

Práctica 4:

Diseño, desarrollo e instalación de un sistema de información Web con persistencia de datos



Andrea Ferradas Agudo

Jorge Hernández Aznar

Mario Clavero Artal

Sistemas de Información 2024/2025

Índice

Introducción	3
Aspectos de diseño y técnicos	3
Configuración del entorno de trabajo	3
Capa de persistencia de datos -> VOs y DAOs	4
Capa de modelo y capa de interfaz	5
Metodología de trabajo	6
Herramientas utilizadas	6
Organización y distribución del trabajo en equipo	6
Horas de trabajo	6
Decisiones tomadas	7
Dificultades encontradas	8
Plan de pruebas de usuario	9
Manual del usuario	10
Perfil de Administrador	10
Perfil de Público Objetivo	10
Plan de mantenimiento	13
Seguridad del sistema	14
Accesibilidad y usabilidad del sistema	14

Introducción

Hemos decidido desarrollar un sistema que guarde la información de los partidos de baloncesto que se lleven a cabo a través de la Federación Española de Baloncesto. Se gestionará información de las competiciones, equipos y jugadores que participen.

Las principales funcionalidades en las que el usuario más interactúa serán realizar y publicar comentarios sobre partidos concretos y además los propios jugadores tendrán un chat propio para mandar mensajes a sus seguidores. A parte de estas funcionalidades, el usuario podrá añadir a favoritos sus jugadores, equipos y competiciones de las cuales podrá obtener información detallada.

Los principales perfiles de usuario serán el público objetivo y el administrador. En el público objetivo se pueden distinguir dos subtipos, los jugadores que tendrán su propio chat para mandar mensajes, y los seguidores que son simplemente aficionados que podrán comentar los partidos a su vez. La figura del administrador se encargará de introducir la información necesaria sobre jugadores, equipos, competiciones y partidos.

Aspectos de diseño y técnicos

Configuración del entorno de trabajo

Para la realización del trabajo hemos utilizado Eclipse Java Developers como IDE (Entorno de desarrollo integrado) principal. De esta forma hemos desarrollado las pantallas mediante archivos jsp y un stylesheet css. Para la Base de Datos hemos usado PostgreSQL y para gestionarla con más facilidad usamos DBeaver. Después, como servidor hemos utilizado Apache Tomcat.

Inicialmente al comenzar nos topamos con algunos problemas con el servidor Apache Tomcat debido a que no se lanzaba correctamente y no era accesible. Finalmente pudimos comprobar que este problema se debía a la versión del Dynamic Web Module ya que tenía que ser la versión 4.0 o inferior. Para gestionar la base de datos se instaló PostgreSQL junto con su driver correspondiente.

El proyecto fue planteado de tal manera que cada uno de los miembros teníamos nuestra propia base de datos local, de modo que tuvimos que configurar una contraseña común para el usuario postgres para todos de forma que pudiéramos acceder todos a nuestras respectivas bases locales. Esta configuración se almacena en el fichero *content.xml*. Del mismo modo nos instalamos la extensión DBeaver en Eclipse para poder crear y manejar la base de datos con mayor facilidad.

Por último para poder trabajar de forma conjunta en el proyecto se creó un repositorio de GitHub para poder trabajar entre todos los miembros del grupo de forma conjunta. Este repositorio fue clonado en los equipos de los miembros y vinculado con el workspace de cada uno para que pudiéramos recibir los cambios de los compañeros y subir los propios directamente.

Capa de persistencia de datos -> VOs y DAOs

En esta capa hemos seguido un patrón de VOs y DAOs, de esta forma organizamos el acceso a los datos y representamos cada entidad de la base de datos, respectivamente. Además, se consigue facilitar tanto el mantenimiento de la web como la escalabilidad ya que se separa la lógica de acceso a la base de datos.

Las clases VO que utilizados son ComentarioVO, CompeticionFavVO, CompeticionVO, EquipoFavVO, EquipoRankingVO, EquipoVO, HistoricoVO, JugadorFavVO, JugadorVO, MensajeVO, PartidoVO, PtsJugarParVO y UsuarioVO. Esta es la forma que hemos creído más conveniente para separar los datos de la base de datos de manera que sea mucho más sencillo acceder a ellos, ya que representan las entidades de la base de datos en el código de la web.

