

Proyecto Integrado:

"Aplicación de Deportes"

Desarrollo de Aplicaciones Web
Ciclo Formativo Grado Superior
IES Rafael Alberti



Manuel Puerto Gutiérrez manuelpuertogutierrez@gmail.com Abril-Junio de 2020.

ÍNDICE DE CONTENIDOS

1. Presentación del proyecto	3
1.1 Introducción.....	3
1.2 Tecnologías.....	4
2. Descripción	6
3. Instalación y preparación	9
3.1. Opción convencional	9
3.2 Utilizando la herramienta Docker	9
4. Diseño de la base de datos.....	10
4.1 Definición teórica de los requisitos	10
4.2 Modelo Entidad Relación	11
4.3 Modelo Relacional	11
5. Desarrollo	13
6. Pruebas	18
7. Despliegue.....	19
8. Manual de usuario	21
8.1 Iniciación de la aplicación en Docker	21
8.2 Uso de la aplicación	21
9. Conclusiones.....	23
10. Índice de tablas e imágenes.....	24
11. Bibliografía y referencias.....	25

1. Presentación del proyecto

1.1 Introducción

Este proyecto consiste en la creación de una aplicación web de deportes, la cual obtiene los datos desde una API para crear las competiciones, los jugadores, los equipos y los partidos los cuales se almacenan en una base de datos MYSQL para que el usuario pueda procesarlos con mayor rapidez en la interfaz gráfica.

Los usuarios que estén registrados en la aplicación como colaboradores además de visualizar los datos podrán modificar la descripción de los atletas.

El objetivo de esta aplicación es el visionado de datos de los profesionales del fútbol de las principales ligas, aunque debido a la arquitectura de la aplicación es fácilmente extensible a otros deportes, la aplicación solo admite las principales ligas porque hemos seleccionado la versión gratuita de la API; pero si algún equipo se mostrara interesado podría adquirir la versión de pago que contiene ligas de divisiones menores y utilizar la aplicación como una herramienta de “scouting” para ver que equipos destacan y poder comentar las características de sus jugadores y poder tener un seguimiento de estos.

La aplicación permite a través de la obtención de datos de una API crear competiciones de forma automática y asignar a cada competición la clasificación actual, también dispone de una gráfica que permite visualizar los principales datos de los equipos de la competición de forma mucho más sencilla, además es posible ver los jugadores que forman parte de cada conjunto, así como su descripción y si el usuario esta registrado podrá añadir o modificar las descripciones para que el resto de usuarios puedan disponer de esta información. La aplicación contiene en el inicio una sección de noticias que se actualizan de forma automática a través del uso de una API.

Aunque hay varias APIS de fútbol y también periódicos deportivos, mi aplicación se encontraría a medio camino entre ambas, permitiendo obtener información de forma rápida y eficaz, pero disponiendo de una interfaz gráfica que sin embargo no se encuentra sobrecargada de texto y opiniones como lo hacen muchos periódicos

deportivos, lo que permitiría un acceso más rápido a los datos relevantes por parte del usuario. Además, si una empresa o equipo profesional estuviera interesada, con la API de pago se podría la aplicación se puede utilizar para tener un seguimiento de los jugadores que no son tan conocidos, compartiendo en la descripción los informes de los jugadores.

1.2 Tecnologías

Para la realización del proyecto de han utilizado las siguientes tecnologías.

- **Java:** Lenguaje de programación orientado a objetos, de código abierto utilizado para todo tipo de aplicaciones desarrollado por Sun Microsystem y adquirido por Oracle.
- **MySQL:** es un sistema de gestión de bases de datos relacional desarrollado por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo.
- **JavaScript:** es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web.
- **jQuery:** es una biblioteca de JavaScript rápida, pequeña y rica en funciones. Hace que cosas como el desplazamiento y la manipulación de documentos HTML, el manejo de eventos, la animación y Ajax sean mucho más simples con una API fácil de usar que funciona en una multitud de navegadores.
- **Bootstrap:** es un framework CSS y Javascript diseñado para la creación de interfaces limpias y con un diseño responsive. Además, ofrece un amplio abanico de herramientas y funciones, de manera que los usuarios pueden crear prácticamente cualquier tipo de sitio web haciendo uso de estas ayudas.
- **Docker:** es una tecnología en que posibilita la creación y el uso de

contenedores Linux. La virtualización basada en contenedores aísla las aplicaciones entre sí en un sistema operativo (OS) compartido.

