al email la contraseña generada al registrarse como usuario.
2. Le cuesta mucho encontrar el botón de logout, le resulta muy poco intuitivo.
3. Para encontrar el usuario que ha seguido, primero accede a mis seguidores, en vez de mis seguidos, puede que valga la pena poner “mis seguidos” a la izquierda.
4. Para ver el historial de un usuario, no le resulta intuitivo el link “ver” al buscar un usuario. Ello hace que le resultase muy difícil hallar finalmente la opción “historial de usuario”.

Como Administrador:

5. *Para crear una nueva alineación, no le resulta intuitivo el link futbol y accede a la opción genérica alineaciones primero.*
6. *Al crear una alineación, no está segura si la ha creado y vuelve a hacer clic en crear alineación, creando una segunda alineación idéntica a la anterior.*
7. *Al crear un quiz, como muchos otros usuarios, ingresa la opción correcta y vuelve a ingresar la misma opción correcta después mezclada entre las incorrectas.*

Observaciones generales de usabilidad:

8. *En general, le ha resultado una interacción intuitiva y ha podido realizar todos los casos de uso. Si bien, en este usuario se ha observado significativa frustración con el caso de uso "seguir a usuario". Al no resultarle intuitivo el link "ver" en "buscar usuario" ha navegado en otras direcciones tardando casi 1 minuto en encontrar el camino.*
9. *En general, se puede observar, centra mucho su atención en las imágenes, por un lado, y el título descriptivo de cada fila en las tablas, situado a la izquierda. Se acentúa mucho, por lo tanto, la dirección occidental de percibir/leer un documento (de arriba abajo y de izquierda a derecha).*
10. *Utiliza, en alguna ocasión el botón del navegador "atrás" causándole cierta desorientación e incomodidad el no poder usarlo.*