



Documento
**DOCUMENTO TÉCNICO DEL
PROYECTO**

Plataforma para Clubes Deportivos

TracksYours

[code=TRACKS]



Nombre fichero:	DAW_PWR_TRACKS_UT01.4_Documento técnico.odt
Fecha de esta versión:	14/02/2026
Propiedad:	RHMS Sporting Developments

HISTORIAL DE REVISIONES

Fecha	Descripción	Autor
14/02/2026	Primera versión del <i>documento técnico del proyecto</i>	Roberto Himar Medina Sosa

Sumario

INTRODUCCIÓN.....	4
ARQUITECTURA GENERAL DEL SISTEMA.....	4
ESPECIFICACIÓN TÉCNICA DEL DESARROLLO.....	4
Tecnologías y frameworks de Front-End.....	4
Tecnologías y frameworks de Back-End.....	5
Tecnologías de implementación del modelo de datos.....	6
ESPECIFICACIÓN TÉCNICA DE LA PLATAFORMA DE DESPLIEGUE.....	6
TECNOLOGÍAS DE SOPORTE.....	7
OTRAS ESPECIFICACIONES.....	7

1 INTRODUCCIÓN

Este documento se constituye como uno de los entregables de la fase de análisis del proyecto *TracksYoours*. Pretende ser la referencia de consulta, para los interesados, de las soluciones tecnologías que se han decidido utilizar en la ejecución del citado proyecto. La finalización de la fase de análisis de requisitos y la definición del diseño a emplear nos permite escoger las tecnologías de más idónea aplicación para lograr el producto final. Esta elección también complementada con la consideración de las destrezas y experiencia del equipo del proyecto y la situación del entorno tecnológico actual.

Primeramente se dará una definición general de la arquitectura del sistema que se implementará en este proyecto, para continuar con un recorrido por: las tecnologías y soluciones empleadas en el desarrollo del producto, las tecnologías de despliegue del sistema cuando ya esté en explotación y por otras herramientas de soporte empleadas en el proyecto.

2 ARQUITECTURA GENERAL DEL SISTEMA

El producto resultante del proyecto será una aplicación Web tipo *landing page* implementada siguiendo un patrón de arquitectura de software **modelo-vista-controlador**, MVC en adelante, y con un estilo de despliegue **monolítico**. La decisión de esta arquitectura se basa en el hecho de que este proyecto es una prueba de concepto o piloto de un producto mínimo viable de análisis de respuesta o aceptación en el mercado. Posibles ampliaciones o versiones posteriores valorarán otro tipo de arquitectura distribuida, incluso con presencia de API, pero no es objetivo del actual proyecto.

La arquitectura monolítica del producto final se presenta en el siguiente esquema:

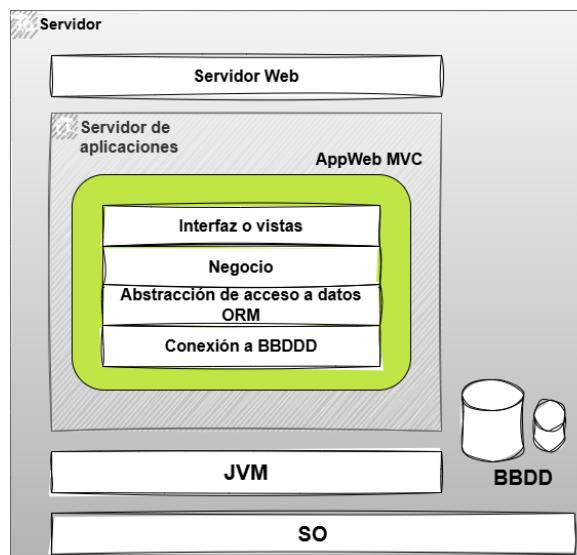


Figura 1: esquema general del sistema desplegado

3 ESPECIFICACIÓN TÉCNICA DEL DESARROLLO

Se listan a continuación las tecnologías que se emplean para la construcción del producto final, la aplicación web *TrackYours*, con indicación de versiones y justificación de la elección.

3.1 Tecnologías y frameworks de Front-End

Las vistas o interfaces de la aplicación web emplearán tecnologías marco de *front-end*, éstas son, HTML5, CSS3 y JS y haciendo uso del framework **BootStrap**. Además indicar la presencia de un soporte al renderizado de vistas, que si bien no es puramente de *front-end*, semánticamente se podría enmarcar aquí. Tal soporte se añade mediante el framework Thymeleaf.