- **HTML 5:** Lenguaje de Marcas de Hipertexto, se utiliza para el desarrollo de páginas de Internet.
- **CSS3:** Hojas de estilo en cascada, es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Se utiliza para añadir estilo a documentos estructurados. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML.
- **Google Charts:** es una API que nos permite, mediante una petición http a esta, generar una imagen dinámica de tipo PNG que podemos colocar en cualquiera de nuestras páginas web.
- **Tomcat:** es una implementación de código abierto de las tecnologías Java Servlet, JavaServer Pages, Java Expression Language y Java WebSocket implementando sus especificaciones y haciendo posible que estos funcionen.
- **Balsamiq Mockups 3:** herramienta de diseño gráfico que se utiliza para la creación de bocetos.
- **Dia:** aplicación informática de propósito general para la creación de diagramas.

2. Descripción

El resultado ha sido el que se pretendía, una interfaz de uso sencillo que permite el visionado de datos de forma rápida y la posibilidad de ver ciertos datos de forma gráfica lo que supone una mejor experiencia por parte del usuario.

Además, los usuarios registrados disponen de la posibilidad de compartir modificar la aplicación, añadiendo información lo que puede incentivar a los usuarios a usar la aplicación web y a que esta disponga de más datos gracias a la aportación de la comunidad o compartir datos privados si se trata de una empresa del sector.

Al ser una aplicación Web, todas las funciones las presta a través de páginas Web a las que se acceden por medio de un navegador. Se ha optado por un diseño sencillo y minimalista ya que el principal es la consulta de datos. Todas las páginas se adaptan al tamaño de la pantalla de los dispositivos gracias a la tecnología Bootstrap.

En todas las páginas de la aplicación excepto en el registro y login; disponemos de un header y una barra de navegación, que nos permite interactuar con la misma y dirigirnos a las secciones seleccionas de forma sencilla y un footer con la información de la página.

- **Página de Inicio:** dispone de una slider con imágenes de los equipos campeones. Debajo se muestran noticias deportivas recientes obtenidas a partir de la API de noticias <http://newsapi.org>.
- **Página de Competiciones:** contiene un “select” que permite filtrar las competiciones por país. Debajo del mismo se muestran las competiciones seleccionadas, en cada una de las cuales se puede Seleccionar los equipos, la clasificación y un gráfico de con que muestra los datos más relevantes de la misma.
- **Clasificación:** muestra la clasificación de la competición seleccionada y los datos más importantes de cada equipo utilizando una tabla que se adapta

a las proporcionas de cada pantalla gracias a la clase “table” de Bootstrap.

- **Gráfico:** muestra de forma gráfica los datos de los equipos de la competición seleccionada ordenados por su posición en la competición. La página muestra un gráfico de barras que es creado gracias a la ayuda de la tecnología Google Chart.
- **Mostrar Equipos:** muestra los equipos que participan en la competición seleccionada y cada equipo contiene un enlace que permite ver a la información de todos los jugadores que participan en el mismo.
- **Jugadores Combinado:** muestra todos los jugadores del equipo seleccionado, cada jugador tiene un enlace a editar jugador.
- **Editar Jugador:** esta página solo se encuentra disponible para los usuarios que están registrados y han iniciado sesión, muestra la información del jugador seleccionado además de un formulario para editar la descripción del jugador.
- **Jugadores:** Muestra todos los jugadores, aunque estos se pueden filtrar por país y por posición gracias a dos selects en la parte inferior. Cada jugador se muestra en una Card de Bootstrap el nombre del jugador, la descripción, si esta ha sido introducida por algún usuario colaborador o por el administrador, (en caso contrario aparecerá la descripción vacía), el país de origen, la fecha de nacimiento y un botón editar que es un enlace a la página editar jugador del jugador seleccionado.
- **Registro e Iniciar sesión.** Las páginas registro e iniciar sesión contienen un diseño diferente al resto, para dar una mayor variedad a la aplicación. Ambas contienen un enlace al inicio de la aplicación por si el usuario quiere seguir como invitado, y dos inputs uno con el usuario y otro con la contraseña.

En el caso del Registro estos datos una vez hayan pasado las restricciones de tamaño (han de tener al menos cinco caracteres y que el Nick no se encuentre seleccionado por ningún usuario) se introducirán en la base de datos.

En el caso de Iniciar Sesión si los datos introducidos en los inputs contienen al menos cinco caracteres, la aplicación comprobará que existe un usuario

registrado con ese nick y esa contraseña.

Tanto en el registro como en el inicio de sesión si los datos son correctos se enviará al usuario al inicio, pero habiendo inicializado las variables de sesión lo que permitirá a los usuarios realizar cambios en la aplicación; en caso de que los datos introducidos fueran erróneos se mostrará un mensaje de error en el HTML.