Las clases DAO son ComentarioDAO, CompeticionFavDAO, CompeticionDAO, EquipoFavDAO, EquipoDAO, HistoricoDAO, JugadorFavDAO, JugadorDAO, MensajeDAO, PartidoDAO, PtsJugarParDAO, TokenDAO y UsuarioDAO. De esta manera cada clase contiene las operaciones necesarias para acceder a las tablas de datos.

Capa de modelo y capa de interfaz

Se gestiona principalmente mediante Servlets, los cuales reciben las solicitudes a realizar mediante las interacciones del usuario con la web, coordinan con las clases DAO para después redirigir al archivo jsp a mostrar. En resumen, se encargan de gestionar las pantallas y sesiones del usuario y de redirigir a las pantallas correctas según las acciones realizadas.

La interfaz ha sido diseñada principalmente para la visualización de la web en un ordenador, aunque también puede adaptarse para otros dispositivos. Esto se debe a que el archivo css se ha realizado con las proporciones de la pantalla de un ordenador.

Por ejemplo, las clases *thecontainer* y *container* son las básicas y las utilizamos en todas las pantallas diseñadas. Por otra parte están las clases específicas de cada pantalla, como son el caso de *quienes_somos-text* y *cabecera2-section*.

```
.thecontainer {  
    display: flex;  
    flex-direction: column;  
}  
.container { /* Contenedor básico */  
    display: flex;  
    width: auto; /* Asegura que el contenedor ocupe el 100% */  
    height: auto;  
    background-color: white;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
}  
.quienes_somos-text {  
    display: block;  
    font-size: 22px;  
    margin-bottom: 10px;  
    padding-left: 8px;  
}  
.cabecera2-section {  
    background-color: var(--color-primary); /*Color de fondo usando la variable */  
    text-align: left; /* Alineación del texto a la izquierda */  
    z-index: 1000; /* Asegura que esté por encima de otros elementos */  
    width: 100vw; /* Ocupa el 100% del ancho de la pantalla */  
    height: 138px; /* Establece una altura fija para la cabecera */  
    display: flex; /* Usa flexbox para la alineación */  
    align-items: center; /* Centra verticalmente el contenido */  
    justify-content: flex-start;  
}
```

Metodología de trabajo

Herramientas utilizadas

Las herramientas que hemos utilizado en el proyecto han sido Eclipse Java Developers como IDE, Apache Tomcat, PostgreSQL y DBeaver. Además, para gestionar los archivos de una manera eficiente entre todos los integrantes del grupo, hemos empleado un repositorio de GitHub para todos los archivos del proyecto y una carpeta compartida de Google Drive para la creación de las memorias.

De esta forma hemos podido acceder en todo momento a cualquier documento para editar y avanzar en el proyecto y que el resto del grupo tenga esos archivos modificados disponibles.

Organización y distribución del trabajo en equipo

Para la realización del trabajo hemos optado mayormente por trabajar juntos tanto en videollamadas como presencialmente, aunque también hemos adelantado por separado. Además, como utilizamos GitHub nos hemos asegurado de no sobreescribir el trabajo de los demás compañeros.

Para la distribución del trabajo hemos procurado que todos los miembros del grupo participen en cada sección del proyecto de forma que así cada uno comprenda todas las secciones del proyecto.

Horas de trabajo

Tareas a realizar	Horas empleadas (h)
Configurar entorno de trabajo, servidor web y conexión al SGBD	4
Integrar diseño e implementación de la capa de persistencia de datos	5
Diseñar y desarrollar la capa modelo	5
Completar diseño de la capa vista (interfaz)	85
Realizar las pruebas y tests	3
Elaborar memoria	11
Total	113

En el caso de *Realizar las pruebas y tests* se han puesto 2h ya que fueron únicamente de pruebas. Durante la realización de la interfaz se fueron realizando distintas pruebas para verificar lo que se iba diseñando.

Decisiones tomadas

Hemos decidido realizar la mayor parte del sistema en *java* debido a que es el lenguaje de programación presentado en clase. Para las páginas del sistema *html* de la misma manera hemos usado *.jsp* ya que permite introducir código *java* en *html* para que genere contenido dinámico en las páginas.

