

IES FRANCISCO DE GOYA

GymTracker

Proyecto fin de Ciclo Formativo

Daniel Marín Gómez
Álvaro Gallego Yáñez
David Martín Díaz



Contenido

1 Introducción.....	7
1.1 Objetivos iniciales.....	7
1.2 Tecnologías.....	7
1.2.1 HTML5.....	7
1.2.2 CSS3.....	8
1.2.3 Bootstrap.....	8
1.2.4 JavaScript.....	8
1.2.4.1 Chart.js.....	8
1.2.4.2 Axios.....	9
1.2.5 MySQL y MySQL Workbench.....	9
1.2.6 GitHub.....	9
1.2.7 Java.....	10
1.2.8 Spring Boot.....	10
1.2.8.1 Dependencias para la API de ejercicios.....	11
1.2.8.2 Dependencias para Gymtracker.....	11
1.2.9 Docker.....	12
1.2.10 AWS (instancia linux amazon).....	12
1.2.11 Shell.....	13
2 Análisis de Requisitos.....	13
2.1 Mapa de pantallas.....	14
2.2 Manual de usuario.....	15
2.2.1 Acceso a GymTracker.....	15
2.2.2 Inicio.....	16
2.2.2.1 Inicio Sesión.....	16
2.2.2.2 Crear usuario.....	17
2.2.2.3 Crear Perfil.....	18
2.2.3 Menú.....	18
2.2.4 Nueva rutina.....	20
2.2.4.1 Añadir ejercicio.....	20
2.2.4.2 Añadir Serie.....	21
2.2.5 Mis entrenamientos.....	22
2.2.6 Ejercicios.....	24
2.2.7 Estadísticas.....	25
2.2.8 Clasificación.....	27
2.2.9 Nuestras rutinas.....	28
2.2.10 Calendario.....	29
2.2.11 Perfil.....	29
2.2.11.1 Editar perfil.....	30
2.2.11.2 Cambiar contraseña.....	31

2.2.12 Menú Sidebar.....	32
2.2.13 Error.....	33
2.3 Manual de administrador.....	34
2.3.1 Administrar Usuarios.....	34
2.3.1.1 Cambiar contraseña del usuario.....	35
2.3.1.2 Acceder a la cuenta del usuario.....	35
2.3.1.3 Eliminar usuario.....	35
2.3.1.4 Crear usuario.....	36
3. Adecuación del entorno de trabajo.....	37
3.1 Instalación de MySQL:.....	37
3.2 Instalación de Git:.....	37
3.3 Clonación del Repositorio:.....	37
3.4 Instalación de JDK:.....	37
3.4.1 Instalación de JDK en Linux.....	38
3.4.2 Instalación de JDK en Windows.....	38
4.4.2.1 Configurar la variable de entorno JAVA_HOME:.....	38
3.5 Instalación de Maven:.....	38
3.5.1 Instalación de Maven en Linux.....	38
3.5.2 Instalación de Maven en Windows.....	38
3.5.2.1 Configurar la variable de entorno MAVEN_HOME:.....	39
3.6 Configuración del IDE (STS - Spring Tool Suite).....	39
3.6.1 Instalación de Spring Tool Suite:.....	39
3.6.2 Configuración de Lombok:.....	39
3.6.3 Importar Proyectos con STS:.....	39
3.7 Configuración del IDE (IntelliJ IDEA).....	40
3.7.1 Instalación de IntelliJ IDEA.....	40
3.7.2 Importar Proyectos con IntelliJ IDEA:.....	40
4. Arquitectura del proyecto.....	41
4.1 Arquitectura de la API.....	41
4.1.1 Estructura y Funcionalidad.....	42
4.1.2 Recursos.....	42
4.1.3 Tecnologías Utilizadas.....	43
4.2 Arquitectura de la APP.....	44
4.2.1. Configuración.....	45
4.2.2. Controladores (Controllers).....	45
4.2.3. Data Transfer Objects (DTO).....	45
4.2.4. Servicios (Services).....	45
4.2.5. Recursos (Resources).....	46
4.2.6 Dependencias.....	46
4.2.7 Mapeo con ORM.....	46
4.2.8 Flujo de Datos.....	46
4.2.9 Gestión de Seguridad y Sesiones.....	46

4.2.10 Uso de Lombok.....	47
4.2.11 Conclusión.....	47
5. Desarrollo de la base de datos.....	47
5.1 Arquitectura de la base de datos.....	48
5.1.1 Esquema de la base de datos:.....	48
5.1.2 Tablas Principales:.....	48
5.1.3 Relación entre Rutinas, Ejercicios y Series:.....	48
5.1.4 Relaciones:.....	49
5.1.5 Claves Externas y Borrados en Cascada:.....	49
5.2 Explicación y justificación de la base de datos.....	49
5.2.1 Integridad de los Datos:.....	49
5.2.2 Escalabilidad:.....	50
5.2.3 Eficiencia en Consultas:.....	50
5.2.4 Consistencia y Mantenibilidad:.....	50
6. Desarrollo del proyecto.....	50
6.1 Desarrollo de la API.....	50
6.1.1 Configuración CORS para el acceso a la API.....	51
6.2 Desarrollo de la APP.....	52
7. Despliegue.....	52
7.1 Despliegue en Aws.....	52
7.1.1 Lanzar instancia.....	52
7.1.2 Conectarse a la instancia.....	55
7.1.3 Preparar instancia para el despliegue.....	56
7.1.4 Ejecutar GymTracker.....	57
7.1.5 IP Elástica.....	59
7.2 Despliegue en local con Docker.....	62
7.2.1 Instalación de docker.....	62
7.2.1.1 Linux:.....	62
7.2.1.2.Para Windows:.....	62
7.2.2 Preparación para el despliegue.....	65
7.2.3 Despliegue.....	70
7.3 Despliegue en local con Spring Tool Suite (STS).....	71
7.3.1 Prerrequisitos.....	71
7.3.1.1 Instalación de Prerrequisitos.....	71
Windows.....	71
7.3.1.1.1 Instalar JDK.....	71
7.3.1.1.2 Instalar Git.....	71
7.3.1.1.3 Instalar Spring Tool Suite (STS).....	72
7.3.1.1.4 Instalar MySql.....	72
Linux.....	76
7.3.1.2.1 Instalar JDK.....	76
7.3.1.2.2 Instalar Git.....	76

7.3.1.2.3 Instalar Spring Tool Suite (STS).....	76
7.3.1.2.4 Instalar MySql.....	77
7.3.2 Clonar el Repositorio desde GitHub.....	78
Windows.....	78
Linux.....	79
7.3.3 Instalar la base de datos.....	79
Windows.....	79
Linux.....	80
7.3.4 Ejecutar la Aplicación.....	80
Windows.....	80
Linux.....	84
7.3.5 Acceder a la Aplicación.....	87
7.4 Despliegue en local desde Github.....	88
7.4.1 Descarga de los archivos JAR.....	88
7.4.1.1 Linux:.....	88
7.4.1.2 Windows:.....	88
7.4.2 Preparación para el despliegue.....	89
7.4.2.1 Windows:.....	90
7.4.2.1 Linux:.....	92
7.4.2.1.1 Clonar repositorio de Github para windows.....	93
7.4.2.1.2 Clonar repositorio de gitHub para linux.....	94
7.4.2.2.1 Volcar base de datos en windows.....	94
7.4.2.2.2 Volcar base de datos en linux.....	95
7.4.3 Despliegue.....	96
7.4.3.1 Despliegue en windows:.....	96
8. Pruebas (!IMPORTANT puntúa bien).....	97
9. Conclusiones.....	99
10. Bibliografía.....	99
10.1 Enlaces de descarga.....	99
10.1.1 Oracle JDK.....	99
10.1.2 OpenJDK.....	99
10.1.3 Git.....	100
10.1.4 Spring Tool Suite.....	100
10.1.5 MySql.....	100
10.1.6 Lombok.....	100
10.2 Documentación Oficial.....	100
10.2.1 Spring Boot.....	100
10.2.2 Java.....	100
10.2.3 MySql.....	100
10.2.4 JavaScript.....	100
10.2.5 Axios.....	100
10.2.6 Bootstrap.....	100

10.2.7 HTML.....	100
10.2.8 CSS.....	100
10.2.9 AWS.....	100
10.2.10 Docker.....	101
10.2.11 Git.....	101
10.2.12 Lombok.....	101
10.2.13 Apache Tomcat.....	101
10.3 Otras páginas de consulta.....	101
10.3.1 Stack Overflow.....	101
10.3.2 W3Schools.....	101
10.3.3 ChatGPT.....	101
10.3.4 Microsoft designer.....	101
10.4 Páginas importantes en el desarrollo.....	101
10.4.1 Duck DNS.....	101
10.4.2 AWS Academy.....	101
10.4.3 Github.....	101
10.5 Enlaces para el despliegue en Docker.....	102
10.5.1 Tutorial para instalar docker.....	102
10.5.2 Instalar WSL.....	102
10.5.3 Wait-for-it.....	102
10.5.4 Instalar docker.....	102

1 Introducción

El documento presente es la memoria del Proyecto de Fin de Ciclo Formativo de DAW (Desarrollo Aplicaciones Web), Gymtracker es una aplicación web en la que los usuarios tienen un perfil y pueden gestionar sus rutinas de gimnasio, así como seguir un historial de las misma, para tener así conciencia sobre su progresión en el gimnasio.

1.1 Objetivos iniciales

Como finalidad, el proyecto pretende construir una página web completa , la aplicación gira entorno a la arquitectura de MVC (Modelo Vista Controlador) utilizando como eje principal el framework de Spring boot, usando una temática de gimnasio en la que cada usuario tiene su perfil y sus rutinas las cuales puede gestionar y guardar para tener por así decirlo un diario de sus entrenamientos.

Los objetivos iniciales que se querían abarcar y superar son:

- Proporcionar a los usuarios una plataforma intuitiva y eficiente para planificar y seguir su progreso en el gimnasio.
- Consolidar conocimientos sobre el framework de Spring Boot así como explorar funcionalidades como lo son la seguridad con spring security.
- Implementar una arquitectura MVC en Gymtracker con Spring Boot, donde los datos se gestionan en servicios que interactúan con la base de datos, utilizando DTOs para mapear y transmitir los datos entre el controlador y las vistas.
- Desarrollar una API separada para gestionar ejercicios, la cual consumiría Gymtracker, ampliando así la funcionalidad del sistema y mejorando la modularidad y escalabilidad del proyecto.

1.2 Tecnologías

Los lenguajes de programación/marcas y tecnologías/entornos usados son:

1.2.1 HTML5

HTML, o "HyperText Markup Language" (Lenguaje de Marcado de Hipertexto), sirve como el esqueleto fundamental de nuestra aplicación web al definir su estructura y contenido. Utilizado principalmente para desarrollar las vistas, HTML organiza la información y elementos visuales de la página web mediante etiquetas que representan diversos elementos como texto, imágenes, enlaces y formularios, facilitando así la presentación de contenido al usuario final en el navegador web.

1.2.2 CSS3

CSS3, o "Cascading Style Sheets" (Hojas de Estilo en Cascada), complementa a HTML al proporcionar el diseño y estilo visual de una aplicación web. Utilizado para definir la apariencia de los elementos HTML, CSS3 permite personalizar aspectos como el color, la tipografía, el espaciado y el diseño de la página, lo que mejora significativamente la presentación estética y la experiencia del usuario en el navegador web.

1.2.3 Bootstrap

Bootstrap es un marco de trabajo front-end de código abierto que facilita el desarrollo rápido y sensible de sitios web y aplicaciones web. Al aprovechar las capacidades de CSS3 y JavaScript, Bootstrap proporciona una amplia variedad de componentes y estilos predefinidos, como botones, barras de navegación, cuadros modales y diseños de rejilla responsivos. Esto permite a los desarrolladores crear interfaces de usuario atractivas y funcionales de manera eficiente, y nos sirvió como herramienta integral para la maquetación y los estilos principales de la página web.

1.2.4 JavaScript

JavaScript es un lenguaje de programación de alto nivel, interpretado y orientado a objetos, utilizado principalmente para agregar interactividad y dinamismo a las páginas web. En nuestra aplicación Gymtracker, JavaScript fue de gran utilidad para añadir dinamismo a las páginas, permitiendo a los usuarios interactuar con la interfaz de manera fluida y eficiente. Con JavaScript, pudimos manipular el contenido de la página, responder a eventos del usuario, validar formularios, realizar solicitudes a servidores web para cargar datos dinámicamente y crear efectos visuales, lo que contribuyó significativamente a mejorar la experiencia de usuario y la funcionalidad de la aplicación.

1.2.4.1 Chart.js

Chart.js es una biblioteca de JavaScript que facilita la creación de gráficos y diagramas interactivos en aplicaciones web. Proporciona una variedad de tipos de gráficos, incluyendo gráficos de barras, líneas, radiales y más, que pueden ser personalizados y animados. En Gymtracker, utilizamos Chart.js para generar diagramas de barras en la pantalla de estadísticas de la aplicación. Esto nos permite visualizar de manera clara y

atractiva el progreso y los datos de los usuarios, mejorando la comprensión y el análisis de su rendimiento en el gimnasio. Gracias a Chart.js, podemos ofrecer una experiencia visualmente enriquecida que ayuda a los usuarios a interpretar sus datos de manera más efectiva.

1.2.4.2 Axios

Axios es una biblioteca de JavaScript utilizada para realizar solicitudes HTTP desde el navegador o desde Node.js. En el contexto de Gymtracker, hemos utilizado Axios para consultar la API de ejercicios externa y para realizar peticiones a controladores que responden con datos en formato JSON en lugar de plantillas HTML completas. Con Axios, pudimos realizar solicitudes HTTP de manera fácil y eficiente, lo que nos permitió obtener y enviar datos entre la aplicación y la API de ejercicios, así como entre la aplicación y los controladores del backend. Esto ha sido fundamental para obtener información dinámica y actualizada sobre ejercicios y rutinas de ejercicio, y para interactuar con la lógica del backend de manera más ágil y flexible.

1.2.5 MySQL y MySQL Workbench

Con MySQL y MySQL Workbench, creamos la base de datos de Gymtracker. MySQL es un sistema de gestión de bases de datos relacional de código abierto ampliamente utilizado, mientras que MySQL Workbench proporciona una interfaz gráfica para diseñar, modelar, administrar y consultar bases de datos MySQL de manera intuitiva. Estas herramientas nos permitieron diseñar y gestionar eficientemente la estructura de la base de datos de Gymtracker, lo que fue fundamental para almacenar y organizar la información de los usuarios, las rutinas de ejercicio y otros datos relevantes para el funcionamiento de la aplicación.

1.2.6 GitHub

GitHub es una plataforma de desarrollo colaborativo que utiliza el sistema de control de versiones Git, permitiendo a los desarrolladores trabajar juntos en proyectos de software de manera eficiente y transparente. En GitHub, los proyectos se alojan en repositorios, donde se almacena el código fuente y se registra el historial de cambios a lo largo del tiempo. Esto facilita la colaboración, ya que múltiples desarrolladores pueden trabajar simultáneamente en el mismo proyecto, realizar contribuciones, revisar y comentar el

código de otros, y realizar un seguimiento de los problemas o tareas pendientes a través del sistema de seguimiento de problemas integrado.

Gracias a GitHub, podemos tener un control de versiones efectivo para el desarrollo de Gymtracker. Utilizamos un enfoque basado en ramas, donde cada tarea o issue se asocia con una rama específica. Esto nos permite trabajar de manera organizada y modular, ya que cada miembro del equipo puede trabajar en su propia rama sin interferir con el trabajo de los demás. Una vez que se completa una tarea, se fusiona la rama correspondiente en la rama principal, lo que facilita la integración continua y garantiza un progreso constante del proyecto. Esta metodología nos brinda un control detallado sobre los cambios realizados en el código, permitiendo una colaboración fluida y eficiente entre los miembros del equipo.