- **Crear.** Consiste en un select mediante el cual el usuario selecciona una de las ligas disponibles y al pulsar el botón Crear/Actualizar competición la competición seleccionada se crea si no existe y en el caso de que exista se actualiza introduciendo los datos que aún no existan.

Los datos se hacen persistentes porque al hacer clic en el botón indicado se añaden las competiciones, equipos y jugadores que no existían con anterioridad, en la base de datos.

El funcionamiento interno de la aplicación se explicará con más detalle en el apartado de desarrollo.

3. Instalación y preparación

En cuanto los preparativos necesarios para poder poner a funcionar nuestro proyecto o ejecutarlo, existen dos opciones.

3.1. Opción convencional.

Para poder ejecutar y modificar nuestro proyecto necesitamos un entorno de ejecución de java compatible con el que se ha creado el proyecto.

```
C:\Users\puerto>java -version
java version "1.8.0_231"
Java(TM) SE Runtime Environment (build 1.8.0_231-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.231-b11, mixed mode)
```

Imagen 1: Entorno de Ejecución de Java.

Debemos tener una IDE compatible con Java en mi caso he usado Eclipse IDE. Y un gestor de bases de datos, la elección para mi proyecto ha sido la de WAMP.

Una vez disponemos de las herramientas adecuadas simplemente debemos exportar el .war en el que se encuentra el proyecto (o crearlo como Dynamic Web Project en mi caso) y ejecutarlo y ejecutarlo en el servidor.

3.2 Utilizando la herramienta Docker.

También podemos usar Docker: para ello necesitamos tener instalado Docker en nuestro ordenador y a través de la CMD dentro de la carpeta dockerización utilizar el comando docker-compose up para crear los contenedores una vez hecho esto debemos entrar en la dirección http://localhost:8080/aplicacion_deportes_manuel/index.html, esto se explicará con más detalle en el número 7.

4. Diseño de la base de datos

El diseño de la base de dato fue realizado en tres partes la primera parte consistió en una definición teórica de los requerimientos, después a partir de esta definición, pudimos crear el modelo Entidad Relación y a partir de este pudimos realizar el Modelo Relacional.

4.1 Definición teórica de los requisitos.

Una vez clarificados los requisitos de la aplicación, nos encontramos con las siguientes premisas o condicionantes los cuales ha de cumplir nuestra base de datos:

- Un deporte puede tener una o varias competencias y una competición solo puede ser de un deporte.
- En una competición participan al menos dos combinados y un combinado puede participar en de 1 a N competencias, cada equipo tiene unos puntos, unas anotaciones a favor y unas anotaciones en contra para todas las competencias en las que participa.
- Una competición consta de uno o varios partidos y un partido solo puede pertenecer a una competición.
- Un combinado (local) juega uno o varios partidos contra otro combinado(visitante);(esta relación es ternaria y reflexiva al mismo tiempo).
- Los combinados contratan a varios jugadores y un jugador puede estar contratado en varios equipos durante un tiempo determinado (equipo y selección, por ejemplo).
- Un deporte es practicado por varios jugadores y un jugador solo puede participar en un deporte.

4.2 Modelo Entidad Relación

Una vez obtenidas las restricciones del problema podemos obtener el modelo entidad relación el cual se representa en la siguiente imagen.