Asimismo usamos en ciertos puntos *javascript* entre los *.jsp*, para poder hacer el sistema más interactivo con el usuario, como para, por ejemplo, ser capaz de ver la contraseña y para mostrar el contenido dinámico de los buscadores. Finalmente para los buscadores se ha usado *gson* para transferir los datos fácilmente con un formato concreto.

Hemos decidido que si se produce prórroga por empate durante los cuatro cuartos, los puntos anotados en todos los tiempos de la prórroga jugados se sumen a los del último cuarto y aparezcan todos juntos.

Para diferenciar ambos usuarios se optó por diferenciar las interfaces con dos colores diferentes, que son además los colores corporativos de BaskeTracker, el naranja y el azul, el naranja es el color de los usuarios y jugadores que utilicen la web y el azul marino es el color elegido para la interfaz del usuario.

Dificultades encontradas

Hemos encontrado varias dificultades que a lo largo de la realización del proyecto:

- GitHub: el principal problema fue la gestión del código fuente. Esto sucede ya que si dos personas cogen la última versión de un archivo y lo modifican, cuando lo suban una de las versiones se borrará. La versión que se quedará será la última que se haya subido al repositorio, por lo que se pueden generar conflictos entre las versiones por mala sincronización. Para evitar que suceda esto hemos tenido que organizarnos de forma que no modifiquemos simultáneamente ningún archivo.
- HTML: la adaptación a un nuevo lenguaje de programación que no habíamos utilizado hasta el momento. Al principio nos costó entender la estructura del código para que se visualizara correctamente la página, por ello nos ayudamos de los ejemplos de las diapositivas de teoría.
- CSS: hemos tenido bastantes problemas de ubicación y colocación de los elementos mediante el `stylesheet.css` ya que nos costó ubicar cada cosa en el sitio que la queríamos. Tuvimos que realizar muchas pruebas para conseguir encontrar los parámetros adecuados para tener una correcta colocación y que la interfaz fuera funcional y estéticamente bonita a nuestro gusto.
- Servlets: al igual que el HTML, los servlets también han sido algo nuevo que hemos tenido que aprender a implementar. Tras ir creando varios e irlos probando fuimos mejorando y aprendiendo a cómo hacerlos de manera correcta.
- Coordinación entre tecnologías: al tener distintas tecnologías como Java o PostgreSQL, tuvimos algunos problemas de comunicación entre las diferentes capas ya que la comunicación entre ellas no era la esperada. Para resolverlo tuvimos que probar muchas veces hasta conseguir una correcta conexión.

Plan de pruebas de usuario

Dentro del sistema había que verificar distintos aspectos relacionados con la navegabilidad e inserción de datos. Durante la propia implementación se han ido comprobando sus posibles fallos.

Se ha verificado que hay navegabilidad entre todas las pantallas del usuario, es decir puedes acceder en todo momento a la búsqueda de jugadores, equipos o competiciones, a los registros de tus favoritos y los chats de mensajes con tus jugadores preferidos.

Del mismo modo todos los datos se actualizan al momento, por ejemplo si seleccionas a un jugador como favorito te aparece dinámicamente en el apartado de favoritos y si lo eliminas de favoritos a la inversa.

Por otro lado también ha sido preciso verificar la correcta inserción de los datos por el administrador, para esto nos hemos centrado en la prevención de errores.