1.2.7 Java

Java es el lenguaje de programación principal utilizado en Gymtracker, ya que la aplicación se desarrolla utilizando el framework Spring Boot, que está basado en Java. Java es un lenguaje de programación de propósito general ampliamente utilizado en el desarrollo de aplicaciones empresariales y web debido a su portabilidad, robustez y seguridad. Spring Boot, por su parte, proporciona un entorno de desarrollo ágil y eficiente para la creación de aplicaciones web Java, facilitando tareas como la configuración, la gestión de dependencias y la implementación de patrones de diseño. Gracias a Java y Spring Boot, podemos desarrollar Gymtracker de manera efectiva, aprovechando las características y funcionalidades que ofrecen para garantizar la fiabilidad, la escalabilidad y el rendimiento de la aplicación.

1.2.8 Spring Boot

Spring Boot es un framework de desarrollo de aplicaciones Java que ofrece un entorno simplificado y eficiente para crear aplicaciones web y servicios RESTful. En Gymtracker, hemos utilizado Spring Boot para desarrollar tanto el backend principal de la aplicación como la API RESTful de ejercicios, la cual está separada del backend principal. Para el backend de Gymtracker, Spring Boot nos proporciona las herramientas necesarias para gestionar la lógica de la aplicación, la conexión a la base de datos y la seguridad a través de Spring Security. Gracias a Spring Boot, pudimos crear una aplicación robusta y segura que cumple con los requerimientos de Gymtracker y garantiza una experiencia de usuario óptima.

1.2.8.1 Dependencias para la API de ejercicios

Las dependencias utilizadas en la API de ejercicios son las siguientes:

- **Spring Boot DevTools:** Facilita el desarrollo y la depuración al proporcionar herramientas para la recarga automática de cambios en la aplicación durante el desarrollo.
- **Spring Web:** Proporciona funcionalidades para el desarrollo de aplicaciones web, como la creación de controladores RESTful y la gestión de solicitudes HTTP.
- **JDBC API:** Ofrece una interfaz de programación para acceder a bases de datos relacionales desde aplicaciones Java, permitiendo la conexión y manipulación de datos de forma eficiente.
- **Spring Data JPA:** Simplifica el acceso y la manipulación de datos en bases de datos relacionales utilizando el framework de persistencia JPA (Java Persistence API), integrado con Spring para una configuración y uso sencillos.
- **H2 Database:** Proporciona una base de datos relacional en memoria que se puede utilizar para pruebas y desarrollo de aplicaciones, ofreciendo un entorno ligero y fácil de configurar para la gestión de datos durante el ciclo de desarrollo.

Utilizamos Spring Data JPA para simplificar los filtrados de la API, permitiendo una fácil manipulación y consulta de los datos de ejercicios establecidos. Además, empleamos H2 Database porque la API solo muestra ejercicios predefinidos y permite únicamente solicitudes GET para mostrar datos, lo que se alinea con la naturaleza de solo lectura de la API, sin permitir la creación, modificación o eliminación de ejercicios.

1.2.8.2 Dependencias para Gymtracker

Las dependencias utilizadas en la aplicación de Gymtracker son las siguientes:

- **Spring Boot DevTools:** Facilita el desarrollo y la depuración al proporcionar herramientas para la recarga automática de cambios en la aplicación durante el desarrollo.
- **Spring Web:** Proporciona funcionalidades para el desarrollo de aplicaciones web, como la creación de controladores RESTful y la gestión de solicitudes HTTP.
- **JDBC API:** Ofrece una interfaz de programación para acceder a bases de datos relacionales desde aplicaciones Java, permitiendo la conexión y manipulación de datos de forma eficiente.
- **Spring Data JPA:** Simplifica el acceso y la manipulación de datos en bases de datos relacionales utilizando el framework de persistencia JPA (Java Persistence API), integrado con Spring para una configuración y uso sencillos.
- **MySQL Driver:** Proporciona conectividad JDBC para interactuar con la base de datos MySQL.
- **Lombok:** Simplifica el desarrollo eliminando la necesidad de escribir código boilerplate en Java.

- **Spring Session:** Gestiona la sesión del usuario en aplicaciones Spring, permitiendo almacenar y compartir información de sesión entre múltiples solicitudes HTTP.
- **Spring Security:** Proporciona una capa de seguridad robusta para aplicaciones Spring, gestionando la autenticación, autorización y otras características de seguridad.
- **Thymeleaf:** Motor de plantillas para la creación de vistas HTML en aplicaciones Spring, permitiendo una integración fácil y flexible con datos del servidor.

Utilizamos Thymeleaf para la creación de plantillas HTML en Gymtracker, lo que nos permite generar vistas dinámicas e integrar fácilmente datos del servidor en la interfaz de usuario. Con Spring Data JPA, en lugar de utilizar métodos predefinidos, empleamos nuestras propias queries personalizadas para acceder a datos específicos de la base de datos mediante la inyección de `EntityManager`. Spring Security se encarga de garantizar la seguridad de la aplicación, gestionando la autenticación, autorización y otros aspectos de seguridad para proteger los datos y los recursos de manera efectiva.

Spring Session nos ayuda a gestionar los roles de los usuarios y otras características de la sesión en la aplicación, permitiendo un control adecuado sobre los accesos y acciones de los usuarios. Lombok es empleado para simplificar la sintaxis en la creación de DTOs, reduciendo la cantidad de código boilerplate y facilitando la inyección de dependencias a través de la generación automática de constructores y otros métodos.

1.2.9 Docker

Docker es una plataforma de contenerización que permite a los desarrolladores empaquetar aplicaciones y sus dependencias en contenedores portátiles y ligeros. Estos contenedores aseguran que la aplicación se ejecute de manera consistente en diferentes entornos, desde el desarrollo hasta la producción. En Gymtracker, utilizamos Docker para el despliegue de la aplicación. Creamos contenedores a partir de los archivos JAR de Gymtracker para la aplicación principal y la API de ejercicios, y utilizamos Docker Compose para orquestar estos contenedores junto con un contenedor para la base de datos. Esto nos permite desplegar la aplicación de manera eficiente y reproducible, garantizando que todos los componentes funcionen juntos de manera coherente en cualquier entorno.

1.2.10 AWS (instancia linux amazon)

AWS (Amazon Web Services) es una plataforma de servicios en la nube que proporciona una amplia gama de servicios de infraestructura escalables y de bajo costo. En GymTracker, utilizamos una instancia de Amazon Linux en AWS para el despliegue de la aplicación. Configuramos la instancia para alojar nuestra aplicación descargando los archivos JAR necesarios y configurando la base de datos MySQL directamente en la instancia. Esto incluyó la instalación de MySQL, la creación de la base de datos y la importación de los datos necesarios. Este enfoque nos permite aprovechar la escalabilidad y la fiabilidad de la infraestructura de AWS, asegurando un rendimiento óptimo y una alta disponibilidad de la aplicación.

1.2.11 Shell

El shell es una interfaz de línea de comandos que permite a los usuarios interactuar con el sistema operativo mediante la ejecución de comandos. En Linux, Bash (Bourne Again Shell) es uno de los shells más comunes y proporciona un entorno poderoso para la creación y ejecución de scripts.

Tanto en el despliegue en Docker como en AWS, utilizamos scripts en Bash para automatizar y gestionar diversas tareas. En el despliegue con Docker, empleamos el script **wait-for-it** en el entorno Bash, que descargamos de un repositorio de GitHub. Este script asegura que el contenedor de la aplicación GymTracker espere hasta que la base de datos esté lista antes de crearse. Para implementar esto, incluimos el script en el **Dockerfile.app** y lo ejecutamos durante la ejecución de **docker-compose**. En el despliegue en AWS, creamos dos scripts en Bash: uno para levantar los archivos JAR de la aplicación y otro para detenerlos. Estos scripts nos permiten iniciar y detener la aplicación de manera eficiente, facilitando el proceso de administración y mantenimiento en la instancia de Amazon Linux.

2 Análisis de Requisitos

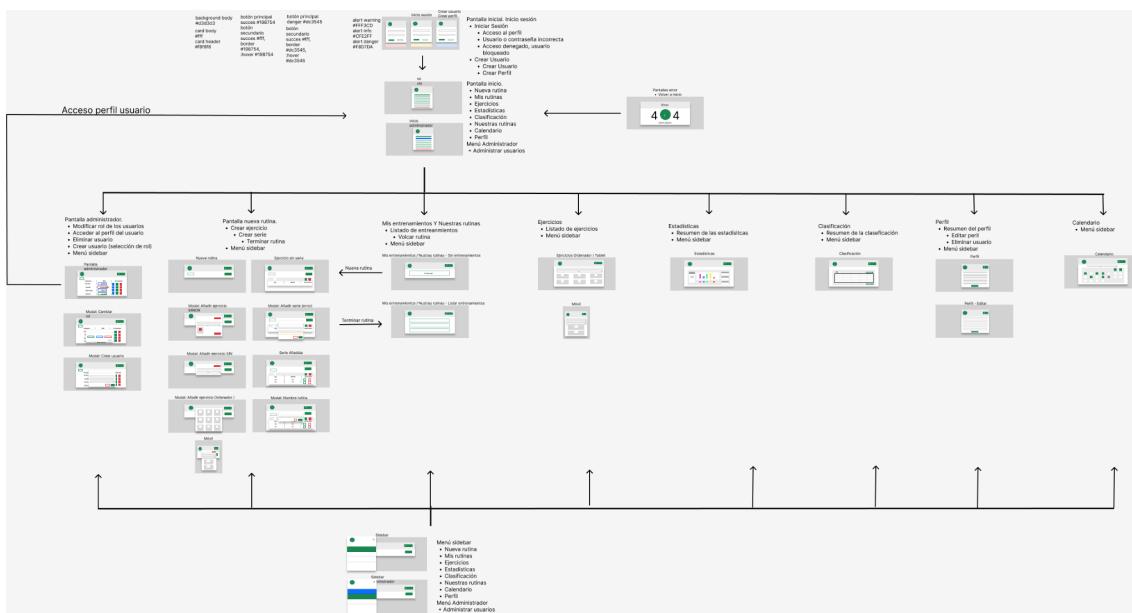
El proyecto, GymTracker, tiene como motivo la creación de una aplicación web para el control y gestión individual de los entrenamientos o rutinas de gimnasio. Con sus características, a diferenciar entre: Una página de inicio donde poder tener como primera vista de la aplicación un login para los usuarios de la aplicación y un formulario de registro para los que aún no lo son. Gestionando los posibles errores de usuario o contraseña con el aviso correspondiente. En los casos de creación de nuevos usuarios comprobar que el nombre de usuario sea válido al igual que los valores del perfil, con el aviso correspondiente. Es necesario tener un menú donde se encuentren todas las opciones que el usuario, ya sea administrador o no, pueda utilizar. Una página que permita la creación de las rutinas/entrenamientos y el guardado de estas, ligada a una página de “mis entrenamientos”, que muestre todos las rutinas que ha realizado el usuario con la aplicación permitiendo volver a recrear cualquiera de ellas o comenzar una de cero. Un listado de los posibles ejercicios a realizar desde la aplicación con una pequeña descripción e imágenes para su correcta ejecución. Una página de estadísticas donde el usuario puede consultar información interesante acerca de sus entrenamientos. Una pantalla de clasificación que permite comparar tus estadísticas frente a las del resto de usuarios. Una página con rutinas predeterminadas para aquellos usuarios que no desean crear una desde cero y no sepan cómo comenzar. Un resumen de tu perfil que permite al usuario

modificarlo en todo momento respetando las mismas restricciones que encontró en el momento de la creación del perfil. Una página de Administración donde se controla que únicamente los usuarios con el rol de administradores puedan acceder. En ella se encuentran las funcionalidades de creación y borrado de usuarios, cambios de rol y de contraseña o el acceso a cuentas de los usuarios. Y por último y no menos importante, ofrece una opción de cierre de sesión para que los usuarios puedan cerrar su sesión de manera segura.

De forma más gráfica hemos creado un mapa de pantallas con el esquema de cada una de ellas y las transiciones entre sí.

2.1 Mapa de pantallas

<https://www.figma.com/design/l7dGsOrB1fvV1kaDOwWcVj/Pantallas-GymTracker?node-id=0-1&t=HyYpMlcOaltamNlc-0>



Para continuar les mostramos el manual de usuario y administrador, presentando las distintas funcionalidades del proyecto y el correcto uso de la aplicación.

2.2 Manual de usuario

2.2.1 Acceso a GymTracker

Para acceder a la aplicación web, debes ingresar la URL en tu navegador:

<https://gymtracker.duckdns.org> .



⚠️ Advertencia: riesgo potencial de seguridad a continuación

Firefox ha detectado una posible amenaza de seguridad y no ha cargado **localhost**. Si visita este sitio, los atacantes podrían intentar robar información como sus contraseñas, correos electrónicos o detalles de su tarjeta de crédito.

[Más información...](#)

Retroceder (recomendado)

Avanzado...

Es posible que aparezca una advertencia de seguridad debido al certificado SSL. Verifica la URL y, si es correcta, selecciona el botón "Avanzado".

Ahora, aparecerá un segundo aviso explicando el motivo. Haz clic en "Aceptar el riesgo y continuar".

A screenshot of a certificate warning dialog box. It contains the following text:

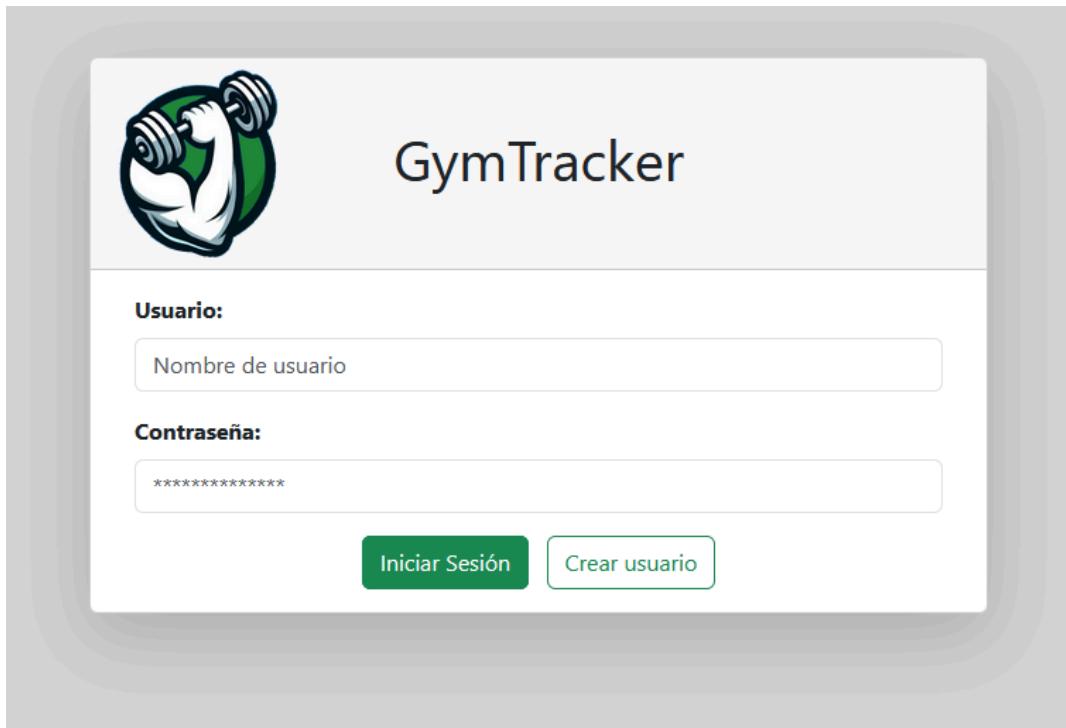
localhost usa un certificado de seguridad no válido.
No se confía en el certificado porque está autofirmado.
Código de error: [MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT](#)

[Ver certificado](#)

At the bottom are two buttons: "Retroceder (recomendado)" (Back (recommended)) in blue and "Aceptar el riesgo y continuar" (Accept the risk and continue) in yellow.