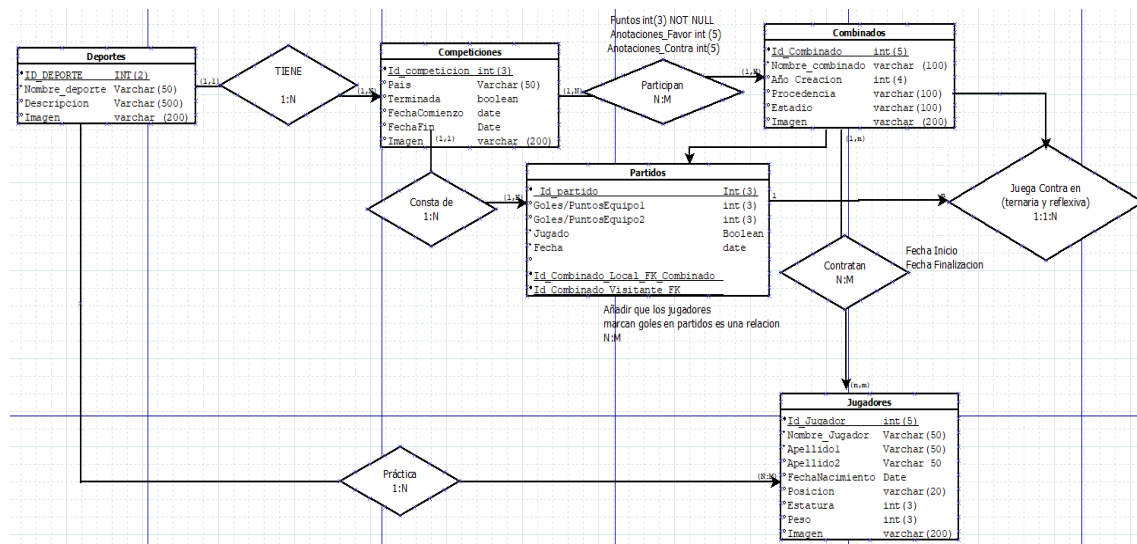


Imagen 2: Modelo Entidad Relación.

4.3 Modelo Relacional

A partir del Modelo Entidad Relación anterior hemos realizado el proceso de normalización de la base de datos, a partir del cual hemos obtenido el Modelo Relacional de la misma el cual consta de las siguientes tablas.

- **Deportes** (Id_Deporte INT(2) (PK), Nombre_Deporte varchar(50), Descripción varchar(500), Imagen varchar (200))
- **Competiciones** (Id_Competicion INT(3)(PK), Nombre varchar(100), Pais Varchar (50), Terminada Boolean, Fecha_ Comienzo date , Fecha_Fin date, Imagen varchar (200)
- **Combinados** (Id_Combinados Int(5) (PK), Nombre_combinado varchar(100), año_creacion int(4), procedencia varchar(100), estadio varchar(100), Imagen varchar (200))
- **Combinados_Participan_en_competiciones** (Id_Combinados int(5) (PK

combinados), Id_Competicion int(3) PK competición), Puntos int(3) not null, Anotaciones_Favor int(5), Anotaciones_Contra int(5).

- **Partidos** (ID_Partido int(3) (PK), Goles/Puntos_Combinado1 int(3), Goles/Puntos_Combinado2 int(3), Jugado boolean, Fecha date, Id_Competicion int (3)(FK_Competiciones), Id_Combinado1 int(5)(FK_combinados) , Id_Combinado2 int(5) (FK_combinados)).

- **Jugadores**(ID_Jugador(PK) int(6) , Nombre_Jugador varchar(50), Apellido1 varchar(50), Apellido2 varchar(50) , Descripción Varchar(500), Fecha_Nacimiento date, Imagen varchar (200), Posicion varchar(50) , ID_Deporte int(2) (fk_deportes).

- **Combinados_Contratan_Jugadores** (Id_Combinado Int(5) (PK_Combinados), Id_Jugador int(6) (PK_Jugadores), Fecha_Inicio_Contrato date, Fecha_Fin_Contrato date).

5. Desarrollo

El proceso de elaboración del proyecto se realizó de la siguiente forma.

El primer paso fue definir las funcionalidades que debía tener la aplicación y las páginas de iba a contener la misma, de forma aproximada.

De forma posterior se realizó la decisión de las tecnologías a utilizar.

Una vez ya disponía de una aproximación teórica de lo que se quería conseguir, realicé el diseño de la base de datos; primero definiendo las restricciones necesarias para conseguir toda la información disponible, después realizando el modelo entidad relación, una vez conseguido se pasó al modelo relacional y se realizaron la creación de la base de datos y de las tablas correspondientes.

Una vez obtenida la base de datos pasé a realizar el diseño de la parte visual utilizando la aplicación Balsamiq Mockups 3.

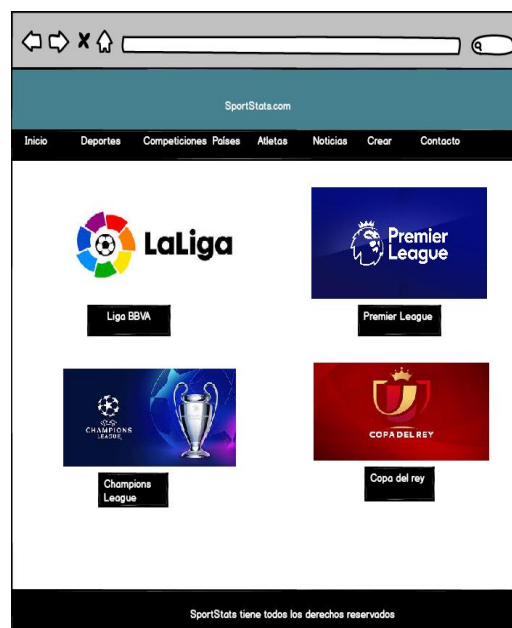


Imagen 3: Diseño del Inicio.