La siguiente tabla muestra las principales tecnologías empleadas:

Id	Designación	Versión	Tipo	Función o uso
 HTML5	5		Lenguaje de marcado	Definir contenido y estructura
 CSS3	3		Lenguaje de estilos	Definir estilos
 JavaScript (JS)	ECMAScript ES6+ (ES2025)		Lenguaje front-end	Aporta comportamiento dinámico
 BootStrap	5.x		Framework de front-end	Aporta componentes de diseño
 Thymeleaf	3.1.x		Framework de renderizado	Permite renderizado de vistas

- La elección de las tecnologías HTML5 y CSS3 atiende a que son un estándar ampliamente adoptado en las interfaces web y que las versiones son las liberadas desde hace años y consideradas estables.
- El uso de JavaScript en última versión reseñable ES6+ obedece a las mismas justificaciones indicadas para HTML5 y CSS3. Se empleará en aportar algún comportamiento dinámico puntual que se salga del alcance de BootStrap, pero básicamente para aplicación de características de validación de los inputs de formularios que posibilita.
- Se decide emplear el framework BootStrap por la versatilidad que permite para introducir componentes de forma fácil, incluyendo éstos ya comportamiento responsive, lo cual ahorra mucho tiempo de diseño del *frontend*. También aportan el uso de modales de forma fácil, muy útiles para nuestro prototipo de interfaz acordado.
- Para aportar dinamismo a la hora de mostrar la información de los items (clubes, usuarios, actividades, publicaciones, productos), presentes en el sistema se hace útil Thymeleaf, un lenguaje de renderizado de vistas que permite desde instancias de modelos de aquellos tipos de objetos, mostrar su información contenida en las vistas generadas de una manera fácil.

3.2 Tecnologías y frameworks de Back-End

La implementación de toda la lógica de negocio y el acceso a datos se implementará en el *backend* de la aplicación web, basada éste en el framework **Spring Boot**. Por tanto, el lenguaje base será Java. Indicar también que se hará uso de una capa de acceso a datos basada en ORM haciendo uso del framework **Hibernate**.

La siguiente tabla muestra las principales tecnologías empleadas:

Id	Designación	Versión	Tipo	Función o uso
 Spring Boot	4.0.x		Framework de <i>backend</i>	Permite app con MVC y muchas características
 Hibernate (Spring Data)	7.2.x		Framework de acceso a datos	Implementa un ORM, usando JPA como interfaz
 Java (JDK)	17 (LTS)		Lenguaje de desarrollo backend	Lenguaje robusto y ampliamente utilizado
 Maven	3.9.x		Gestión de dependencias y artefactos	Permite gestión de dependencias de forma fácil

- La elección del lenguaje Java está basada en la madurez y robustez del lenguaje, que facilitan la disponibilidad de recursos y soporte. A ello unir que es el lenguaje que más conoce el equipo de desarrollo y en el que posee mayor destrezas, lo cual puede inferir en reducción de tiempos de desarrollo. Se hará uso del JDK 17 por ampliar compatibilidad y por ser LTS.
- Una vez escogido el lenguaje, el framework Spring Boot es el predominante para desarrollo en Java con patrón MVC, de ahí su elección. Además incluye componentes de valor como son Spring Web o Spring Security, este último aportando características que permiten implementar la seguridad de acceso a la aplicación.
- Para la gestión de dependencias y como repositorio de artefactos se incorpora el producto Maven. Además permitirá la automatización de la construcción (*build*). Es ampliamente utilizado.
- Finalmente, para facilitar la implementación de un modelo de acceso a datos de tipo ORM, se hará uso del framework ya existente JPA Spring, que es una implementación de Hibernate hecha por Spring y que atiende a la especificación JPA (*Java Persistence API*).

3.3 Tecnologías de implementación del modelo de datos

Como se indicó en el apartado anterior, la capa de acceso a datos se va a implementar con Spring Data pero se debe definir la implementación del esquema de base de datos. En este sentido se ha decidido que el esquema de base de datos se implemente en un servidor de base de datos corriendo el gestor de base de datos MySQL.

Por tanto:

Id	Designación	Versión	Tipo	Función o uso
	MySQL	8.0.x	Gestor de base de datos	Implementa el servidor de base de datos
	MySQL Connector/J	9.6.0	JDBC de acceso a la BBDD MySQL API estándar de Java para conectarse a MySQL	

- MySQL es un gestor de base de datos eficiente y adecuado a la dimensión del proyecto. Además de acceso y uso permitido en proyectos de índole educativa o de libre distribución, como es nuestro caso.
- Se usará el conector oficial de la API de Java para conectarse a MySQL liberado por Oracle.

4 ESPECIFICACIÓN TÉCNICA DE LA PLATAFORMA DE DESPLIEGUE

La solución *TrackYours* se ha definido para ser desarrollada, compilada, empaquetada y desplegada como una entidad monolítica; esto es, todos sus componentes como una única aplicación web. Todo corriendo en un único servidor de aplicaciones, que además incluirá el servidor de base de datos albergando el esquema de base de datos. Las especificaciones de disponibilidad y criticidad de la aplicación, unido a ser una *PoC* así lo posibilitan. Más tarde se pueden valorar escalabilidades horizontales o verticales de la plataforma si se detectará necesidad.