2.2.2 Inicio

2.2.2.1 Inicio Sesión



La aplicación muestra como pantalla inicial un inicio de sesión.

Tenemos dos opciones.

- En el caso de ya tener una cuenta creada, introducimos nuestro usuario y contraseña para poder acceder y pulsamos en “**Iniciar sesión**”. Pueden darse tres casos:

- El primer caso idílico sería que las credenciales sean correctas y en ese caso accederemos directamente al menú inicial del que hablaremos más tarde.
- **Error de usuario y/o contraseña**, veremos un manseje amarillo como aviso indicándonos que el usuario o contraseña que hemos introducido son incorrectos. Es decir que están mal escritos o en su defecto no existen.

Usuario o contraseña incorrectos

- **Error por usuario bloqueado**, puede darse el caso de que por distintos motivos la cuenta con la que estás intentando acceder está bloqueada, es decir que tiene restringido el acceso temporalmente por un administrador. En caso de tener algún problema no dudes en ponerte en contacto a la siguiente dirección administraciongymtracker@gmail.com , alguno de los administradores atenderá tu problema con la mayor brevedad posible.

Usuario bloqueado

2.2.2.2 Crear usuario

- Si no tienes cuenta, pulsa “Crear usuario”

Esto nos llevará a un pequeño formulario para llenar los datos de nuestro usuario.
Nombre y contraseña.

Junto a él un mensaje informativo acerca de la creación del perfil.



Crear Usuario

Nombre:

Contraseña:

Aceptar **Cancelar**

No olvides que no es necesario que completes tu perfil en este momento, pero recuerda actualizarlo más tarde. Pulsa enter para omitirlo.

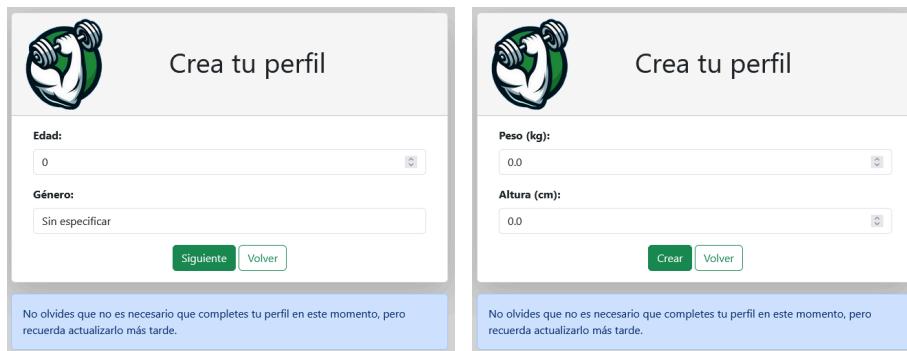
En el caso de que el nombre usuario elegido ya esté asignado a otra persona nos encontraremos un mensaje de aviso explicando la situación. La diferencia entre mayúsculas y minúsculas no es relevante. Seguirá saltando el mensaje.

Nombre de usuario ya en uso

Sigue probando nombres hasta dar con el que más te guste.
Una vez tengas nombre de usuario pasamos a la creación del perfil. Pulsando “Aceptar”. Para saltarte este paso pulsa el enter en tu teclado.

2.2.2.3 Crear Perfil

Completa la información de tu perfil, ante cualquier duda o cambio el perfil es editable



The image contains two side-by-side screenshots of a mobile application's profile creation screen. Both screens feature a green circular icon with a white figure of a person lifting weights in the top-left corner. The title "Crea tu perfil" is centered at the top of both screens.

Left Screen (Age and Gender):

- Edad:** A text input field containing "0".
- Género:** A dropdown menu showing "Sin especificar".
- Buttons:** "Siguiente" (Next) and "Volver" (Back).

Right Screen (Weight and Height):

- Peso (kg):** A text input field containing "0.0".
- Altura (cm):** A text input field containing "0.0".
- Buttons:** "Crear" (Create) and "Volver" (Back).

Note at the bottom of both screens:

No olvides que no es necesario que completes tu perfil en este momento, pero recuerda actualizarlo más tarde.

Cualquier campo relleno con información inválida no será aceptado. Se mostrará un mensaje de error.

Ante cualquier cambio se podrá retroceder en el formulario pulsando “**Volver**”.

Una vez llenado el formulario de usuario y perfil. Pulsaremos “**Crear**”.

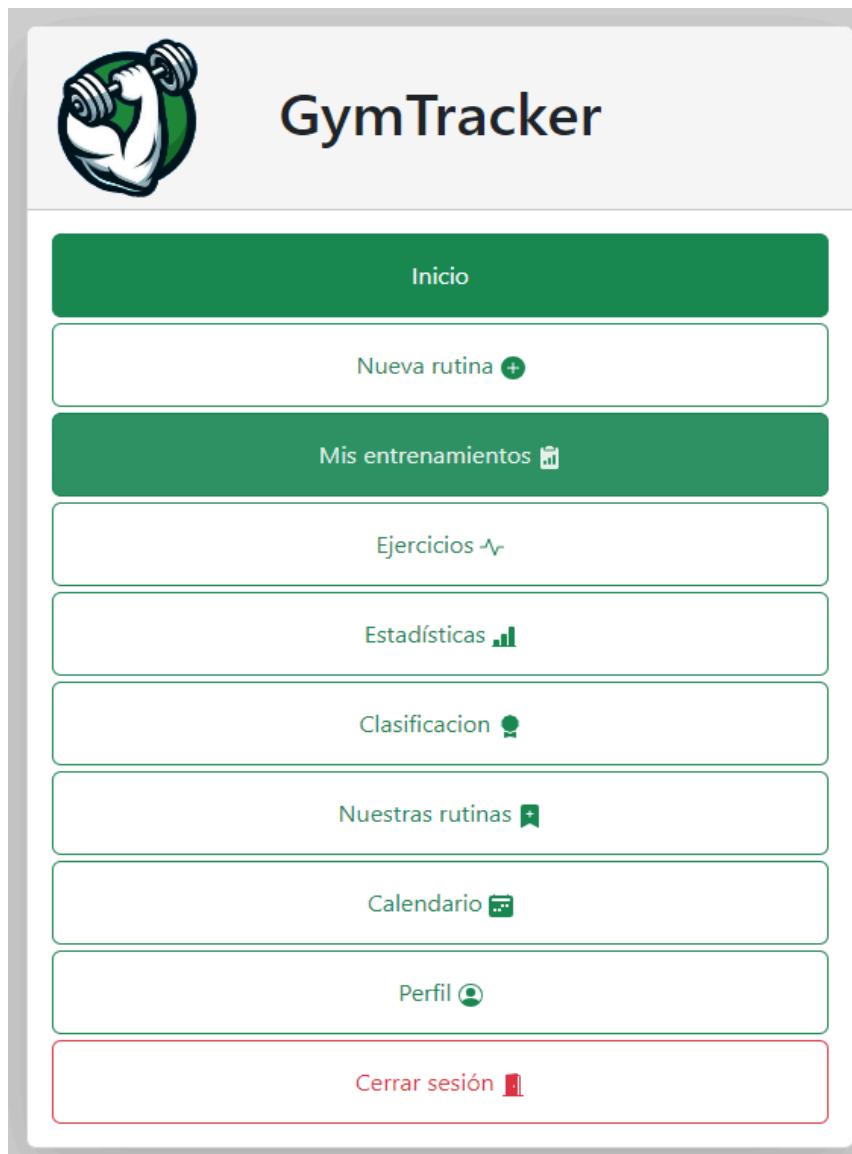
Volveremos a la pantalla inicial “**Inicio de Sesión**”, ahora puedes acceder con el nuevo usuario.

2.2.3 Menú

Una vez hemos accedido a la aplicación. Nos encontraremos con el menú inicial el cual nos indica todas las posibles acciones que podemos realizar desde la aplicación y permite navegar entre ellas. Siempre remarcado en verde la opción en la que te encuentras actualmente.

Siendo estas:

- **Inicio**
- **Nueva rutina**
- **Mis entrenamientos**
- **Ejercicios**
- **Estadísticas**
- **Clasificación**
- **Nuestras rutinas**
- **Calendario**
- **Perfil**
- **Cerrar sesión**



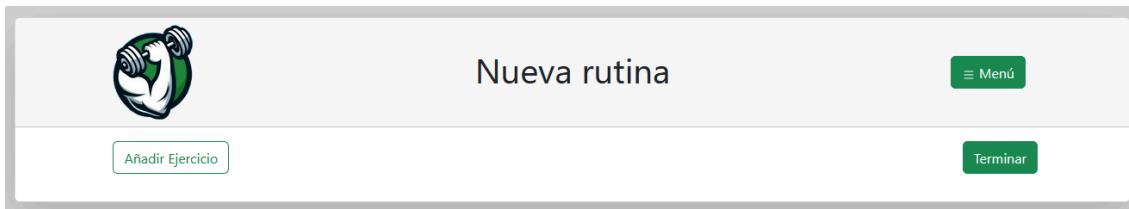
En el caso de tener alguna opción más, ponerse en contacto con la administración al correo indicado previamente (administraciongymtracker@gmail.com) para que le sean retirados los permisos. Pedimos la mayor seriedad posible en cuanto a este tema.

- **Nueva rutina**, te permitirá el acceso a la creación de una nueva rutina propia desde cero.
- **Mis entrenamientos**, mostrará un listado de todos tus entrenamientos que hayas realizado con GymTracker.
- **Ejercicios**, muestra el listado de los ejercicios de los que dispone la aplicación e información básica acerca de cada uno de ellos.
- **Estadísticas**, muestra un resumen de tus estadísticas dentro de la aplicación
- **Clasificación**, muestra una tabla con las estadísticas de todos los usuarios con acceso a la aplicación.
- **Nuestras rutinas**, un pequeño listado de rutinas predeterminadas creadas por nuestros entrenadores para un mejor entrenamiento.
- **Calendario**, un calendario para poder llevar el seguimiento de tus entrenamientos.
- **Perfil**, muestra la información de tu perfil.

- **Cerrar sesión**, cerrar y eliminar tu sesión actual, redirigiéndote a la pantalla inicial de Inicio Sesión.

2.2.4 Nueva rutina

Al acceder a la nueva rutina encontramos una pantalla prácticamente vacía para rellenarla con el entrenamiento que queremos crear.



Dispone de tres botones. “Menú”, “Añadir Ejercicio” y “Terminar”.

- **Menú**: Abrirá el menú sidebar de la aplicación
- **Añadir Ejercicio**: Muestra el conjunto de los ejercicios posibles que puedes introducir a tu rutina. Junto a varios filtros por nombre, músculo que trabaja o equipamiento necesario.
- **Terminar**: Permite finalizar la rutina.

2.2.4.1 Añadir ejercicio

Muestra el listado de ejercicios de la web

Buscar Ejercicios

[Cancelar](#)

Buscar ejercicios por nombre

Todos los equipamientos

Todos los músculos

Listado de ejercicios

Sentadillas con barra
Ejercicio para piernas y gluteos

Press de banca
Ejercicio para el pecho y tríceps

Dominadas
Ejercicio para la espalda y bíceps

Peso muerto
Ejercicio para la espalda baja, gluteos y piernas

Fondos en paralelas
Ejercicio para el pecho y tríceps

Remo con barra
Ejercicio para la espalda y bíceps

Al seleccionar cualquiera de los ejercicios listados este será añadido a nuestra rutina.

Puedes añadir tantos ejercicios como quieras e incluso varios iguales, estos solo se guardarán como uno mismo.

2.2.4.2 Añadir Serie

Junto a nuestro ejercicio se encuentran dos botones. Uno verde con un + y otro rojo con el dibujo de una papelera.

El botón verde sirve para crear una **nueva serie**.

Abrirá: una pequeña ventana para llenar la información acerca de la serie.

Una vez lleno pulsar “Aceptar” para añadir la serie.

En el caso de querer cancelar la acción pulsar “Cancelar”.

Puedes añadir tantas series como quieras.



Junto a la serie se encuentran dos botones. Uno verde-blanco con el dibujo de un lápiz y otro rojo-blanco con el dibujo de una papelera.

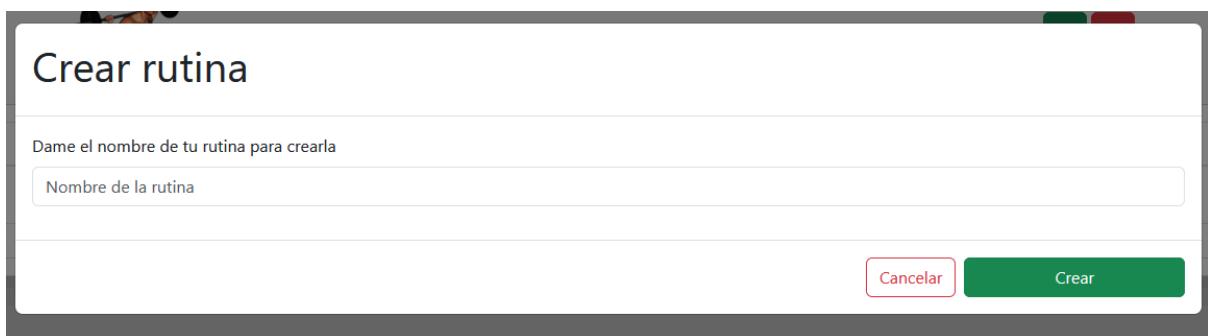
El lápiz permite **modificar los campos de la serie ya creada**.

La papelera **eliminará la serie**.

El botón rojo **eliminará el ejercicio** al completo de la rutina.

- **Terminar**

Al pulsar el botón de “Terminar”, abrirá una ventana para asignarle un nombre a nuestra rutina, este paso es obligatorio.



Con dos botones. “Cancelar” y “Crear”

- **Cancelar:** Cerrará la ventana y te permitirá seguir editando tu rutina
- **Crear:** Intentará crear la rutina. En el caso de no introducir nombre o con algún problema en la rutina (rutinas vacías...) Mostrará un mensaje de aviso explicando el problema.

El nombre de la rutina es obligatorio

Debes completar la rutina antes de terminar

Una vez creada la rutina, le redirigirá a la pantalla de “Mis entrenamientos”, entre las cuales ahora se encuentra la rutina que acaba de crear.