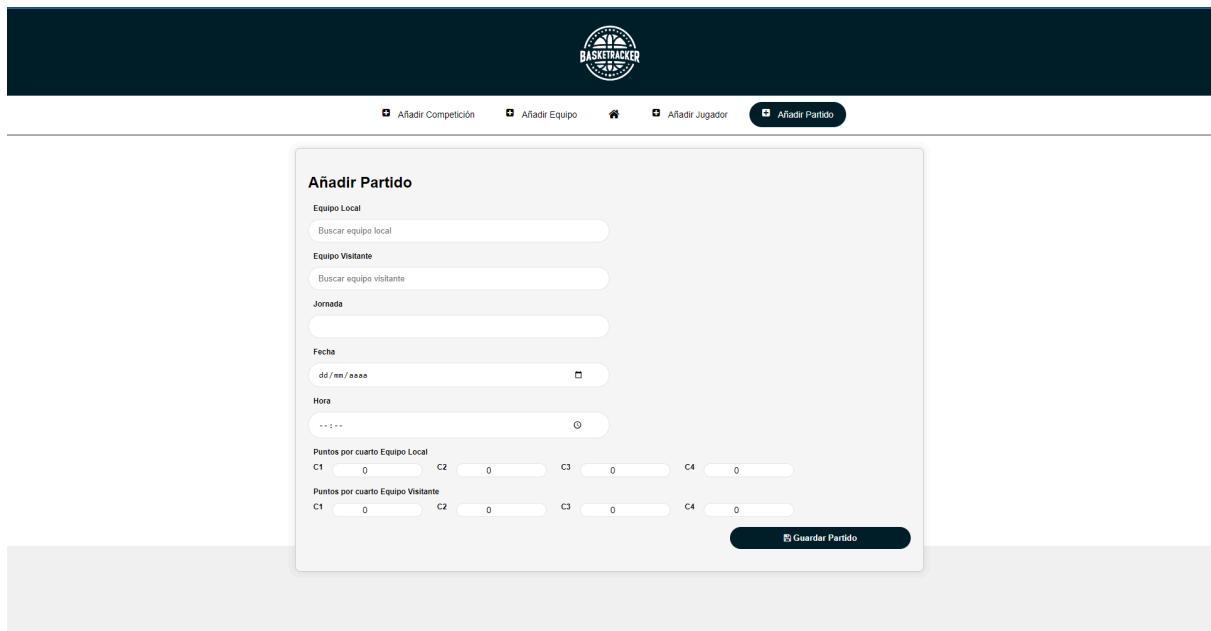
Por ejemplo, para insertar un equipo nos aseguramos de que esté asociado con una competición ya existente y que no haya equipos con el mismo nombre en esa competición. En el caso de los jugadores hay que seleccionar un equipo válido del mismo modo. Y para insertar un partido se verifica que los equipos pertenezcan a la misma competición y sean distintos. Un jugador dentro de un partido concreto no puede anotar más tiros libres de los que lanza y no puede anotar menos puntos que los triples junto con los tiros libres.

Manual del usuario

Inicialmente para poder acceder al sistema es necesario autenticarse mediante un inicio de sesión. Desde esta se permite acceder a un pantalla para registrarte del mismo modo o recuperar la contraseña en caso de que la hubiera olvidado el usuario.

Perfil de Administrador

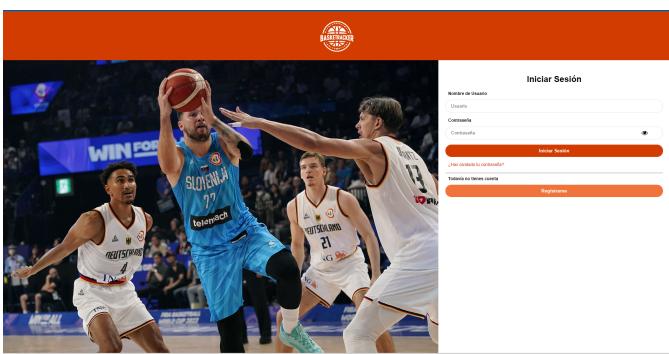
Desde la pantalla de inicio si tu usuario se identifica como un usuario administrador se redirige a la pantalla de inicio de usuario. En ella con una barra de navegación se pueden insertar competiciones, equipos, jugadores y partidos. Asimismo las estadísticas de un jugador en un partido concreto de su equipo.



The screenshot shows a dark-themed web application interface. At the top, there's a navigation bar with links for 'Añadir Competición', 'Añadir Equipo', 'Añadir Jugador', and 'Añadir Partido'. Below the navigation is a section titled 'Añadir Partido' with sub-sections for 'Equipo Local' and 'Equipo Visitante', each with a search input field. There are also fields for 'Jornada', 'Fecha' (date), 'Hora' (time), and two sets of scoring fields for 'Cuarto' (quarter) C1 through C4. A large blue button labeled 'Guardar Partido' (Save Match) is at the bottom right of the form area.