Al estar completado el diseño de la parte visual se implementó usando Html5, Bootstrap 4 y Css3.

Una vez definido el proyecto y comenzado la parte visual, se procedió a buscar

alguna API gratuita de futbol de la que obtener los datos y decidí utilizar la API “api.football-data.org”; la cual me permite obtener información sobre diferentes competiciones de fútbol, así como las clasificaciones, los equipos y los jugadores que pertenecen a los mismos de forma gratuita.

(Sub)Resource	Action	URI	Filters	Sample
Competition	List all available competitions.	/v2/competitions/	areas={AREAS} plan={PLAN}	Open
Competition	List one particular competition.	/v2/competitions/2000		Open
Team	List all teams for a particular competition.	/v2/competitions/{id}/teams	season={YEAR} stage={STAGE}	Open
Standings	Show Standings for a particular competition.	/v2/competitions/{id}/standings	standingType={standingType}	Open
Match	List all matches for a particular competition.	/v2/competitions/{id}/matches	dateFrom={DATE} dateTo={DATE} stage={STAGE} status={STATUS} matchday={MATCHDAY} group={GROUP} season={YEAR}	Open
Scorers	List goal scorers for a particular competition.	/v2/competitions/{id}/scorers	limit={LIMIT}	Open

Imagen 4: Ejemplos de peticiones a la API api.football-data.org.

Una vez seleccionada la API que iba a utilizar se pasó al diseño de la arquitectura, para la cual sirvió de base un proyecto que realizamos en clase sobre una aplicación meteorológica.

La arquitectura se basa en una petición AJAX realizada mediante JQUERY a la API de futbol esta devuelve datos que son procesados por JQUERY los cuales se envían a un Servlet a través de otra petición AJAX a partir de los cuales se crean los objetos que serán renviados como respuesta a la petición AJAX y serán procesados de nuevo utilizando JQUERY y mostrados por pantalla.

Este procedimiento puede variar ya que a veces se obtienen los datos de la base de datos en lugar de la API o los datos se procesan de forma directa sin tener que transformarlos en objetos.

La mayor dificultad del proyecto radica en la inserción en la base de datos, ya que esta se ha de realizar de forma automática cuando el usuario en la página de crear seleccione una de las opciones disponibles y haga clic en el botón. Esto funciona de la siguiente manera:

Si se selecciona uno de los campeonatos que están disponibles a través de JQuery se recoge el id de la competición seleccionada (la cual hemos asignado como un valor del option en el archivo html).

The image shows a web interface for Sportstat.com. At the top, there is a teal header with the site name and a trophy icon. Below this is a dark navigation bar with links: Inicio, Competiciones, Jugadores, Crear, Iniciar Sesión, Registro, Cerrar Sesión, and nuevousuario. The main content area has a light orange background. It starts with the heading 'Competiciones Posibles' followed by a dropdown menu currently showing 'Liga Holandesa'. Below the dropdown is a blue button with the text 'Crear/Actualizar Competición'.

Imagen 5: Selección de la liga a crear.

Una vez obtenido ese Id; llamamos a la función “crear_competicion” y se pasamos el Id que hemos como argumento; esta función realiza una petición a la API para obtener los datos de la competición los cuales introduciremos en nuestra base de datos, una vez realizado este proceso se llama a la función “insercion_equipos” con el id de la competición como argumento.

```

function crear_competicion(id_competicion){
    var respuesta="";
    var equipos="";
    var url='http://api.football-data.org/v2/competitions/'+id_competicion.toString();
    $.ajax({
        headers: {'X-Auth-Token':'8da834ac6d9e42df939ea632eadd8fe9'},
        url:url,
        dataType: 'json',
        type: 'GET',
    }).done(function(response) {
        respuesta=response;

        var nombre=respuesta.name;
        var pais=respuesta.area.name;
        var terminada=false;
        if(respuesta.winner != undefined){
            terminada = true;
        }

        var Fecha_comienzo=respuesta.currentSeason.startDate;
        var Fecha_fin=respuesta.currentSeason.endDate;
        var data={
            'ID': respuesta.id,
            'Nombre': respuesta.name,
            'Pais': respuesta.area.name,
            'Terminada': terminada,
            'Fecha_comienzo':respuesta.currentSeason.startDate,
            'Fecha_fin':respuesta.currentSeason.endDate,
        };

        $.ajax ({
            url: 'Introducir_Competiciones',
            type: 'POST',
            data:data,
            dataType: 'text',
            success: function(data) {
                insercion_equipos(id_competicion);
            },
            error: function() {
                alert("Error en la comunicacion");
            }
        });
    });
}

```

Imagen 6: Función con peticiones Ajax.