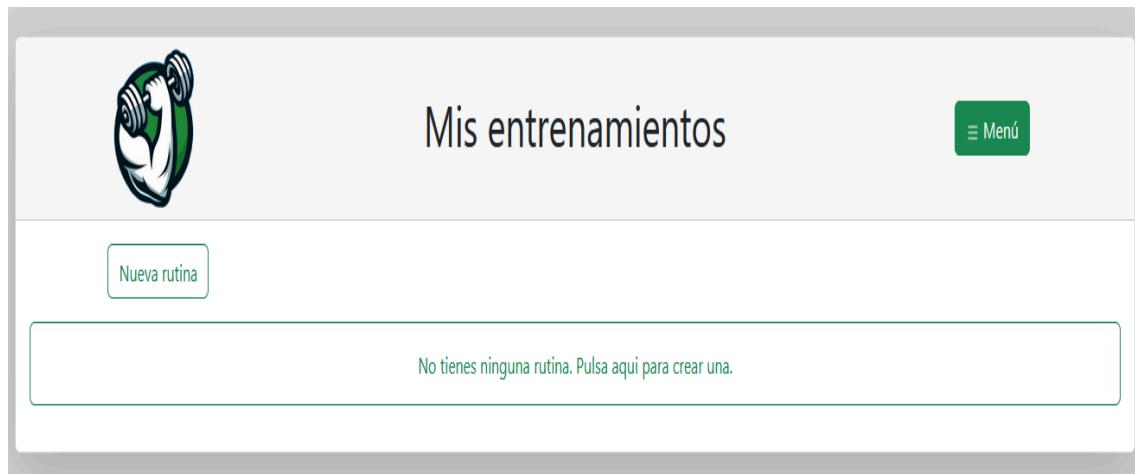
2.2.5 Mis entrenamientos

Muestra un listado de sus entrenamientos dentro de la aplicación

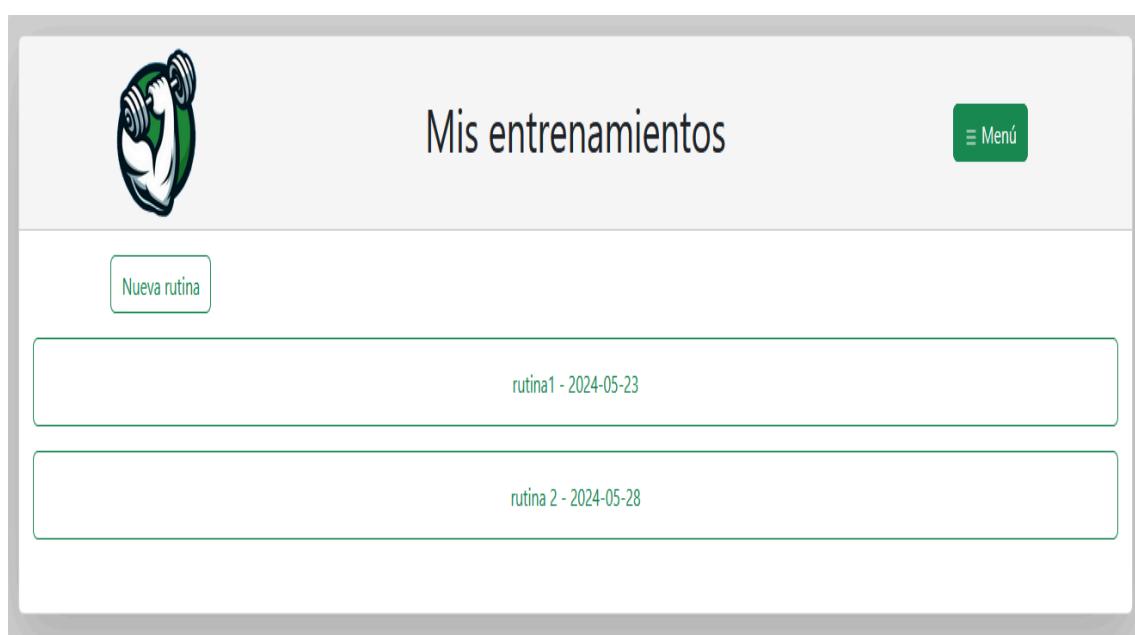
Dispone de dos botones: “Menú” y “Nueva rutina”.

- **Menú:** Abrirá el menú sidebar de la aplicación
- **Nueva rutina:** Redirige a “Nueva rutina” para crear una nueva rutina.

En el caso de no tener entrenamientos registrados en la web. Se mostrará un mensaje que le permitirá crear su primera rutina.



En caso de si tener rutinas registradas se listará en orden de creación ascendente, indicando su nombre y la fecha en la que se creó cada rutina.



Puede clickar encima de cada una de las rutinas para volver a realizarla y se le auto completarán los ejercicios y series automáticamente según los realizados en esa rutina. Puedes repetirla tantas veces como quieras.



rutina1

[☰ Menú](#)

[Añadir Ejercicio](#)

[Terminar](#)

	Sentadillas con barra	+	-
	Peso(kg)	Repeticiones	Acciones
	20	20	 
	40	1	 

	Press de banca	+	-
	Peso(kg)	Repeticiones	Acciones
	50	20	 
	100	1	 

2.2.6 Ejercicios

Dispone del botón: “Menú”.

- **Menú:** Abrirá el menú sidebar de la aplicación

Muestra el conjunto de los ejercicios posibles que puedes introducir a tu rutina. Junto a varios filtros por nombre, músculo que trabaja o equipamiento necesario.



Listado de Ejercicios

☰ Menú

Buscar ejercicios por nombre

Todos los equipamientos ▾ Todos los músculos ▾



Sentadillas con barra
Ejercicio para piernas y gluteos



Press de banca
Ejercicio para el pecho y tríceps



Dominadas
Ejercicio para la espalda y bíceps



Peso muerto
Ejercicio para la espalda baja, gluteos y piernas



Fondos en paralelas
Ejercicio para el pecho y tríceps



Remo con barra
Ejercicio para la espalda y bíceps

2.2.7 Estadísticas

La aplicación cuenta con un botón denominado “Menú” al hacer clic en este botón, se abrirá el menú lateral (sidebar) de la aplicación.

La pantalla principal muestra un resumen de tus estadísticas, incluyendo:

- Número total de entrenamientos realizados.



- Número de series completadas.



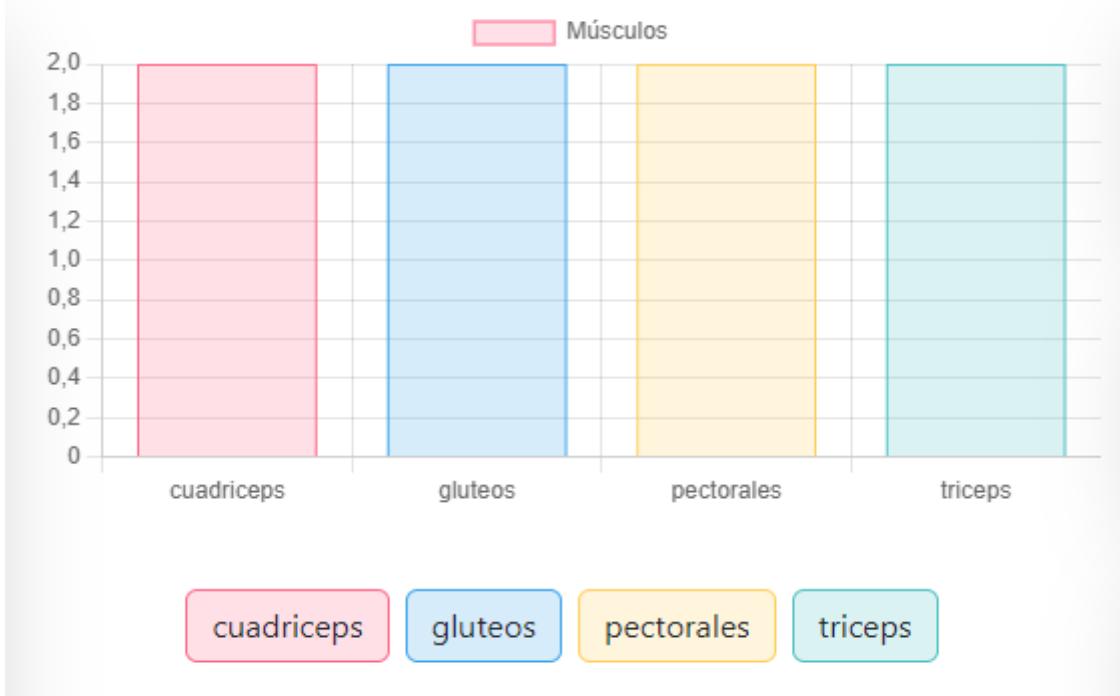
- Volumen total de peso levantado.



- Cantidad de ejercicios distintos y ejercicio favorito (que será el que hayas realizado mayor número de veces)



Además, se presenta una gráfica de barras que muestra la cantidad de ejercicios realizados por cada grupo muscular. Puedes interactuar con los distintos botones para ocultar o mostrar cada campo de la gráfica, o incluso todos los campos simultáneamente.



Pantalla completa:



2.2.8 Clasificación

Dispone de dos botones “Menú”, yo.

- **Menú:** Abrirá el menú sidebar de la aplicación
- **Yo:** Encuéntrate en la tabla.

Muestra una tabla paginada con las estadísticas de todos los usuarios con acceso a la aplicación.

Encima de la tabla se encuentra un desplegable que permite ordenarla según la estadística que prefiera.

A su izquierda se encuentra el botón YO, al pulsarlo la tabla se posicionará automáticamente en la ubicación de tu perfil.

The screenshot shows the 'Clasificación de usuarios' (User Classification) screen. At the top center is the title 'Clasificación de usuarios'. In the top right corner is a green button labeled 'Menú'. On the left side, there is a sidebar with a user icon. On the right side, there is a dropdown menu labeled 'Entrenos'. Below these, there is a table with the following columns: Nombre (Name), Entrenos (Trainers), Ejercicios (Exercises), Series (Series), and Volumen (kg) (Volume (kg)). The table contains the following data:

Nombre	Entrenos	Ejercicios	Series	Volumen (kg)
damago	11	7	14	1684.0
alvaro	0	0	0	0.0
kaka	0	0	0	0.0
w	0	0	0	0.0
david	0	0	0	0.0

At the bottom right, there are page navigation buttons: '1' (highlighted in green), '2', and '»'.

2.2.9 Nuestras rutinas

Dispone de dos botones: "Menú" y "Nueva rutina".

- **Menú:** Abrirá el menú sidebar de la aplicación
- **Nueva rutina:** Redirige a "Nueva rutina" para crear una nueva rutina.

Un pequeño listado de rutinas predeterminadas creadas por nuestros entrenadores para un mejor entrenamiento.

Nuestras rutinas

Pecho y Tríceps - 2004-01-16

Espalda y Bíceps - 1995-11-28

Piernas y Glúteos - 2004-11-02

Nueva rutina

Menú

Puede clickar encima de cada una de las rutinas para volver a realizarla y se le auto completarán los ejercicios y series automáticamente según los realizados en esa rutina. Puedes repetirla tantas veces como quieras.

rutina1

Añadir Ejercicio

Terminar

Sentadillas con barra

Press de banca

Peso(kg)	Repeticiones	Acciones
20	20	[edit] [delete]
40	1	[edit] [delete]

Peso(kg)	Repeticiones	Acciones
50	20	[edit] [delete]
100	1	[edit] [delete]

2.2.10 Calendario

Dispone del botón: "Menú".

- **Menú:** Abrirá el menú sidebar de la aplicación

Tiene acceso a un calendario para poder llevar el seguimiento de tus entrenamientos. Mostrándose en verde los días en los que se ha registrado alguna rutina y en gris la fecha actual.

En la parte superior se encuentran el **mes** y **año** del calendario. Puede navegar tanto como quiera con él.

Mayo 2024						
Lunes	Martes	Miércoles	Jueves	<th>Sábado</th> <th>Domingo</th>	Sábado	Domingo
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

2.2.11 Perfil

Dispone de tres botones: "Menú", "Editar perfil" y "Cambiar contraseña".

- **Menú:** Abrirá el menú sidebar de la aplicación.
- **Editar perfil:** Permite que modifique los campos de su perfil.
- **Cambiar contraseña:** Permite cambiar tu contraseña.

Muestra la información de tu perfil.

Nombre: damago

Edad: 25

Género: Masculino

Peso (kg): 75.0

Altura (cm): 180.0

2.2.11.1 Editar perfil

Habilita los campos para poder modificar la información de su perfil.

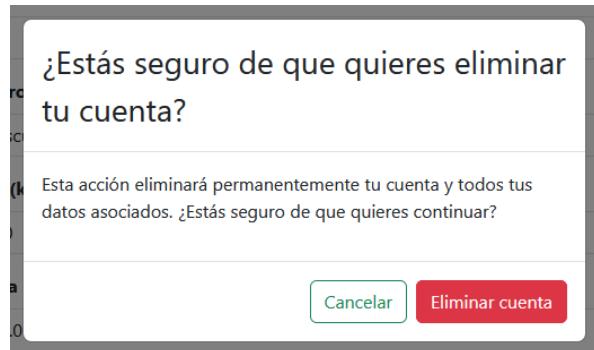
Muestra tres botones “Guardar”, “Cancelar”, “Eliminar Cuenta”, Y un mensaje informativo.

- **Guardar:** Actualiza la información introducida en el perfil. En el caso de encontrar cualquier problema se mostrará un mensaje de aviso explicando la situación. Sigue las mismas restricciones que en la creación de usuario y perfil. Una vez actualizado, volverá al resumen del perfil.

Nombre de usuario ya en uso

- **Cancelar:** No guardar los cambios introducidos y volver al resumen del perfil
- **Eliminar Cuenta:** Muestra una ventana de confirmación con dos botones, “Cancelar” y “Eliminar cuenta”
Si pulsa **Cancelar** volverá a la edición del perfil

Si pulsa **Eliminar cuenta** se borrará el usuario, perfil y toda la información



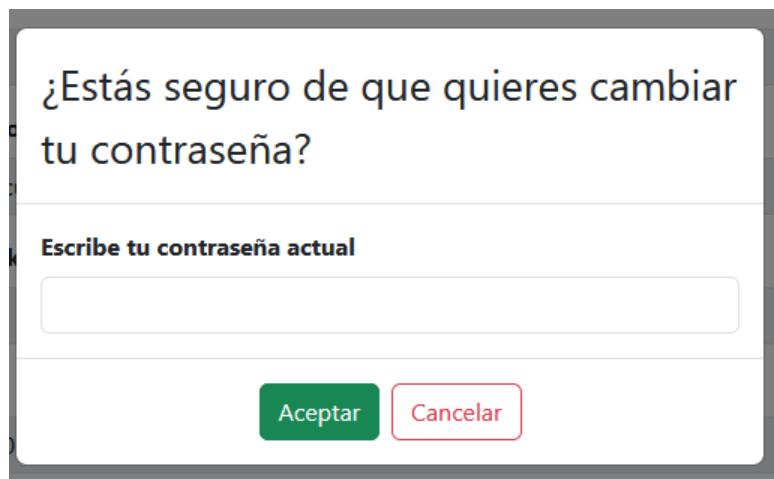
relacionada con el usuario. Se redirigirá a la pantalla de inicio “Inicio Sesión”.

2.2.11.2 Cambiar contraseña

Permite cambiar su contraseña.

- Se mostrará una ventana en la que deberá introducir su contraseña actual.

Dispondrá de dos botones “Aceptar” y “Cancelar”.



- **Aceptar:** Comprobará si la contraseña introducida es correcta.
 - Si lo es se cerrará la ventana y se abrirá una nueva para introducir la nueva contraseña.
 - Dispondrá de dos botones “Aceptar” y “Cancelar”
 - **Aceptar** se mantendrá deshabilitado hasta que la contraseña y la repetición de la contraseña sean iguales. Si clica en él se cambiará su contraseña. Seguirá en el usuario y en resumen del perfil.
 - **Cancelar**, cerrará la ventana y se mantendrá en el resumen del perfil.

Si no es correcta se mostrará un mensaje de aviso.

- **Cancelar:** Cerrará la ventana y se mantendrá en el resumen del perfil.

Crea tu nueva contraseña

Nueva contraseña

Repite la nueva contraseña

Aceptar
Cancelar

2.2.12 Menú Sidebar

Indica todas las posibles acciones que podemos realizar desde la aplicación y permite navegar entre ellas. Siempre remarcado en verde la opción en la que te encuentras actualmente.