En la mayoría de ellas se puede obtener datos mediante buscadores para poder relacionar los datos más fácilmente. Igualmente se incluyen filtros de verificación de los datos para comprobar que es posible insertarlos, y en caso contrario dar el feedback necesario.

Perfil de Público Objetivo



Cuando el usuario inicia la página web, empieza en la pantalla de inicio de sesión, allí el usuario puede realizar diferentes acciones. Si el usuario no ha entrado nunca en la web, necesita registrarse primero porque seleccionaría el botón de registrarse y le llevaría a esa página, donde tendría que llenar los campos necesarios para registrarse. Si todo es correcto será redirigido a iniciar sesión. En caso de que el usuario no recuerde su

contraseña puede pedir cambiarla. Si el usuario ya se registró previamente, simplemente tendrá que llenar su usuario y contraseña y entrará a la página.

El usuario verá una página de **inicio** donde se muestran unos jugadores, equipos y competiciones destacadas que puede añadir a favoritos. El usuario podrá consultar todos los jugadores, equipos y competiciones favoritas y ver diferentes datos de ellas. Además tiene un buscador donde puede buscar todo lo anterior y acceder más fácil si sabe lo que buscar.

Desde la pantalla de **favoritos** al usuario le saldrán tres filas con todos sus favoritos. Y podrá hacer click en los botones correspondientes para que pueda acceder a la información de lo que deseé consultar.

En la pantalla de **mensajes** el usuario podrá acceder a los canales de difusión de sus jugadores favoritos, pero en caso de que sea jugador el usuario que se ha registrado tendrá acceso a su canal de difusión para poder escribirlo que quiera a sus seguidores.

Las pantallas de **consultar jugadores** y **usuarios** son muy parecidas. En la de jugadores muestra la información del último partido y el historial de ese jugador. Sin embargo, si es usuario se mostrarán los jugadores favoritos para que pueda acceder a sus datos. En los jugadores podrá ver los próximos partidos de ese jugador y en el caso del usuario los próximos partidos de sus jugadores favoritos.

En la página de consulta de los equipos, el usuario podrá acceder a un calendario donde están marcados todos los partidos registrados. Además puede consultar los jugadores que pertenecen a ese equipo.

Máximos anotadores			
	P.J.	PTS	PTS/P.J.
1. Mario López García	2	28	14
2. Rafael Torres Romero	2	24	12
3. Marina González Ruiz	1	22	22
4. Mario Ferradas Aznar	1	20	20
5. Alba Torres Ramírez	1	19	19
6. Jorge Clavero Agudo	1	18	18
7. Carlos Díaz Martínez	1	18	18
8. Isabel Gómez Sánchez	1	18	18
9. Andrea Hernández Artal	1	15	15
10. José Pérez López	1	15	15

Máximos triples			
	P.J.	TRP	TRP:P.J.
1. Martín Rubio Díaz	1	1	1

de los jugadores en diferentes estadísticas: máximos triples anotados, máximos tiros libres anotados y más puntos anotados.

Y para terminar puede ver la información de los partidos, donde puede ver los puntos en cada cuarto, ver los rankings propios de ese partido y donde podrá hacer comentarios para este partido.

	C1	C2	C3	C4
Huesca Stars	18	22	17	21
Tarragona Eagles	15	20	19	18

Mayores Anotadores

Mayores Tripistas

Mayores Anotadores

Comentarios
 @ma_kaa
 El mejor partido de la temporada

Escribir Comentario Enviar

Además de que el usuario cuenta con otras funcionalidades como consultar la página de quiénes somos, una página de ayudas y consultas, puede cambiar el correo asociado a la cuenta o sin embargo eliminar su cuenta, en el caso de los usuarios normales. Si es jugador tendrá que mandar una sugerencia en un formulario para eliminar su cuenta.

La web cuenta con numerosos atajos que el usuario puede utilizar y que ayudan a acceder a mayor cantidad de información en menor tiempo.