Esta función obtiene los datos de los equipos de la competición seleccionada y los introduce en la base de datos; además por cada equipo llama a las funciones “combinados_participan_competiciones”; a la que se le pasa como argumentos el id del equipo y el id de la competición y los introduce en la base de datos para que podamos obtener cuando realicemos una query todos los equipos que participan en una competición: además por cada equipo del bucle se llama a la función

“inserccion_jugadores” y se le pasa como argumento el id del equipo.

En la función `inserccion_jugadores` se introducen los jugadores en la base de datos y por cada jugador se llama `combinados_contratan_jugadores` en la que le pasamos como argumentos el id del equipo y el id de jugador esta función introduce esos datos en la base de datos para que podamos obtener en una query todos los jugadores que pertenecen a un equipo.

En la función `introducir Equipos` se llama una sola vez a la función “`combinados_contratan_jugadores`” y se le pasa como argumento el id de la competición esta función obtiene el número de jornadas de una competición y llama a la función “`partidos_jornada`” pasando el id de la jornada como argumento, esta función obtiene todos los partidos de la jornada y los introduce en la base de datos;(aunque los datos de los partidos no se utilizan en la aplicación de la forma actual pueden ser útiles si se quieren introducir nuevas funcionalidades).

6. Pruebas

Las pruebas se han realizado de forma manual, comprobando todos los casos de uso; el mayor número de errores se encontraba en que a veces la API no enviaba una respuesta a algunas peticiones puesto que no dispone de esos datos y esto debía ser controlado.

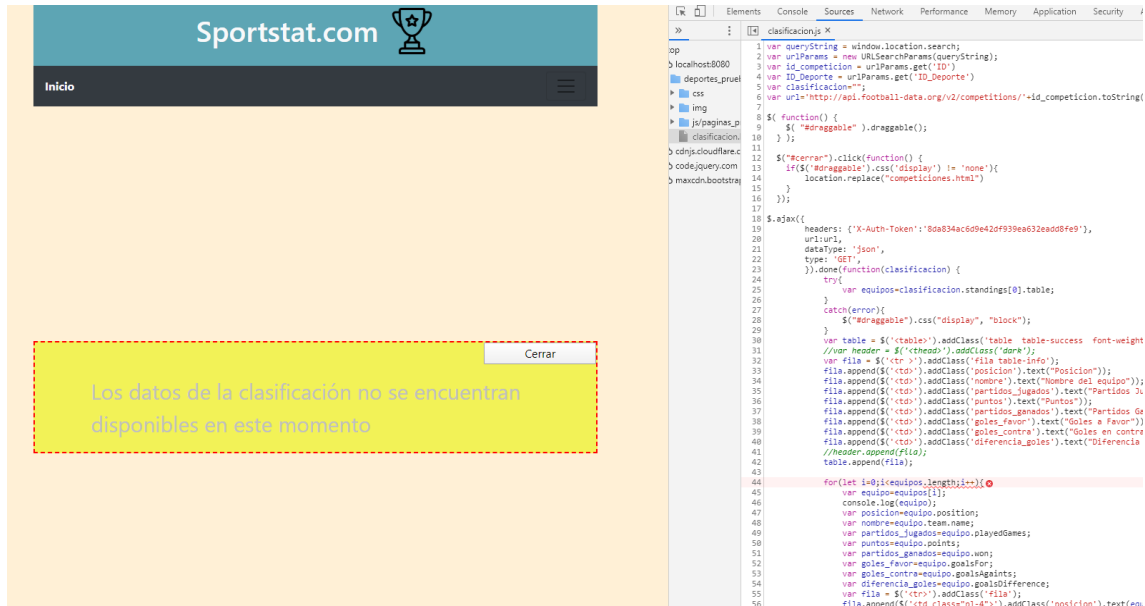


Imagen 7: Comprobación de errores.

La mayoría de los errores en las respuestas se solucionaron comprobando si las variables existían comprobando su longitud y gracias a los estamentos try y catch.

Además, en las peticiones a la API también se ha comprobado la conexión con la base de datos y las peticiones a la misma.