Siendo estás:

- Inicio
- Nueva rutina
- Mis entrenamientos
- Ejercicios
- Nuestras rutinas
- Calendario
- Perfil
- Cerrar sesión

En el caso de tener alguna opción más, ponerse en contacto con la administración al correo indicado previamente (administraciongymtracker@gmail.com) para que le sean retirados los permisos. Pedimos la

mayor seriedad posible en cuanto a este tema.



2.2.13 Error

Si se ha encontrado una pantalla de error

Dispondrá de un botón arriba a la derecha “Inicio”, el cual le llevará a la pantalla principal, Menú.

Dentro de esta pantalla podrá recopilar información acerca de los sucedido.

Arriba en el centro encontrará el mensaje de error y a su izquierda el código al que pertenece este error.

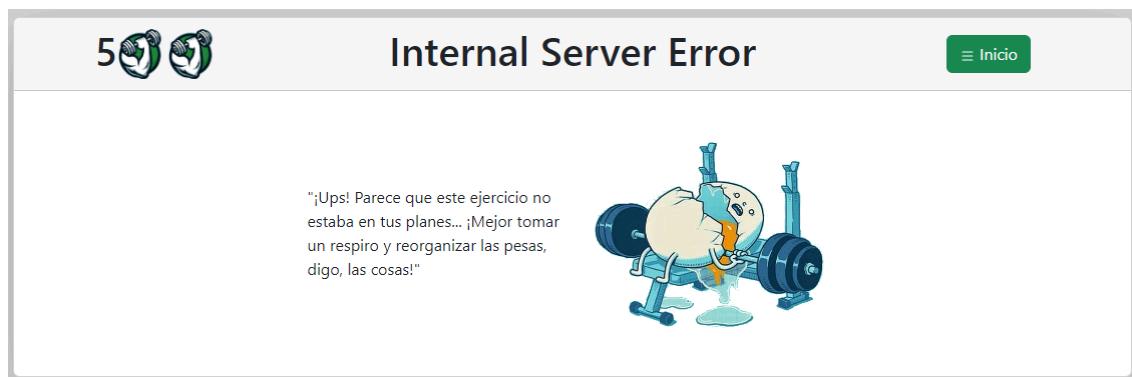
Posibles errores

404 -> la página solicitada por el usuario no se encontró en el servidor.

400 -> el servidor no puede procesar o reconocer la solicitud.

500 -> el servidor encuentra una condición inesperada que le impide cumplir con la solicitud.

Si el problema persiste no dude en ponerse en contacto con nosotros.



2.3 Manual de administrador

En el caso de los administradores la gran mayoría de funcionalidades serán las mismas que un usuario normal salvo excepciones.

El menú inicial y sidebar de los administradores mostrará un campo más en color azul “Administrar usuarios”.

The image shows two side-by-side screenshots of the GymTracker application interface. On the left, the 'User' sidebar menu is displayed, featuring a green header bar with 'Inicio' and a blue bar below it with 'Administrar usuarios'. Other items in the sidebar include 'Nueva rutina', 'Mis entrenamientos', 'Ejercicios', 'Estadísticas', 'Clasificación', 'Nuestras rutinas', 'Calendario', and 'Perfil'. The bottom item, 'Cerrar sesión', is highlighted with a red border. On the right, the 'Admin' sidebar menu is shown, which is identical to the user's except for the blue 'Administrar usuarios' bar at the top. The rest of the sidebar items ('Nueva rutina', 'Mis entrenamientos', etc.) are in green. Below the sidebar, the main content area is titled 'Inicio' and contains links for 'Administrador usuarios', 'Nueva rutina', 'Mis entrenamientos', 'Ejercicios', 'Estadísticas', 'Clasificación', 'Nuestras rutinas', 'Calendario', 'Perfil', and 'Cerrar sesión'.

2.3.1 Administrar Usuarios

Dispone de dos botones: “Menú”, “Crear usuario”..

- **Menú:** Abrirá el menú sidebar de la aplicación.
- **Crear usuarios:** Permite la creación de nuevos usuarios.

A screenshot of a table titled 'Lista de usuarios'. The table has columns: 'Id', 'Usuario', 'Rol', and 'Acciones'. There are 8 rows of data. Row 1: Id -1, Usuario -usuario_rutinas, Rol -bloqueado, Acciones -green minus, green plus, red minus, red plus. Row 2: Id 1, Usuario danago, Rol -usuario, Acciones -blue minus, blue plus, red minus, red plus. Row 3: Id 2, Usuario alvaro, Rol -administrador, Acciones -blue minus, blue plus, red minus, red plus. Row 4: Id 7, Usuario 4, Rol -bloqueado, Acciones -green minus, green plus, red minus, red plus. Row 5: Id 8, Usuario q, Rol -usuario, Acciones -blue minus, blue plus, red minus, red plus. At the top right of the table is a 'Todos los usuarios' dropdown menu.

Lista de usuarios			
Menú			
Crear usuario			
Id	Usuario	Rol	Acciones
-1	usuario_rutinas	bloqueado	
1	danago	usuario	
2	alvaro	administrador	
7	4	bloqueado	
8	q	usuario	

Muestra una tabla paginada de todos los usuarios de la aplicación

La tabla puede ser filtrada por tipo de rol de los usuarios “Administrador”, “Usuario”, “Bloqueado” o “Todos los usuarios”.

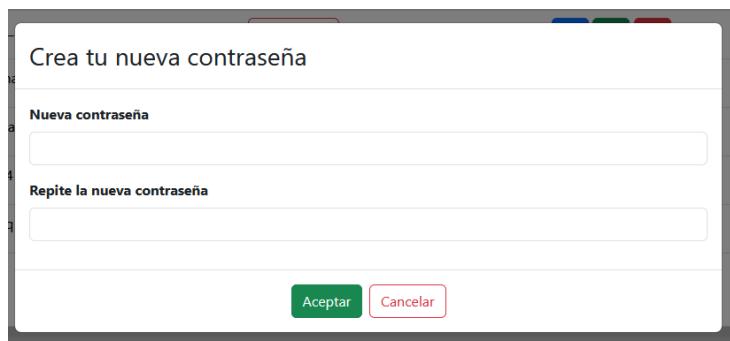
Cada fila de la tabla muestra un usuario y junto con él su id, nombre de usuario, rol y tres botones de acción: Uno azul con una llave, uno verde con una puerta u otro rojo con una papelera.

2.3.1.1 Cambiar contraseña del usuario



Llave.

- Permite cambiar la contraseña del usuario sin necesidad de saber la que tenía previamente
 - Dispondrá de dos botones “Aceptar” y “Cancelar”
 - **Aceptar** se mantendrá deshabilitado hasta que la contraseña y la repetición de la contraseña sean iguales. Si clica en él se cambiará su contraseña.
 - **Cancelar**, cerrará la ventana, sin cambios



2.3.1.2 Acceder a la cuenta del usuario



Puerta.

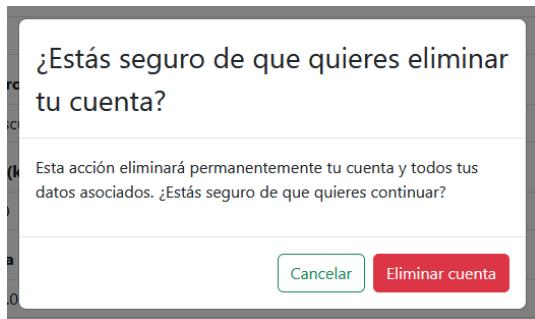
- Dará acceso a la cuenta del usuario seleccionado

2.3.1.3 Eliminar usuario



Papelera.

- Muestra una ventana de confirmación con dos botones, “Cancelar” y “Eliminar cuenta”
 - Dispondrá de dos botones “Cancelar” y “Eliminar cuenta”
 - Si pulsa **Cancelar** volverá a la edición del perfil
 - Si pulsa **Eliminar cuenta** se borrará el usuario, perfil y toda la información relacionada con el usuario.



En el campo rol de cada usuario al pulsarlo se le abrirá una ventana con los posibles roles que se le pueden asignar al usuario.

Rol

usuario

Cambiar rol

administrador usuario bloqueado

Volver

De forma adicional puede encontrarse símbolo gris de una persona, indicativo del perfil en que está el administrador en este momento.



1

damago

administrador

key lock trash can

person icon

2.3.1.4 Crear usuario

- Permite crear un usuario siguiendo las mismas condiciones que la creación de usuario en la pantalla inicial. Con la diferencia que los administradores pueden elegir el rol que tendrá el usuario a crear.
- Dispondrá de dos botones “Aceptar” y “Cancelar”

Crear Usuario

Nombre:
Nombre de usuario

Contraseña:

Rol usuario:
Usuario

Nombre de usuario ya en uso

Cancelar Aceptar

- **Cancelar**, cerrará la ventana, el usuario no será creado.

3. Adecuación del entorno de trabajo

Las instalaciones necesarias para llevar a cabo el desarrollo de GymTracker son mínimas, principalmente el IDE de trabajo (STS en Linux e IntelliJ en Windows), MySQL y la clonación del repositorio remoto de GitHub.

A continuación se detallan las configuraciones necesarias para ambos entornos:

3.1 Instalación de MySQL:

- Instalar MySQL Server tanto en linux, como en windows según las instrucciones proporcionadas en la documentación oficial de MySQL: [MySQL Community Downloads](#).
- O en Linux simplemente con los comando:
 - **sudo apt update**
 - **sudo apt install mysql-server**

3.2 Instalación de Git:

- En linux:
 - **sudo apt update** (actualizar paquetes)
 - **sudo apt install git** (instalar git)
 - **git --version** (verificar instalación)
- En windows:
 - Descarga la última versión del instalador de Git desde el sitio oficial: [Git for Windows](#).
 - Sigue las instrucciones del instalador. Durante la instalación, puedes elegir las opciones predeterminadas o personalizar la instalación según tus preferencias.
 - Verificar la instalación: **git --version**

3.3 Clonación del Repositorio:

- Clonar el repositorio de GymTracker desde GitHub:
 - **git clone https://github.com/deiivvv/GymTracker.git**

3.4 Instalación de JDK:

Para el desarrollo de GymTracker, necesitaremos una JDK de Java versión 17 o superior. A continuación se detallan los pasos para instalar JDK en Linux y Windows.

3.4.1 Instalación de JDK en Linux

- Actualizar el índice de paquetes:
 - **sudo apt update**
- Instalar OpenJDK 17:
 - **sudo apt install openjdk-17-jdk**
- Verificar la instalación:
 - **java -version**

Deberías ver una salida que indique que Java 17 está instalado.

3.4.2 Instalación de JDK en Windows

- Descargar el JDK de Oracle: [Oracle JDK 17 Downloads](#)
- Ejecutar el instalador: sigue las instrucciones del instalador para instalar el JDK.

4.4.2.1 Configurar la variable de entorno JAVA_HOME:

- Abre el Panel de Control y ve a "Sistema y Seguridad" > "Sistema" > "Configuración avanzada del sistema".
- En la pestaña "Avanzado", haz clic en "Variables de entorno".
- En "Variables del sistema", haz clic en "Nueva" e introduce:
 - Nombre de la variable: **JAVA_HOME**
 - Valor de la variable: **Ruta al directorio de instalación del JDK** (por ejemplo, C:\Program Files\Java\jdk-17)
- Busca la variable Path en "Variables del sistema" y haz clic en "Editar". Añade una nueva entrada con la ruta **%JAVA_HOME%\bin**.
- Verificar la instalación:
 - **java -version**

3.5 Instalación de Maven:

3.5.1 Instalación de Maven en Linux

- Actualizar el índice de paquetes:
 - **sudo apt update**
- Instalar Maven:
 - **sudo apt install maven**
- Verificar la instalación:
 - **mvn -version**

Deberías ver una salida que indique la versión de Maven instalada.

3.5.2 Instalación de Maven en Windows

- Descarga la última versión binaria zip de Maven desde el sitio oficial: Apache Maven Downloads.

- Extrae el contenido del archivo zip descargado en una ubicación de preferencia (por ejemplo, C:\Program Files\Maven).

3.5.2.1 Configurar la variable de entorno MAVEN_HOME:

- Abre el Panel de Control y ve a "**Sistema y Seguridad**" > "**Sistema**" > "**Configuración avanzada del sistema**".
- En la pestaña "**Avanzado**", haz clic en "**Variables de entorno**".
- En "**Variables del sistema**", haz clic en "**Nueva**" e introduce:
 - Nombre de la variable: **MAVEN_HOME**
 - Valor de la variable: **Ruta al directorio de Maven** (por ejemplo, C:\Program Files\Maven\apache-maven-3.8.6).
- Busca la variable Path en "**Variables del sistema**" y haz clic en "**Editar**". Añade una nueva entrada con la ruta **%MAVEN_HOME%\bin**.
- Verificar la instalación:
 - Abre una nueva ventana de Command Prompt (CMD) y ejecuta:
 - **cmd**
 - **mvn -version**

Deberías ver una salida que indique la versión de Maven instalada.

3.6 Configuración del IDE (STS - Spring Tool Suite)

3.6.1 Instalación de Spring Tool Suite:

- Descargar e instalar Spring Tool Suite (STS) desde el sitio oficial: [Spring Tool Suite](#).
- Descomprimir el archivo descargado en una ubicación de preferencia en tu sistema.

3.6.2 Configuración de Lombok:

- Descargar el archivo JAR de Lombok desde su sitio oficial: [Project Lombok](#).
- Ejecutar el archivo JAR descargado: java -jar lombok.jar.
- Se abrirá un instalador, sigue las instrucciones para integrar Lombok con Spring Tool Suite.

3.6.3 Importar Proyectos con STS:

Para importar los proyectos GymTracker-AppWeb y Api-Ejercicios en Spring Tool Suite (STS), sigue estos pasos:

1. Abre Spring Tool Suite.
2. Ve a "**File**" (Archivo) en la barra de menú.
3. Selecciona "**Import**" (Importar) en el menú desplegable.
4. En el cuadro de diálogo de importación, selecciona "**Maven**" > "**Existing Maven Projects**" (Proyectos Maven Existentes).
5. Haz clic en "**Next**" (Siguiente).
6. En el campo "**Root Directory**" (Directorio Raíz), navega hasta la carpeta clonada del repositorio de Git donde se encuentra el proyecto GymTracker.
7. Dentro de la carpeta clonada, busca y selecciona la carpeta "**GymTracker-AppWeb**" y luego repite el proceso para "**Api-Ejercicios**".
8. Haz clic en "**Finish**" (Finalizar) para importar el proyecto en Spring Tool Suite.

3.7 Configuración del IDE (IntelliJ IDEA)

3.7.1 Instalación de IntelliJ IDEA

- Ve al sitio oficial de JetBrains y descarga IntelliJ IDEA: [JetBrains IntelliJ IDEA](#).

3.7.2 Importar Proyectos con IntelliJ IDEA:

Para importar los proyectos GymTracker-AppWeb y Api-Ejercicios en IntelliJ IDEA, sigue estos pasos:

1. Abre IntelliJ IDEA.
2. En la pantalla de bienvenida, selecciona "Open" (Abrir).
3. Navega hasta la carpeta clonada del repositorio de Git donde se encuentra el proyecto GymTracker.
4. Selecciona la carpeta "GymTracker-AppWeb" y haz clic en "OK" para abrir el proyecto.
5. Espera a que IntelliJ IDEA indexe el proyecto y descargue las dependencias necesarias.
6. Para importar otro proyecto, repite los pasos:
 - Ve a "File" (Archivo) en la barra de menú.
 - Selecciona "New" > "Project from Existing Sources..." (Nuevo > Proyecto desde fuentes existentes).
 - Navega hasta la carpeta clonada del repositorio de Git y selecciona la carpeta "Api-Ejercicios".
 - Haz clic en "OK" para abrir el proyecto.
7. IntelliJ IDEA te pedirá que configures el proyecto como un proyecto Maven. Acepta las configuraciones predeterminadas y haz clic en "Next" (Siguiente) y luego en "Finish" (Finalizar).