En la página correspondiente a la consulta de una competición tiene acceso a todos los partidos de todos las jornadas de esa temporada donde podrá hacer click y consultar la información de esos partidos. Luego el usuario puede acceder a la clasificación de esa temporada para ver cómo van los diferentes equipos. Puede ver todos los equipos pertenecientes a esa competición y consultar el ranking

Plan de mantenimiento

El sistema está diseñado para mostrar información continuamente y con actualizaciones frecuentes, por este mismo motivo es necesario configurar una infraestructura que permita mantener toda la masa de información al día. Del mismo modo se podría llevar a algún acuerdo con la Federación Aragonesa de Baloncesto para intercambiar la información, porque a pesar de que los datos de los resultados de las competiciones son públicos la búsqueda e inserción sería más fluida.

Para ello se ha creado un perfil de administrador que sea capaz de crear competiciones, equipos y jugadores. Así como introducir los resultados entre partidos y las estadísticas de los jugadores. Esto es posible autenticando al usuario mediante una interfaz intuitiva y sencilla.

Actualmente solo se dispone de un usuario con permiso de administrador, no obstante es probable que se puedan crear nuevos perfiles de administrador para poder llevar a cabo la inserción desde distintos puntos. Por ejemplo, los árbitros de mesa que toman nota de los puntos durante el partido podrían tener un rol de administrador que les permitiese introducir los datos de un partido concreto fácilmente.

Pese a todo esto, en la mayoría de competiciones se disputan partidos semanalmente por lo que la cantidad de datos a insertar en la base de datos puede llegar a ser muy insostenible de forma manual. Para este caso se pensó la posibilidad de insertar los datos desde el administrador en formato de ficheros de texto o ficheros .csv en los que almacenen la información de los partidos semanalmente. Esta funcionalidad no ha sido por ahora implementada pero no se descarta llevarla a cabo en una posible futura actualización del sistema.

Además se querría añadir nuevas funcionalidades como permitir registrar en la base de datos a los entrenadores de cada uno de los equipos, ya que son parte importante de la gestión y de los resultados que obtendrá ese partido, esa información no ha sido añadida pero se añadirá en un futuro.

Otra funcionalidad a implementar sería la posibilidad de que los usuarios en los canales de difusión pudieran reaccionar a los mensajes mandados por sus jugadores favoritos con diferentes emoticonos, ya que esta primera versión del canal de difusión es bastante sencilla que solo permite la interacción de jugadores a usuarios, de la misma forma que se implementará la posibilidad de responder a comentarios en los partidos, para poder generar conversación y fomentar que la gente habla de los partidos en nuestra web.

Para una siguiente actualización se incluiría la posibilidad de añadir más cuartos a la hora de registrar el resultado de un partido. De esta forma cuando se produzca una prórroga, el administrador podrá ir añadiendo los períodos adicionales necesarios.

Seguridad del sistema

Para que nuestra aplicación fuera segura, una de las medidas tomadas ha sido encriptar las contraseñas a la hora de registrar a los usuarios. Es decir, cuando un usuario se registra en nuestra web, en la base de datos no guardamos su contraseña tal cual porque es invasión de la privacidad y cualquiera que accediese a la base de datos podría conseguir todas las contraseñas. Para ello todas las operaciones con las contraseñas son encriptadas.

Para poder realizarlo hemos utilizado el jBCrypt-0.4.jar que es una biblioteca de Java para proteger las contraseñas. JBCrypt utiliza un algoritmo llamado Blowfish, adaptado para hashing seguro de contraseñas. Realiza un hashing adaptativo y convierte la contraseña en un hash, en una serie de caracteres irreversibles que son únicos y son muy difícil de revertir.

Accesibilidad y usabilidad del sistema

Para hacer el sistema accesible y usable por un público más amplio, se tomaron en cuenta varios principios de diseño accesible. Los colores fueron elegidos para asegurar un contraste suficiente, lo que permite que el texto sea fácilmente legible.

También se incluyeron textos alternativos en las imágenes, facilitando el acceso a las personas que usan lectores de pantalla.

La estructura de la información se organizó de manera que fuera clara y comprensible, mejorando la experiencia de los usuarios y ayudando a quienes tienen dificultades cognitivas a encontrar información de manera eficiente.

Además, se añadió la posibilidad de navegación por teclado en los formularios, permitiendo a los usuarios recorrer los campos mediante el uso del tabulador, lo que beneficia a quienes prefieren o necesitan interactuar sin usar el ratón.