7. Despliegue

El despliegue se ha realizado con Dockers. Para ello hemos creado un archivo Docker compose que realiza lo siguiente:

Primero llama al Dockerfile que se encuentra en la misma carpeta y este crea una imagen de Tomcat en la cual se copia nuestro archivo de la aplicación el cual se encuentra en formato .war dentro de la carpeta webapps de la imagen Tomcat que vamos a crear.

```
1 FROM tomcat:8.0.41-jre8
2 add aplicacion_deportes_manuel.war /usr/local/tomcat/webapps/
3 EXPOSE 9002
4 CMD ["catalina.sh", "run"]
```

Imagen 8: Dockerfile Tomcat

Una vez creado el servidor Tomcat vamos a crear la base de datos a la que se va a unir este servidor, la cual se va a tratar de una imagen MySQL en la cual vamos a crear nuestra base de datos llamada “proyecto_deportes” esta imagen es un Dockerfile que se encuentra en el directorio “sql”.

Esta imagen se crea de la siguiente forma:

```
1 From mysql
2 ENV MYSQL_DATABASE proyecto_deportes
3 ENV MYSQL_ROOT_PASSWORD=pestitillo
4 COPY *.sql /docker-entrypoint-initdb.d/
```

Imagen 9: Dockerfile MySQL

Una vez creada esta imagen vamos a crear una imagen de phpMyAdmin la cual va a estar conectada a la imagen de MySQL para poder efectuar cambios de una manera más visual en esta.

Las imágenes están conectadas entre sí en la misma red. Para orquestar los contenedores hemos utilizado un archivo Docker compose.

```
1  version: '3'
2  ∨ services:
3  ∨    web:
4      build: .
5  ∨    ports:
6      - "8080:8080"
7  ∨    links:
8      - "db:proyecto_deportes"
9  ∨    db:
10     build: ./sql
11
12     ports:
13     - "3306:3306"
14
15     PhpMyAdmin:
16     image: nazarpc/phpmyadmin
17     ports:
18     - "8090:80"
19     links:
20     - db:mysql
```

Imagen 10: Docker Compose

Si hubiera algún tipo de error o dificultad en el despliegue se recomienda consultar el manual de usuario.

8. Manual de usuario

8.1 Iniciación de la aplicación en Docker

- 1 - Comprobar que los puertos 8080, 9002, 8090 y 3306 se encuentran cerrados, en caso contrario cerrar los procesos que están abiertos en esos puertos.
- 2 - Utilizando la consola de comandos entrar en la carpeta “dockerización” y ejecutar el comando “docker-compose up”.
- 3- Esperar a que se inician las imágenes (en el caso de que haya algún error reiniciar docker y ejecutar de nuevo el comando “docker-compose up”.

```
C:\Users\puerto\Desktop\dockerizacion>docker-compose up
Pulling PhpMyAdmin (nazarpc/phpmyadmin):...
ERROR: Get https://registry-1.docker.io/v2/: dial tcp: lookup registry-1.docker.io on [::1]:53: read udp [::1]:59725->[::1]:53: read: connection refused
C:\Users\puerto\Desktop\dockerizacion>docker-compose up
```

Imagen 11: Posible error de Docker

8.2 Uso de la aplicación

El primer paso es ver que los contenedores se han creado y están funcionando de forma correcta; para ello ejecutamos el comando docker ps o docker container ps .

```
C:\WINDOWS\system32>docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
2b31fb42dbb3       dockerizacion_web  "catalina.sh run"   4 hours ago        Up 3 hours         0.0.0.0:8080->8080/tcp, 9002/tcp   dockerizacion_web_1
345232478b9a       nazarpc/phpmyadmin "/home/entrypoint.sh" 4 hours ago        Up 4 hours         0.0.0.0:8090->80/tcp               dockerizacion_PhpMyAdmin_1
f25dd2f54abe       dockerizacion_db   "docker-entrypoint.s" 4 hours ago        Up 4 hours         0.0.0.0:3306->3306/tcp, 33060/tcp   dockerizacion_db_1
```

Imagen 12: Contenedores activos y puertos que utilizan

Si los tres contenedores se encuentran funcionando entramos en la dirección:

http://localhost:8080/aplicacion_deportes_manuel/index.html para ver nuestra aplicación.

Si deseamos efectuar cambios en la aplicación debemos pulsar registro en la barra de navegación de nuestra página web.

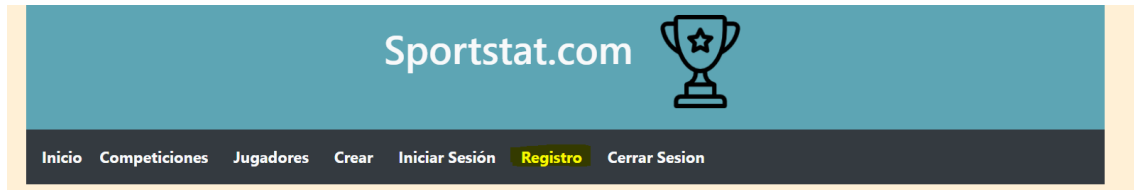


Imagen 13: Barra de navegación

Para completar el registro hay que introducir un correo y una contraseña de al menos 6 caracteres y ya se podrá utilizar la aplicación con todas sus funcionalidades; incluyendo la modificación de la descripción de los jugadores y la creación de nuevas competiciones.