4. Arquitectura del proyecto

El proyecto GymTracker se divide en tres partes principales para asegurar un mayor control y aislamiento de cada componente, lo que facilita su gestión, mantenimiento y evolución. Estas partes son:

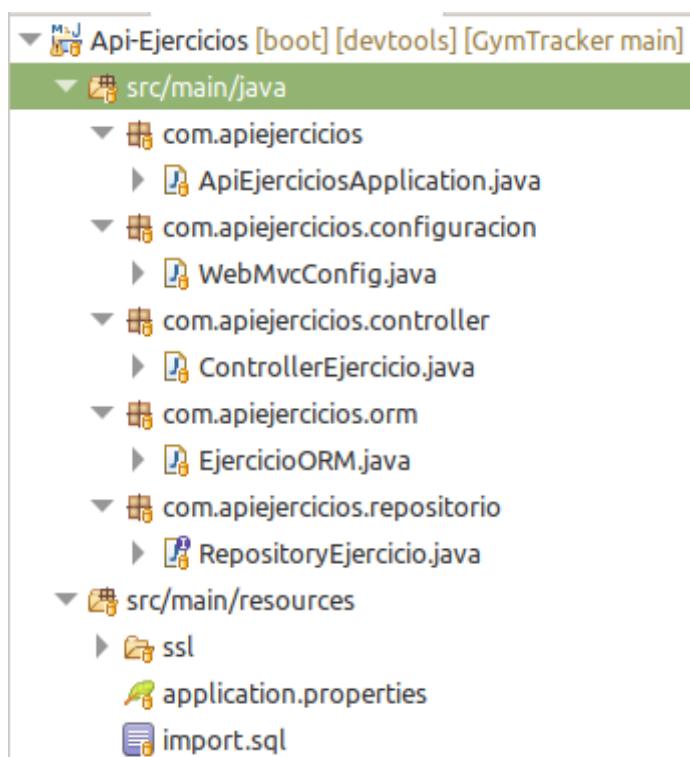
- **Base de datos**
- **Api de ejercicios**
- **App de GymTracker**

Esta arquitectura modular no solo simplifica el desarrollo y la implementación de nuevas funcionalidades, sino que también promueve una mayor reutilización del código y una mejor organización del proyecto.

A continuación la explicación de la arquitectura de la api de ejercicios y la app de GymTracker:

4.1 Arquitectura de la API

Nuestra API de ejercicios está organizada de manera estructurada para facilitar su desarrollo y mantenimiento. A continuación, explicamos la arquitectura y los componentes principales de nuestro proyecto:



4.1.1 Estructura y Funcionalidad

1. **ApiEjerciciosApplication.java:**
 - Es el punto de entrada de nuestra aplicación Spring Boot. Aquí se configura y se inicia la aplicación.
2. **Configuración (configuracion/WebMvcConfig.java):**
 - En este paquete y archivo configuramos las políticas de acceso a la API. Aquí determinamos quién puede acceder a los distintos endpoints, configuramos CORS (Cross-Origin Resource Sharing) y otras configuraciones de seguridad y acceso.
3. **Controlador (controller/ControllerEjercicio.java):**
 - Este controlador se encarga de manejar las solicitudes HTTP. Define los endpoints de la API y utiliza el repositorio para consultar y devolver los datos. En nuestro caso, solo permitimos solicitudes GET para mantener la API en un estado de solo lectura.
4. **ORM (orm/EjercicioORM.java):**
 - Utilizamos Spring Data JPA para definir nuestras entidades de base de datos con anotaciones como @Id y @Entity. Esta clase representa la estructura de la tabla de ejercicios en la base de datos H2 y facilita la manipulación y consulta de los datos.
5. **Repositorio (repositorio/RepositoryEjercicio.java):**
 - Aquí definimos el repositorio que interactúa con la base de datos H2. Spring Data JPA nos simplifica la consulta y filtrado de datos predefinidos de ejercicios. Este repositorio es utilizado por el controlador para obtener los datos necesarios para las respuestas de la API.

4.1.2 Recursos

1. **application.properties:**
 - Contiene las configuraciones de la aplicación, incluyendo detalles de la conexión a la base de datos H2, configuraciones de servidor y otros parámetros importantes.

2. **import.sql**:
 - Este archivo se utiliza para importar datos predefinidos a la base de datos H2 al inicio de la aplicación. Contiene los ejercicios que la API mostrará.
3. **SSL (ssl/certificate.pem, certificate-request.csr, keystore.p12, private-key.pem)**:
 - Aquí almacenamos los certificados SSL y las llaves privadas necesarias para asegurar nuestra API. Estos archivos aseguran que las comunicaciones con la API sean seguras y encriptadas con https.

4.1.3 Tecnologías Utilizadas

- **Spring Data JPA**: Simplifica la manipulación y consulta de datos en la base de datos.
- **H2 Database**: Utilizamos esta base de datos en memoria porque nuestra API solo necesita mostrar datos predefinidos y no permite modificaciones.
- **Spring Boot**: Facilita la creación de aplicaciones stand-alone, de producción listas para ejecutarse.
- **Spring Boot DevTools**: Facilita el desarrollo y la depuración al proporcionar herramientas para la recarga automática de cambios en la aplicación durante el desarrollo.
- **Spring Web**: Proporciona funcionalidades para el desarrollo de aplicaciones web, como la creación de controladores RESTful y la gestión de solicitudes HTTP.

En resumen, hemos diseñado esta API para que sea fácil de mantener y de escalar, con una arquitectura clara y tecnologías que aseguran un rendimiento óptimo y seguridad en el acceso a los datos.

4.2 Arquitectura de la APP

Nuestra aplicación GymTracker está diseñada siguiendo una arquitectura MVC (Modelo-Vista-Controlador) para asegurar una organización clara y una separación de responsabilidades eficiente. A continuación, detallamos la estructura y los componentes principales de nuestro proyecto:



4.2.1. Configuración

En la carpeta **Configuracion**, gestionamos la configuración de la aplicación. Incluye archivos como:

Configuración para poder pillar la conexión a la base de datos con variables de entorno como se hizo para Docker:

- **AppConfig.java**: Configuración general de la aplicación.
- **DataSourceConfig.java**: Configuración de la fuente de datos para la base de datos.

Configuración de seguridad:

- **SecurityConfig.java**: Configuración de seguridad utilizando Spring Security.

4.2.2. Controladores (Controllers)

Los controladores en la carpeta **Controller** gestionan las solicitudes HTTP y dirigen las respuestas adecuadas. Algunos de los controladores son:

- **AdminController.java**: Gestión de la interfaz y funcionalidades del administrador.
- **CalendarioController.java**: Gestión del calendario de entrenamientos.
- **EjerciciosController.java**: Gestión de los ejercicios disponibles.
- **EstadisticasController.java**: Gestión de las estadísticas de los usuarios.
- **LoginController.java**: Gestión del inicio de sesión de los usuarios.

4.2.3. Data Transfer Objects (DTO)

En la carpeta **Dto**, definimos objetos que se utilizan para transferir datos entre la capa de servicio y los controladores. Algunos ejemplos son:

- **EstadisticasUsuarioDTO.java**
- **MisEntrenamientosDTO.java**
- **UsuarioDTO.java**

4.2.4. Servicios (Services)

La carpeta **Service** contiene la lógica de negocio de la aplicación. Aquí es donde interactuamos con la base de datos a través de nuestros servicios. Algunos servicios incluyen:

- **CalendarioService.java**
- **ClasificacionService.java**
- **UsuarioService.java**

4.2.5. Recursos (Resources)

La carpeta **resources** contiene archivos de configuración, recursos estáticos y plantillas Thymeleaf. Incluye:

- **application.properties**: Configuración general de la aplicación.
- **static**: Archivos CSS, imágenes y JavaScript.
- **templates**: Plantillas HTML utilizadas por Thymeleaf.

4.2.6 Dependencias

Utilizamos varias dependencias para desarrollar la aplicación:

- **Spring Boot DevTools**: Facilita el desarrollo y la depuración.
- **Spring Web**: Proporciona funcionalidades para desarrollar aplicaciones web.
- **JDBC API**: Permite el acceso a bases de datos relacionales.
- **Spring Data JPA**: Simplifica el acceso y manipulación de datos.
- **MySQL Driver**: Proporciona conectividad JDBC para MySQL.
- **Lombok**: Simplifica el desarrollo eliminando código boilerplate.
- **Spring Session**: Gestiona las sesiones de usuario.
- **Spring Security**: Proporciona una capa de seguridad robusta.
- **Thymeleaf**: Motor de plantillas para generar vistas HTML dinámicas.

4.2.7 Mapeo con ORM

En el archivo **orm.xml**, se realiza el mapeo de los datos recuperados desde la base de datos para formar objetos del tipo de DTO en los servicios. Este archivo configura cómo los datos deben ser transformados y utilizados dentro de la aplicación, asegurando que los datos se estructuren correctamente al ser transferidos entre la base de datos y los objetos Java.

4.2.8 Flujo de Datos

En nuestra arquitectura MVC, los datos fluyen desde la base de datos a través de los servicios, donde se mapean a DTOs (Data Transfer Objects). Los controladores reciben estos DTOs y los pasan a las vistas. Utilizamos Thymeleaf para integrar fácilmente los datos del servidor en la interfaz de usuario.

Además, los controladores también responden a peticiones Axios para recuperar datos concretos y hacer peticiones a la API de ejercicios, permitiendo una comunicación y una experiencia de usuario más dinámica.

4.2.9 Gestión de Seguridad y Sesiones

- **Spring Security**: Gestiona la autenticación y autorización de usuarios, protegiendo los datos y recursos.

- **Spring Session:** Permite gestionar los roles de usuarios y otras características de la sesión, asegurando un control adecuado sobre los accesos y acciones de los usuarios.

4.2.10 Uso de Lombok

Lombok nos ayuda a reducir el código repetitivo y simplificar la sintaxis en la creación de DTOs. Esto facilita la inyección de dependencias y la generación automática de constructores y otros métodos.

4.2.11 Conclusión

Nuestra aplicación GymTracker está cuidadosamente estructurada para facilitar el desarrollo y el mantenimiento, asegurando una separación clara de responsabilidades y una interacción fluida entre las diferentes capas de la aplicación. Utilizamos una combinación de tecnologías y dependencias robustas para proporcionar una experiencia de usuario segura y eficiente.

5. Desarrollo de la base de datos

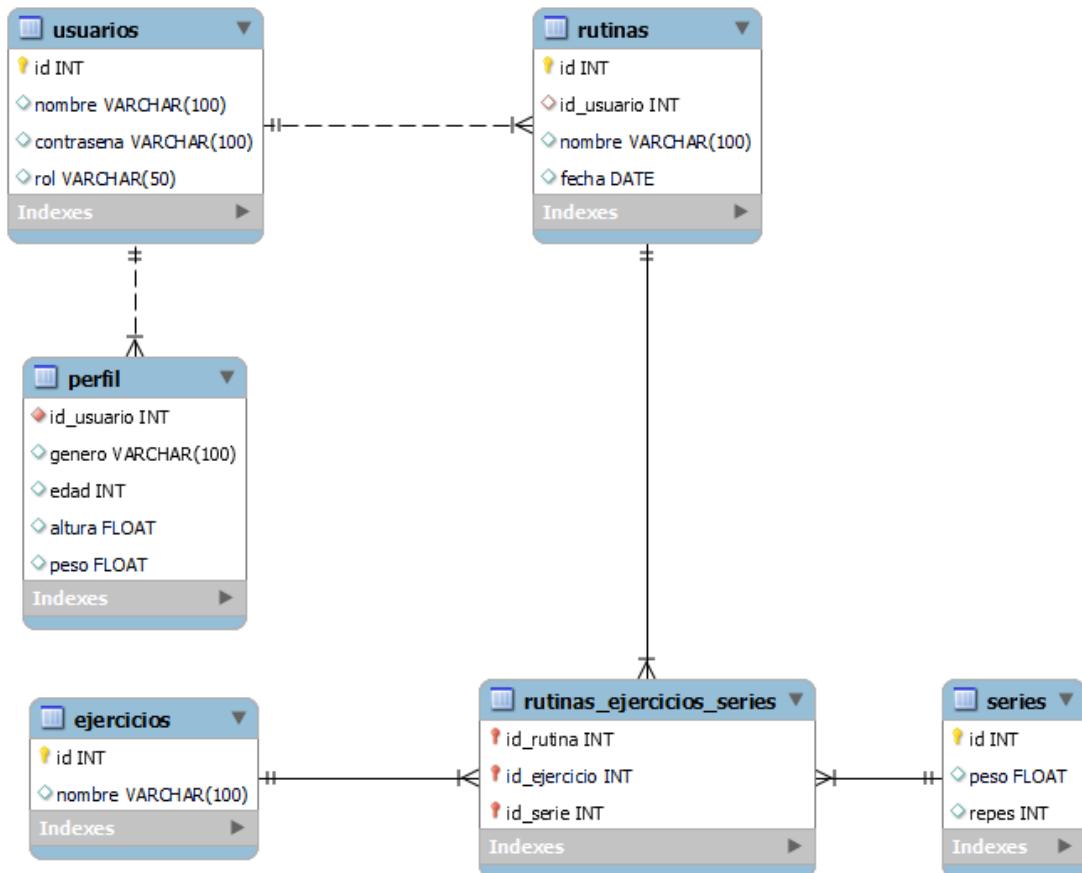
En este punto se documenta todo sobre la base de datos de GymTracker, la cuál se compone de las siguientes 6 tablas:*usuarios, perfil, rutinas, rutinas_ejercicios_series, ejercicios, series*.

```
mysql> use gymtracker;
Database changed
mysql> show tables;
+-----+
| Tables_in_gymtracker |
+-----+
| ejercicios           |
| perfil               |
| rutinas              |
| rutinas_ejercicios_series |
| series               |
| usuarios             |
+-----+
6 rows in set (0.01 sec)
```

A continuación se explica la base de datos más a fondo:

5.1 Arquitectura de la base de datos

5.1.1 Esquema de la base de datos:



5.1.2 Tablas Principales:

- Usuarios:** Almacena información sobre los usuarios del sistema, como su nombre, contraseña y rol.
- Perfil:** Contiene detalles adicionales sobre los usuarios, como género, edad, altura y peso. La clave externa se relaciona con la tabla "Usuarios".
- Rutinas:** Registra las rutinas de entrenamiento, incluyendo el nombre, la fecha y la clave externa que se relaciona con la tabla "Usuarios".
- Ejercicios:** Almacena los diferentes ejercicios disponibles.
- Series:** Guarda información sobre las series de ejercicios, como el peso y el número de repeticiones.

5.1.3 Relación entre Rutinas, Ejercicios y Series:

- La relación entre las tablas "Rutinas", "Ejercicios" y "Series" se establece a través de la tabla intermedia "rutinas_ejercicios_series". Esta tabla actúa como un puente entre las rutinas, los ejercicios y las series.
- Cada fila en "rutinas_ejercicios_series" representa una combinación específica de rutina, ejercicio y serie.

5.1.4 Relaciones:

- **Usuarios → Perfil:** Relación 1:1 (un usuario tiene un perfil).
- **Usuarios → Rutinas:** Relación 1:m (un usuario puede tener varias rutinas).
- **Rutinas → Ejercicios:** Relación m:m (una rutina puede incluir varios ejercicios y un ejercicio puede estar en varias rutinas).
- **Rutinas → Series:** Relación m:m (una rutina puede tener múltiples series y una serie puede estar en varias rutinas).

