

# **MANUAL DE INSTALACIÓN DE LA VERSIÓN PRELIMINAR**

## **GRUPO 27( ALEXANDRE CURRÁS – DANIEL DUQUE)**

### **ÍNDICE**

#### **1.Instalación**

#### **2. Ficheros de la versión preliminar**

##### **2.1 Interfaz de usuario**

##### **2.2 Ficheros java**

## **1. Instalación**

Para esta aplicación se recomienda el uso del siguiente software y/o tecnologías:

- IDE Eclipse Java
- Plugin para Eclipse “Spring Tools”
- XAMPP
- Spring
- Thymeleaf
- Hibernate
- Gestor de base de datos MySQL

Para su instalación y ejecución se deben seguir los siguientes pasos:

1. Abrimos el IDE e importamos la carpeta del código fuente. Debe importarse como un proyecto Maven ya existente.
2. Crear una base de datos vacía llamada “padelbd”.
3. Ir a /padel/src/main/resources/application.properties y descomentar la línea comentada (Hibernate creará automáticamente la BD). Si no tienes bien configurado el IDE puede dar problemas la creación de la BD, por lo que se acompaña el fichero “padelbd.sql” que contiene un script para la creación de la base de datos con todas las tablas, pero es preferible utilizar Hibernate.
4. Importar los datos de prueba contenidos en el fichero “datosPrueba.sql” en la base de datos
5. Ejecutar la aplicación en el IDE, preferiblemente utilizando Spring Dash Board. En caso de no disponer de él, puede ejecutarse con Run As Java Application.
6. Abrir un navegador y acceder al enlace localhost:8080
7. Para acceder como administrador el DNI es “12345678A” y Pass “admin”

Nota: Es importante tener bien configurados los puertos para que coincidan con los ficheros de configuración.

Nota 2: Si se vuelve a ejecutar la aplicación, se recomienda comentar la línea descomentada anteriormente en application.properties para no volver a crear de cero la base de datos.

## **2. Ficheros de la versión preliminar**

En este apartado se mencionarán los archivos que se encuentran en el proyecto que se entrega. Dentro de la carpeta principal del proyecto, todos los ficheros con código fuente de los cuales es importante tener información se encuentran en la carpeta /src. Ahí, después de acceder a la carpeta main, se encuentran dos ubicaciones: java y resources. La primera contiene el código java con el que se ha desarrollado el back-end del proyecto. El segundo contiene el código html/css de la interfaz de usuario de la que dispone la página web.

## **2.1 Interfaz de usuario**

Una vez se entra en la carpeta resources antes mencionada, hay otra división, /static, donde están localizadas las librerías css y JavaScript del framework Bootstrap, utilizado para el desarrollo de la interfaz de esta página web. Además, se almacenan imágenes utilizadas en ciertas vistas de la interfaz. Al volver atrás se puede acceder a la otra carpeta situada dentro de resources, llamada templates. En ella se encuentran todos los ficheros de las vistas. En primer lugar aparece el fichero con la pantalla de login, la pantalla principal (index.html) y el código de la barra de navegación (barranav), la cual se encuentra en todas las vistas para poder cambiar cómodamente de funcionalidad.

Posteriormente se puede comprobar que las vistas han sido divididas en carpetas dependiendo de la parte de funcionalidad a la que corresponden. CampeonatosView contiene la vista que muestra toda la información de un campeonato (CampeonatoDetail), el formulario de creación del campeonato (CampeonatosForm), el formulario de inscripción para el deportista (CampeonatosInscripciónForm), la lista de campeonatos (CampeonatosShowAll) y el formulario para guardar el resultado de un partido (ResultadoForm).

A continuación, se puede acceder a la carpeta PartidosView, en la que se encuentran PartidosForm (para el administrador será el formulario de partido promocionado y para el deportista el de partido ofertado) y partidos showAll (muestra la lista de partidos con su tipo y cierta información). La siguiente carpeta es PistasView, que contiene un formulario para crear una pista (PistasForm) y un listado de todas las pistas creadas (PistasShowAll). El directorio ReservasView contiene el formulario de reserva (ReservasForm), el listado de reservas para cada usuario (ReservasShowAll) y el horario disponible para un día escogido en el formulario (ReservasShowByFecha).

Por último, se entra en UsuariosView, que trata la gestión de usuarios y contiene la vista de la notificación (NotificaciónShow), el formulario de creación de usuario (UsuariosForm), toda la información de un perfil en concreto (UsuariosProfile) y la lista de usuarios (UsuariosShowAll).

## **2.2 Ficheros Java**

Como se ha mencionado anteriormente, el código fuente java del desarrollo back-end se encuentra en la carpeta java. Accediendo a través de esas carpetas llegaremos a /padelclub, donde las clases están divididas en varias carpetas, a excepción de una. Se trata de PadelApplication, la primera clase que se ejecuta y que comienza a ejecutar el resto del proyecto. En la carpeta Model se encuentran declaradas las clases, con sus atributos y se especifica cómo serán guardados en la base de datos. Hay ciertas clases de las cuales se puede deducir que representan gracias a su nombre, mientras que otras resultan un poco más ambiguas. Es, por ejemplo, el caso de ParejaCampeonatold, que almacena una pareja y un id del campeonato en el que está compitiendo. Al mismo tiempo, ParejaCampeonato incluye un objeto parejaCampeonatold e incluye como atributo sus puntos en la liga. Sucede prácticamente lo mismo para UsuarioPartidold (almacena un usuario y el id del partido al que está apuntado) y esto es un atributo de UsuarioPartido, la clase que es marcada como entidad para la base de

datos. Cabe destacar para acabar este apartado la necesidad de la clase Authority, para las funciones de login.

A pesar de que en Model se almacenan todos los atributos con sus tipos de dato, la funcionalidad CRUD se encuentra en la carpeta controller. Así, en CampeonatoController se encuentran las funciones de consulta, inscripción y guardado de campeonatos, en PartidoController el listado por tipo, guardado, buscado e inscripción de un deportista a alguno de los partidos. Sucede prácticamente lo mismo para las clases PistasController y ReservasController. Para finalizar, se encuentra UsuariosController, añadiendo a las anteriores la posibilidad de mostrar toda la información del usuario (reservas, campeonatos y partidos activos).

En la carpeta security se localiza el código relacionado con la encriptación de contraseñas para el login. De hecho, es en la clase Passgenerator donde se cifra la clave introducida al darse de alta en el sistema.

Uno de los directorios más importantes de este proyecto es Service/Impl, ya que es donde se encuentran las clases con la lógica de negocio y la funcionalidad más compleja de implementar. Por ejemplo, CampeonatoServiceImpl contiene las funciones de sorteo para la liga. PartidoServiceImpl contiene el código encargado de cerrar un partido una vez se hayan alcanzado los 4 jugadores. UsuarioPartidoServiceImpl tiene como principal objetivo devolver el listado de los jugadores anotados a los partidos ofertados y promocionados. Todas estas clases implementan unos métodos establecidos previamente en unas interfaces almacenadas en el directorio service/api

Por último lugar, se puede acceder a la carpeta /dao/api, formada por una serie de interfaces que a su vez son extensiones de un repositorio JPA y contienen las sentencias para comunicar las clases java con la base de datos.