The image shows the registration form page of the application. The background is a solid blue color. At the top, the title 'Formulario de Registro' is centered in white. Below the title, a paragraph in white text reads: 'Regístrate como colaborador de forma gratuita para poder introducir cambios en la aplicación y ayudarnos a mejorar. Si ya estás registrado simplemente [inicia sesión](#).' To the left of the form is a decorative image of a silver trophy on a blue base. The form itself is on the right and contains two input fields: 'Nick' with the placeholder 'Introduce tu Nick' and 'Contraseña' with the placeholder 'Contraseña'. Below the 'Nick' field, there is a small line of text: 'Este será tu nick nombre de usuario en la aplicación'. At the bottom of the form are two green buttons: 'Enviar' and 'Login'.

Imagen 14: Registro

9. Conclusiones

Durante todo el ciclo de vida de este proyecto se ha creado una aplicación web adaptable al tamaño de pantalla de diferentes dispositivos, capaz de mostrar noticias, datos y gráficos sobre las principales ligas de fútbol y en el que los usuarios registrados pueden contribuir a la mejora de la aplicación.

La aplicación utiliza Java 8 y se comunica a través de jQuery, con la API football-data.org de la que procesa datos y los muestra o bien los introduce en una base de datos MYSQL para que puedan ser usados con posterioridad.

La aplicación muestra los datos de las competiciones, los equipos de cada competición seleccionada y los jugadores de cada equipo seleccionado de una forma muy sencilla e intuitiva facilitando la navegación por parte de los usuarios.

Además, dispone de funcionalidades especiales como un filtrado especial en el que se pueden comprobar todos los jugadores disponibles en la aplicación; y estos se pueden filtrar por posición y o por nacionalidad.

La aplicación también da la posibilidad de consultar datos de forma visual con diferentes alternativas ya que dispone de un gráfico de barras creado utilizando la tecnología Google Charts que permite visualizar los datos de cada equipo de la competición seleccionada, mostrando todos los equipos junto a sus datos.

La arquitectura de la aplicación se ha creado para que con cambios mínimos sea escalable y se puedan añadir deportes diferentes.

El despliegue se ha realizado a través del uso de Docker.

10. Índice de tablas e imágenes

IMAGEN 1: ENTORNO DE EJECUCIÓN DE JAVA.	9
IMAGEN 2: MODELO ENTIDAD RELACIÓN.	11
IMAGEN 3: DISEÑO DEL INICIO.	13
IMAGEN 4: EJEMPLOS DE PETICIONES A LA API API.FOOTBALL-DATA.ORG	14
IMAGEN 5: SELECCIÓN DE LA LIGA A CREAR.	15
IMAGEN 6: FUNCIÓN CON PETICIONES AJAX.	16
IMAGEN 7: COMPROBACIÓN DE ERRORES.	18
IMAGEN 8: DOCKERFILE TOMCAT	19
IMAGEN 9: DOKERFILE MYSQL	19
IMAGEN 10: DOCKER COMPOSE.	20
IMAGEN 11: POSIBLE ERROR DE DOCKER.	21
IMAGEN 12: CONTENEDORES ACTIVOS Y PUERTOS QUE UTILIZAN	21
IMAGEN 13: BARRA DE NAVEGACIÓN	22
IMAGEN 14: REGISTRO	22

11. Bibliografía y referencias

AJAX: Asynchronous JavaScript and XML

(https://www.w3schools.com/js/js_ajax_intro.asp).

CSS: Cascading Style Sheets (<http://www.w3.org/TR/CSS2/>).

JSON: JavaScript Object Notation (<http://www.json.org/>).

Modelo Relacional:

(<https://www.ilerna.es/blog/aprende-con-ilerna-online/informatica/modelo-entidad-relacion-base-de-datos>) .

Modelo Entidad Relación:

([https://es.wikipedia.org/wiki/Modelo entidad-relación](https://es.wikipedia.org/wiki/Modelo_entidad-relaci3n)).

MySQL unir tablas:

(<https://stackoverflow.com/questions/3709560/joining-three-tables-using-mysql>).

Docker:

(<https://www.redhat.com/es/topics/containers/what-is-docker>).

Tomcat: (<http://tomcat.apache.org/>).

jQuery:(<https://api.jquery.com/>).