5.1.5 Claves Externas y Borrados en Cascada:

- **Clave Externa en “Rutinas” (id_usuario):** Esta clave se relaciona con la tabla “Usuarios”. Si se elimina un usuario, todas sus rutinas asociadas también se eliminarán automáticamente (borrado en cascada).
- **Clave Externa en “Rutinas_ejercicios_series” (id_rutina):** Se relaciona con la tabla “Rutinas”. Si se elimina una rutina, todas las filas asociadas en “rutinas_ejercicios_series” también se eliminarán (borrado en cascada).
- **Clave Externa en “Rutinas_ejercicios_series” (id_ejercicio):** Se relaciona con la tabla “Ejercicios”. Si se elimina un ejercicio, las filas correspondientes en “rutinas_ejercicios_series” también se eliminarán (borrado en cascada).
- **Clave Externa en “Rutinas_ejercicios_series” (id_serie):** Se relaciona con la tabla “Series”. Si se elimina una serie, las filas asociadas en “rutinas_ejercicios_series” también se eliminarán (borrado en cascada).

En resumen, la base de datos permite gestionar usuarios, sus perfiles, las rutinas de entrenamiento que crean, los ejercicios asociados a alguna rutina y las series asociadas a cada ejercicio en una rutina específica. Las relaciones entre las tablas están diseñadas para mantener la integridad referencial y facilitar la gestión de datos.

5.2 Explicación y justificación de la base de datos

La elección y diseño de la base de datos para GymTracker se basó en el objetivo principal de proporcionar a cada usuario su perfil personalizado y la capacidad de gestionar sus rutinas de entrenamiento, junto con consideraciones clave sobre integridad, escalabilidad y eficiencia en consultas.

En esta sección, profundizaremos en la justificación de la elección de esta estructura de base de datos para el sistema GymTracker. Aquí hay algunas consideraciones importantes:

5.2.1 Integridad de los Datos:

- La relación entre usuarios, rutinas, ejercicios y series se establece de manera coherente mediante claves externas. Esto garantiza que los datos estén bien estructurados y que no haya inconsistencias.

- El borrado en cascada asegura que, al eliminar un usuario, todas las entidades relacionadas se eliminan automáticamente sin dejar registros huérfanos.

5.2.2 Escalabilidad:

- El diseño m:m entre rutinas y ejercicios permite flexibilidad. Los usuarios pueden crear rutinas personalizadas con cualquier combinación de ejercicios.
- La tabla intermedia “rutinas_ejercicios_series” facilita la expansión futura. Si se agregan más ejercicios o se modifican las series, la estructura sigue siendo válida.

5.2.3 Eficiencia en Consultas:

- Las relaciones 1:m entre usuarios y rutinas, así como entre rutinas y ejercicios, permiten consultas eficientes. Por ejemplo, buscar todas las rutinas de un usuario específico es rápido.
- Las claves externas optimizan las búsquedas y las uniones entre tablas.

5.2.4 Consistencia y Mantenibilidad:

- La estructura bien definida facilita el mantenimiento y la corrección de errores.
- Las relaciones claras ayudan a los desarrolladores a comprender la lógica de la base de datos.

En resumen, la base de datos “GymTracker” se diseñó cuidadosamente para satisfacer las necesidades de los usuarios, garantizar la integridad de los datos y permitir futuras expansiones

6. Desarrollo del proyecto

En este apartado se detalla el proceso de construcción de nuestra aplicación GymTracker. Comenzamos por establecer la base de datos y desarrollar la API de ejercicios. La idea central consiste en que los ejercicios se almacenen en la API, mientras que en la base de datos solo se guardan aquellos que pertenecen a una rutina específica dentro de GymTracker. A continuación, se describe el desarrollo de la API de ejercicios y la aplicación de GymTracker:

6.1 Desarrollo de la API

6.1.1 Configuración CORS para el acceso a la API

```
@Configuration
public class WebMvcConfig implements WebMvcConfigurer {

    @Override
    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping("/**") // Permitir acceso a todos los endpoints
            .allowedOrigins("https://localhost",
                            "http://localhost:8080",
                            "http://gymtracker.duckdns.org:8080",
                            "http://gymtracker.duckdns.org",
                            "https://gymtracker.duckdns.org") // Permitir solicitudes desde estos orígenes
            .allowedMethods("GET", "POST", "PUT", "DELETE") // Permitir estos métodos HTTP
            .allowCredentials(true) // Permitir credenciales (si se están usando)
            .maxAge(3600); // Tiempo de vida del preflight request en segundos
    }
}
```

WebMvcConfig es una configuración específica para Spring Framework en una aplicación web. Explicación:

1. **@Configuration:** Esta anotación marca la clase como una configuración de Spring. Indica que contiene definiciones de beans o configuraciones personalizadas.
2. **WebMvcConfigurer:** Esta interfaz proporciona métodos para personalizar la configuración de Spring MVC (Model-View-Controller). En este caso, estamos implementando esta interfaz para personalizar la configuración CORS (Cross-Origin Resource Sharing).
3. **addCorsMappings(CorsRegistry registry):** Este método se llama durante la inicialización de la aplicación y configura las políticas CORS. Aquí están los detalles:
 - **registry.addMapping("/**"):** Define un mapeo global para todos los endpoints de la API.
 - **.allowedOrigins(...):** Especifica los orígenes permitidos para las solicitudes CORS. En este caso, se permiten solicitudes desde varios dominios (localhost y gymtracker.duckdns.org).
 - **.allowedMethods(...):** Define los métodos HTTP permitidos para las solicitudes CORS (GET, POST, PUT, DELETE).
 - **.allowCredentials(true):** Permite el envío de credenciales (como cookies o encabezados de autenticación) en las solicitudes CORS.
 - **.maxAge(3600):** Establece el tiempo de vida (en segundos) del preflight request (una solicitud de verificación previa) para evitar solicitudes CORS frecuentes.

6.2 Desarrollo de la APP

7. Despliegue

En este apartado se abordará el proceso de despliegue de la aplicación web, Gymtracker, utilizando diversas plataformas y herramientas para asegurar su disponibilidad y correcto funcionamiento en diferentes entornos. En primer lugar, se detallará el despliegue en Amazon Web Services (AWS), una de las plataformas en la nube más robustas y ampliamente utilizadas. A continuación, se describirá el uso de Docker para la creación de contenedores que faciliten la portabilidad y escalabilidad de la aplicación. También se incluirá la implementación a través de Spring Tools Suite (STS), un entorno de desarrollo integrado especializado en aplicaciones Spring, y finalmente, se explicará el despliegue local, proporcionando una guía para ejecutar la aplicación en un entorno de desarrollo personal. Cada método será presentado con sus respectivas configuraciones, ventajas y consideraciones clave, permitiendo una visión completa y detallada de las opciones de despliegue disponibles.

7.1 Despliegue en Aws

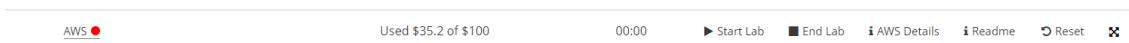
7.1.1 Lanzar instancia

Para el despliegue de en una instancia de aws

Accedemos a la cuenta de aws en nuestro caso será **awsacademy**

<https://awsacademy.instructure.com/>

Una vez dentro encendemos el laboratorio de aws



StarLab y esperamos a que el indicador de AWS cambie de rojo/amarillo a verde 

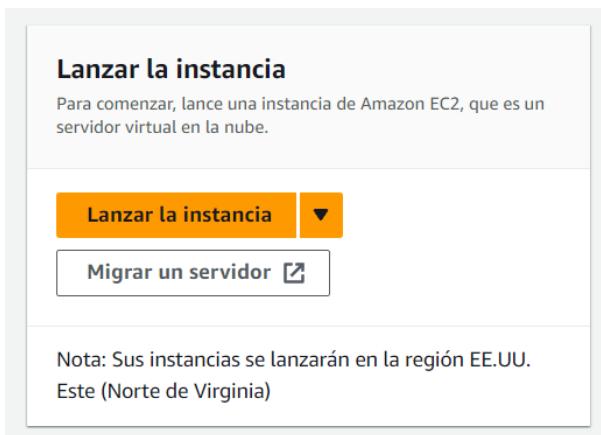
Pulsamos en AWS y entramos a la “Página de inicio de la Consola”

Entre los distintos servicios que nos ofrece buscamos el que dice EC2, en la categoría de informática.



EC2

Vamos a lanzar una instancia nueva, le damos al botón naranja “Lanzar la instancia”.



Ponemos un nombre a la instancia.



Y seleccionamos la imagen que queremos que tenga nuestra instancia, en nuestro caso será **Amazon Linux..**

▼ Imágenes de aplicaciones y sistemas operativos (Imagen de máquina de Amazon)

Una AMI es una plantilla que contiene la configuración de software (sistema operativo, servidor de aplicaciones y aplicaciones) necesaria para lanzar la instancia. Búsq[ue]o o examine las AMI si no ve lo que busca a continuación.

Q Busq[ue]o en nuestro catálogo completo que incluye miles de imágenes de sistemas operativos y aplicaciones

Recientes Inicio rápido

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	SUSE Li	
						Buscar más AMI Inclusión de AMI de AWS, Marketplace y la comunidad

▼ Tipo de instancia [Información](#) | [Obtener asesoramiento](#)

Tipo de instancia

t2.micro	Apto para la capa gratuita
Familia: t2 1 vCPU 1 GiB Memoria Generación actual: true	
Bajo demanda Windows base precios: 0.0162 USD por hora	
Bajo demanda SUSE base precios: 0.0116 USD por hora	
Bajo demanda RHEL base precios: 0.0716 USD por hora	
Bajo demanda Linux base precios: 0.0116 USD por hora	

Se aplican costos adicionales a las AMI con software preinstalado

Todas las generaciones

[Comparar tipos de instancias](#)

Generamos un par de claves nuevas

▼ Par de claves (inicio de sesión) [Información](#)

Puede utilizar un par de claves para conectarse de forma segura a la instancia. Asegúrese de que tiene acceso al par de claves seleccionado antes de lanzar la instancia.

Nombre del par de claves - *obligatorio*

[Crear un nuevo par de claves](#)

Ponemos nombre a la clave y nos aseguramos de crearlas en formato **RSA** y **.pem**

Crear par de claves

Nombre del par de claves
Con los pares de claves es posible conectarse a la instancia de forma segura.

El nombre puede incluir hasta 255 caracteres ASCII. No puede incluir espacios al principio ni al final.

Tipo de par de claves
 RSA
Par de claves pública y privada cifradas mediante RSA
 ED25519
Par de claves privadas y públicas cifradas ED25519

Formato de archivo de clave privada
 .pem
Para usar con OpenSSH
 .ppk
Para usar con PuTTY

⚠️ Cuando se le solicite, almacene la clave privada en un lugar seguro y accesible del equipo. **Lo necesitará más adelante para conectarse a la instancia.** [Más información](#)

[Cancelar](#) [Crear par de claves](#)

Permitimos el tráfico **SSH | HTTP | HTTPS**

▼ Configuraciones de red [Información](#)

Red | [Información](#)
vpc-0156ec04ea749c386

Subred | [Información](#)
Sin preferencias (subred predeterminada en cualquier zona de disponibilidad)

Asignar automáticamente la IP pública | [Información](#)

Habilitar

Se aplican cargos adicionales cuando no se cumplen los límites del **nivel gratuito**

Firewall (grupos de seguridad) | [Información](#)
Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agregue reglas para permitir que un tráfico específico llegue a la instancia.

Crear grupo de seguridad
 Seleccionar un grupo de seguridad existente

Crearemos un nuevo grupo de seguridad denominado "launch-wizard-7" con las siguientes reglas:

Permitir el tráfico de SSH desde [Ayuda a establecer conexión con la instancia](#)
Cualquier lugar
0.0.0.0/0

Y lanzamos la instancia.

The screenshot shows the AWS EC2 Instances page. A green success message box at the top says "Correcto" and "El lanzamiento de la instancia se inició correctamente (i-0a813b39b1f8a0321)". Below this, the main table lists one instance: "GymTrackerAWS" (i-0a813b39b1f8a0321), which is currently "Detenida". The "Actions" dropdown menu for this instance has "Iniciar instancia" highlighted. To the right, a "Resumen" (Summary) panel is open, showing basic instance details like the AMI, instance type (t2.micro), and storage. A callout box highlights the "Nivel gratuito" (Free tier) information, which includes 750 hours of usage for t2.micro instances. At the bottom right of the summary panel are buttons for "Cancelar", "Lanzar instancia" (Launch instance), and "Revisar comandos".

7.1.2 Conectarse a la instancia

Para conectarnos a la instancia por SSH seguiremos los pasos que nos indica.

This is a modal dialog titled "Conectarse a la instancia" with a "Información" link. It provides instructions for connecting to the instance i-0a813b39b1f8a0321 (GymTrackerAWS) via SSH. Step 1: Open an SSH client. Step 2: Locate the private key file "claveGymtracker.pem". Step 3: Execute the command "chmod 400 claveGymtracker.pem". Step 4: Connect to the instance using its public DNS: "ec2-54-210-36-75.compute-1.amazonaws.com". An example command is shown: "ssh -i \"claveGymtracker.pem\" ec2-user@ec2-54-210-36-75.compute-1.amazonaws.com". A note box states: "Nota: En la mayoría de los casos, el nombre de usuario adivinado es correcto. Sin embargo, lea las instrucciones de uso de la AMI para comprobar si el propietario de la AMI ha cambiado el nombre de usuario predeterminado de la AMI." At the bottom are "Cancelar" and "Aceptar" buttons.