Con ello, indicar que la aplicación web se liberará en una sola máquina servidor con SO **Linux**, haciendo uso del servidor de aplicaciones **TomCat**. La misma máquina albergará el servidor de base de datos MySQL.

La siguiente tabla muestra las tecnologías presentes en la arquitectura de despliegue:

Id	Designación	Versión	Tipo	Función o uso
	TomCat	11.0.x	Servidor de aplicaciones Java	Permite servir la appWeb.
	JVM (Java Virtual Machine)	17 LTS	Entorno de ejecución Java	Permite interpretar aplicaciones Java
	MySQL	8.0.x	Gestor de base de datos	Implementa el servidor de base de datos
	Linux	Ubuntu 22.04 LTS	Sistema operativo	Sistema operativo eficiente para servidores

- La aplicación web *TrackYours* se desplegará sobre un servidor de aplicaciones TomCat. Es un servidor para aplicaciones Java ligero y adecuado a la dimensión de nuestra aplicación (no corporativa). Dispone de interfaz gráfica que facilita los despliegues de los empaquetados .war
- Obviamente la máquina actuando de servidor requerirá la presencia del entorno de ejecución de Java, esto es, la JVM.
- Como servidor de base de datos se ha escogido MySQL, con las justificaciones ya descritas en el apartado 3.3 anterior.
- Todas estas tecnologías se albergan en una máquina actuando de servidor que correrá un sistema operativo Ubuntu 22.04 LTS (Linux). Los sistemas Linux son una opción muy válida por eficiencia actuando como servidores.

5 TECNOLOGÍAS DE SOPORTE

El ejecución de cualquier proyecto de desarrollo software implica el uso de multitud de herramientas complementarias a las puramente asociadas al código y a la plataforma de despliegue. En las distintas fases de desarrollo de este proyecto se han usado las herramientas que a continuación se listan para diferentes tareas o campos de actuación. Vemos la tabla de las más relevantes:

Id	Designación	Versión	Tipo	Uso
	Balsamiq	4.8.6	Herramienta de prototipado	Crear <i>wireframing</i> Lo-Fi de las interfaces
	Git/GitHub	2.53.x / 3.19.x	Control de versiones	Mantener repositorio de versiones del proyecto
	Trello/GitHub projects	-- / 3.19.x	Metodologías SCRUM/Kanban	Gestión de fases y tareas del proyecto
	Eclipse	2025-12	IDE de desarrollo	Generación, construcción y depuración de código
	Chrome/Edge	145.0.x / 144.0.x	Navegadores	Depuración y testing

- En la fase de análisis además de documentar y generar gráficos con la herramienta online *Draw.io*, se ha generado un prototipo de baja/media fidelidad con herramienta de escritorio Balsamiq. Es la herramienta de *wireframing* escogida pues no se requería el nivel de detalle de Figma
- El control de versiones se hace con la herramienta Git (en entorno local de desarrollo) y con GitHub a fin de mantener un repositorio en la nube y compartido con todo el equipo interesados. Es una herramienta gratuita y la más ampliamente aceptada, con lo cual no deja dudas se elección a tal fin.
- La gestión de las fases y tareas del proyecto no requiere de una herramienta muy potente pues las dimensiones del proyecto y del equipo no lo infieren. Por ello se han manejado dos opciones: Trello y GitHub Projects, decantándonos finalmente por la segunda de éstas
- En cuanto al IDE o entorno de desarrollo nos hemos decantado por Eclipse, por ser conocida por el equipo de desarrollo. Es quizás menos versátil que VS Code, pero muy robusta y también permite plugins auto-completado, depuración, refactorización, etc.. Además existe una integración perfecta del framework Spring Boot en el IDE Eclipse, la suite Spring Tools 4.
- Finalmente la aplicación se pretende sea compatible y testada con los tres navegadores dominantes del mercado: Chrome, Edge y Firefox. Por ello se emplearán estos tres para pruebas de validación funcional y depuración del *frontend*.

6 OTRAS ESPECIFICACIONES

Se listan otras consideraciones importantes referidos a las tecnologías o soluciones integradas en la aplicación *TrackYours*:

- No hay conexión con interfaces o fuentes de datos externas. Es decir, no se consume ningún tipo de API para obtención de datos. Esto se enmarca dentro de la estrategia del proyecto de generar el mínimo producto viable para lograr el objetivo y requisitos del sistema.
- En cuanto a la implementación de la seguridad definida por requisitos se hará uso de características que nos ofrece el componente **Spring Security** de Spring. Este nos permite implementar autentificación por contraseña con diversas opciones de cifrado disponibles, así como sesión mediante token JWT.