Nos situamos en la ubicación en la que hemos guardado la clave previamente

y nos aseguramos de darle los permisos necesarios.

chmod 400 "claveGymtracker.pem"

```
tfg@tfg-VirtualBox:~/Escritorio$ chmod 400 "claveGymtracker.pem"
```

Nos conectamos a la instancia mediante ssh con el comando

ssh -i "nombre_de_la_clave" usuario@dns_de_la_instancia

```
tfg@tfg-VirtualBox:~/Escritorio$ ssh -i "claveGymtracker.pem" ec2-user@ec2-54-21
0-36-75.compute-1.amazonaws.com
The authenticity of host 'ec2-54-210-36-75.compute-1.amazonaws.com (54.210.36.75
)' can't be established.
ED25519 key fingerprint is SHA256:R0hNg6j8Fv680DVYEs1a8dekE5cfA1RL05BGMbZChuU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-210-36-75.compute-1.amazonaws.com' (ED25519)
to the list of known hosts.

,
  #
 ~\_ #####      Amazon Linux 2023
 ~~ \#####\
 ~~ \###|
 ~~ \#/   https://aws.amazon.com/linux/amazon-linux-2023
 ~~ \~'`-->
 ~~~ /
 ~~ ./
 /m/`-
 [ec2-user@ip-172-31-19-47 ~]$
```

7.1.3 Preparar instancia para el despliegue

Una vez que hemos comprobado que la instancia funciona correctamente la preparamos para el despliegue de la aplicación web.

Descargamos los jars de la aplicación desde github. en el siguiente enlace en la carpeta aws / Jars
<https://github.com/deiivvv/GymTracker>

Para copiar los archivos usamos el comando

scp -i nombreClave.pem nombreJar.jar usuario@dns_de_la_instancia:ubicacionEnLaInstancia

```
tfg@tfg-VirtualBox:~/Escritorio/GymTracker/aws$ scp -i ./Claves/claveGymtracker.p
em ./Jars/GymTracker-0.0.1-SNAPSHOT.jar ./Jars/Api-Ejercicios-0.0.1-SNAPSHOT.ja
r ec2-user@ec2-54-210-36-75.compute-1.amazonaws.com:/home/ec2-user/
GymTracker-0.0.1-SNAPSHOT.jar          100%    53MB  12.4MB/s  00:04
Api-Ejercicios-0.0.1-SNAPSHOT.jar     100%    45MB  15.5MB/s  00:02
```

Podemos comprobar que se copiaron los jars correctamente

```
[ec2-user@ip-172-31-19-47 ~]$ ls
Api-Ejercicios-0.0.1-SNAPSHOT.jar  GymTracker-0.0.1-SNAPSHOT.jar
[ec2-user@ip-172-31-19-47 ~]$ █
```

Nos aseguramos que tenemos Java en la instancia con el comando:

java -version

En el caso de no tenerlo o tenerlo en una versión del jdk por debajo de la 17 usamos el comando
sudo yum install java

Una vez completada la instalación de java se recomienda volver a tirar el java -version para comprobar que se ha instalado correctamente.

Lo mismo hacemos con la base de datos que usaremos, **mariadb**

sudo yum install mariadb105-server

```
[ec2-user@ip-172-31-19-47 ~]$ sudo yum install mariadb105-server
```

Una vez completa la instalación nos aseguramos de que mariadb esté iniciado

sudo systemctl start mariadb

O

sudo systemctl restart mariadb

```
[ec2-user@ip-172-31-19-47 ~]$ sudo systemctl restart mariadb
```

Accedemos a mariadb

sudo mysql

```
[ec2-user@ip-172-31-19-47 ~]$ sudo mysql
Welcome to the MariaDB monitor.  Command:
```

Y volcamos todo el contenido del archivo **gymtracker.sql** que se encuentra en el github en la carpeta bbdd

```
MariaDB [(none)]> DROP DATABASE IF EXISTS gymtracker;
Query OK, 0 rows affected, 1 warning (0.000 sec)
MariaDB [(none)]> CREATE DATABASE gymtracker;
-- -u alvaro -p alvaro
Query OK, 1 row affected (0.000 sec)
MariaDB [(none)]>
MariaDB [(none)]> DROP USER IF EXISTS 'dad'@'localhost';
Query OK, 0 rows affected, 1 warning (0.000 sec)
REVOKE ALL PRIVILEGES ON *.* FROM 'dad'@'localhost';
Query OK, 0 rows affected (0.000 sec)
MariaDB [(none)]> CREATE USER 'dad'@'localhost' IDENTIFIED BY 'padre';
Query OK, 1 row affected (0.001 sec)
MariaDB [(none)]> GRANT ALL PRIVILEGES ON gymtracker.* TO 'dad'@'localhost';
Query OK, 0 rows affected (0.001 sec)
MariaDB [(none)]> INSERT INTO usuarios (id, nombre, contrasena, rol) VALUES ('1', 'alvaro', '$2a$10$in/jj/6C7J4QwEW24yvax0W', 'administrador');
Query OK, 1 row affected (0.001 sec)
MariaDB [(none)]> USE gymtracker;
Database changed
MariaDB [(none)]> CREATE TABLE usuarios (
    >     id INT AUTO_INCREMENT,
    >     nombre VARCHAR(100) UNIQUE,
    >     contrasena VARCHAR(100),
    >     rol VARCHAR(50),
    >     PRIMARY KEY (id)
);
Query OK, 0 rows affected (0.001 sec)
```

The screenshot shows a code editor interface with two tabs: 'Code' and 'Blame'. The 'Code' tab displays the SQL script from the previous terminal session. The 'Blame' tab is visible at the top but contains no content. The code itself is a series of MySQL commands to drop and create a database, create a user, grant privileges, and create a table.

```
Code Blame 201 lines (161 loc) · 6.28 KB
1  DROP DATABASE IF EXISTS gymtracker;
2  CREATE DATABASE gymtracker;
3
4  DROP USER IF EXISTS 'dad'@'localhost';
5  CREATE USER 'dad'@'localhost' IDENTIFIED BY 'padre';
6  GRANT ALL PRIVILEGES ON gymtracker.* TO 'dad'@'localhost';
7
8  USE gymtracker;
9
10 CREATE TABLE usuarios (
11     id INT AUTO_INCREMENT,
12     nombre VARCHAR(100) UNIQUE,
13     contrasena VARCHAR(100),
14     rol VARCHAR(50),
15     PRIMARY KEY (id)
16 );
17
```

Para salir de mariadb ponemos **exit** en la consola.

7.1.4 Ejecutar GymTracker

Creamos los dos scripts necesarios para ejecutar los jar.

O los descargamos desde el github en la carpeta aws / Scripts y los copiamos en la instancia según los pasos previos.

sudo vim gymtracker.sh

```
[ec2-user@ip-172-31-19-47 ~]$ sudo vim gymtracker.sh
```

Le añadimos el siguiente contenido:

```
#!/bin/bash
nohup java -jar GymTracker-0.0.1-SNAPSHOT.jar > first.log 2>&1 &
nohup java -jar Api-Ejercicios-0.0.1-SNAPSHOT.jar > second.log 2>&1 &
```

```
#!/bin/bash
nohup java -jar GymTracker-0.0.1-SNAPSHOT.jar > first.log 2>&1 &
nohup java -jar Api-Ejercicios-0.0.1-SNAPSHOT.jar > second.log 2>&1 &
```

sudo vim stop_gymtracker.sh

```
[ec2-user@ip-172-31-19-47 ~]$ sudo vim stop_gymtracker.sh
```

Le añadimos el siguiente contenido:

```
#!/bin/bash
pkill -f 'java -jar GymTracker-0.0.1-SNAPSHOT.jar'
pkill -f 'java -jar Api-Ejercicios-0.0.1-SNAPSHOT.jar'
```

```
#!/bin/bash
pkill -f 'java -jar GymTracker-0.0.1-SNAPSHOT.jar'
pkill -f 'java -jar Api-Ejercicios-0.0.1-SNAPSHOT.jar'
```

gymtracker.sh -> levantará los dos jars en segundo plano y creará los archivos first.log y second.log donde se almacenará la información que devuelva la consola.

stop_gymtracker.sh -> finaliza la ejecución de los jars

Para ejecutar los scripts usamos el comando con permisos de administrador, **SUDO**,
sudo bash ./nombre_script.sh

sudo bash ./gymtracker.sh

```
[ec2-user@ip-172-31-19-47 ~]$ bash ./gymtracker.sh
```

sudo bash ./stop_gymtracker.sh

```
[ec2-user@ip-172-31-19-47 ~]$ bash ./stop_gymtracker.sh
```

Después de ejecutar el primer script y levantar la aplicación.

Para comprobar que se han levantado correctamente, usamos el comando:

ps -ef | grep java

Busca todos los procesos en ejecución y filtrarlos por “java” .

```
[ec2-user@ip-172-31-19-47 ~]$ ps -ef | grep java
ec2-user 27930      1 45 15:44 pts/0    00:00:02 java -jar GymTracker-0.0.1-SNAPSHOT.jar
ec2-user 27931      1 45 15:44 pts/0    00:00:02 java -jar Api-Ejercicios-0.0.1-SNAPSHOT.jar
ec2-user 27959      2408  0 15:44 pts/0    00:00:00 grep --color=auto java
```

Ahora puedes acceder desde cualquiera de tus navegadores con la url

https://dns_ipv4_publica_de_la_instancia

algo así como

 <https://ec2-18-208-158-106.compute-1.amazonaws.com>

el dns público de la instancia lo podrás encontrar en el apartado de Detalles de tu instancia en aws

The screenshot shows the AWS CloudWatch Metrics interface for an instance named 'GymTrackerAWS'. The top navigation bar includes tabs for 'Detalles', 'Estado y alarmas' (Novedad), 'Monitoreo', 'Seguridad', 'Redes', 'Almacenamiento', and 'Etiquetas'. The 'Detalles' tab is selected. Below the tabs, there's a section titled 'Resumen de instancia' with a 'Información' link. The main content area displays various instance details in a grid format:

ID de la instancia	Dirección IPv4 pública	Direcciones IPv4 privadas
i-0a813b39b1f8a0321 (GymTrackerAWS)	18.208.158.106 dirección abierta	172.31.19.47
Dirección IPv6	Estado de la instancia	DNS de IPv4 pública
-	En ejecución	ec2-18-208-158-106.compute-1.amazonaws.com dirección abierta
Tipo de nombre de anfitrión	Nombre DNS de IP privada (solo IPv4)	Direcciones IP elásticas
Nombre de IP: ip-172-31-19-47.ec2.internal	ip-172-31-19-47.ec2.internal	-
Responder al nombre DNS de recurso privado IPv4 (A)	Tipo de instancia	
	t2.micro	

Para finalizar la ejecución de la aplicación web ejecutamos el comando script stop_gymtracker
bash ./stop_gymtracker.sh

```
[ec2-user@ip-172-31-19-47 ~]$ bash ./stop_gymtracker.sh
```

7.1.5 IP Elástica

La dirección IPv4 de la instancia cambia cada vez que se inicia la instancia.

Para esto vamos a generar un ip elástica para que nuestra instancia mantenga siempre la misma ip y el mismo dns de aws.

Volvemos al panel de AWS, en el apartado de **Red y Seguridad**.

Entramos en **Direcciones IP elásticas**

▼ Red y seguridad

Security Groups

Direcciones IP elásticas

Grupos de ubicación

Pares de claves

Interfaces de red

Pulsamos el botón naranja Asignar la dirección IP elástica"



Mantenemos la configuración como está y le damos a asignar

The screenshot shows the 'Configuraciones de la dirección IP elástica' (Elastic IP configuration) page. At the top, it says 'Gruppo de borde de red' (Border group) and has a search bar with 'us-east-1'. Below this, there's a section for 'Grupo de direcciones IPv4 públicas' (Public IPv4 address group) with two options: 'Grupo de direcciones IPv4 de Amazon' (selected) and 'Conjunto de direcciones IPv4 propiedad del cliente'. There's also a note about AWS Global Accelerator. A 'Crear acelerador' (Create accelerator) button is present. The next section is 'Etiquetas: opcional' (Optional tags), which says 'No hay etiquetas asociadas a este recurso.' (No tags are associated with this resource). An 'Agregar nueva etiqueta' (Add new tag) button is available. At the bottom right are 'Cancelar' (Cancel) and 'Asignar' (Assign) buttons.

Una vez generada marcamos la dirección ip y entre las acciones le damos a “Asociar la dirección IP elástica”

The screenshot shows the 'Direcciones IP elásticas (1/1)' (Elastic IP addresses (1/1)) page. It lists one item: 'Dirección IPv4 pública : 54.156.245.113' (Type: IP pública). In the 'Acciones' (Actions) column, a dropdown menu is open with the 'Asociar la dirección IP elástica' (Associate elastic IP address) option highlighted. Other actions like 'Ver los detalles' (View details), 'Liberar direcciones IP elásticas' (Release elastic IP addresses), 'Desasociar la dirección IP elástica' (Disassociate elastic IP address), 'Actualizar DNS inverso' (Update reverse DNS), 'Activar transferencias' (Enable transfers), 'Desactivar transferencias' (Disable transfers), and 'Aceptar transferencias' (Accept transfers) are also listed.

Marcamos la instancia que hemos creado y le damos a "Asociar"

Asociar la dirección IP elástica[Información](#)

Elegir la instancia o la interfaz de red que se desea asociar a esta dirección IP elástica (54.156.245.113)

Dirección IP elástica: 54.156.245.113

Tipo de recurso
Elija el tipo de recurso al que desea asociar la dirección IP elástica.

Instancia
 Interfaz de red

⚠️ Si asocia una dirección IP elástica a una instancia que ya tiene asociada una dirección IP elástica, la dirección IP elástica asociada anteriormente se desasociará, pero la dirección seguirá asignada a la cuenta.
[Más información](#)

Si no se especifica ninguna dirección IP privada, la dirección IP elástica se asociará a la dirección IP privada principal.

Instancia

Dirección IP privada
La dirección IP privada a la que desea asociar la dirección IP elástica.

Nueva asociación
Especifique si la dirección IP elástica se puede volver a asociar a un recurso diferente en el caso de que ya exista otra asociación.
 Permitir que se vuelva a asociar esta dirección IP elástica

Cancelar **Asociar**

Ya hemos asignado una dirección ip elástica a nuestra instancia. Reinicia la instancia para asegurarte de que esta se asocia correctamente.

No olvides apagar el laboratorio cuando termines.

7.2 Despliegue en local con Docker



7.2.1 Instalación de docker

7.2.1.1 Linux:

Para instalar docker en linux abrimos una terminal y ejecutamos el siguiente comando:

sudo apt install docker.io

Y para poder usar docker sin sudo metemos nuestro usuario en el grupo de docker con este comando:

sudo usermod -aG docker NuestroUsuario

Y posteriormente reiniciamos nuestra máquina para que se aplique el cambio de grupo a nuestro usuario:

sudo reboot

7.2.1.2. Para Windows:

Para instalar docker en windows nos ayudamos de este tutorial:

<https://youtu.be/ZO4KWQfUBBc?si=ii53FktLvmKTIxQ8>

Siguiendo el tutorial vamos a descargar docker con wsl (Windows Subsystems for Linux).

Para ello nos ayudaremos de las documentaciones oficiales:

- documentación oficial de docker:

<https://docs.docker.com/desktop/install/windows-install/>

- documentación oficial de microsoft:

[Install WSL | Microsoft Learn](#)

Comenzaremos por instalar WSL con el siguiente comando en una consola powershell:

wsl –install

```
PS C:\Users\damag> wsl --install
```

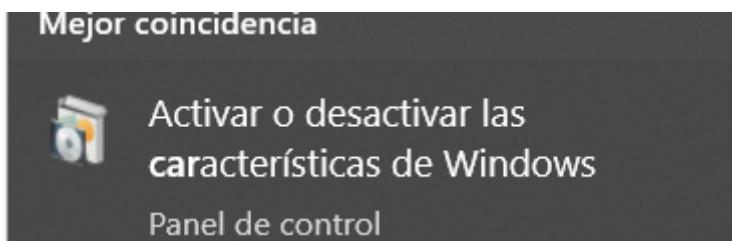
Reiniciamos y se nos abre la ventana de que se está instalando wsl y luego nos pide usuario y contraseña:

```
Ubuntu
Installing, this may take a few minutes...

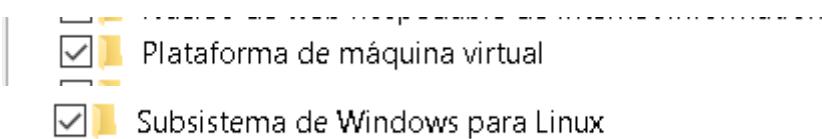
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: 
```

Rellenamos y ya tenemos wsl en nuestro windows.

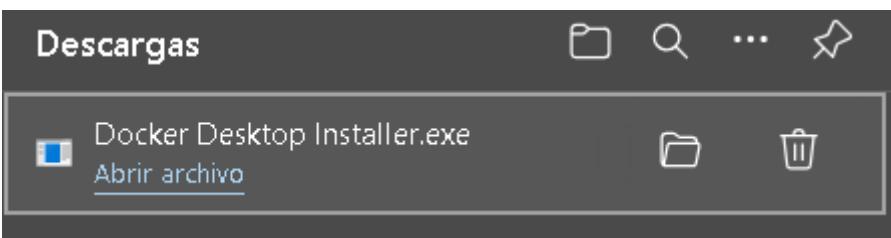
Después de esto seguimos los pasos de la documentación oficial y configuramos la virtualización. Para ello buscamos **features o características**:



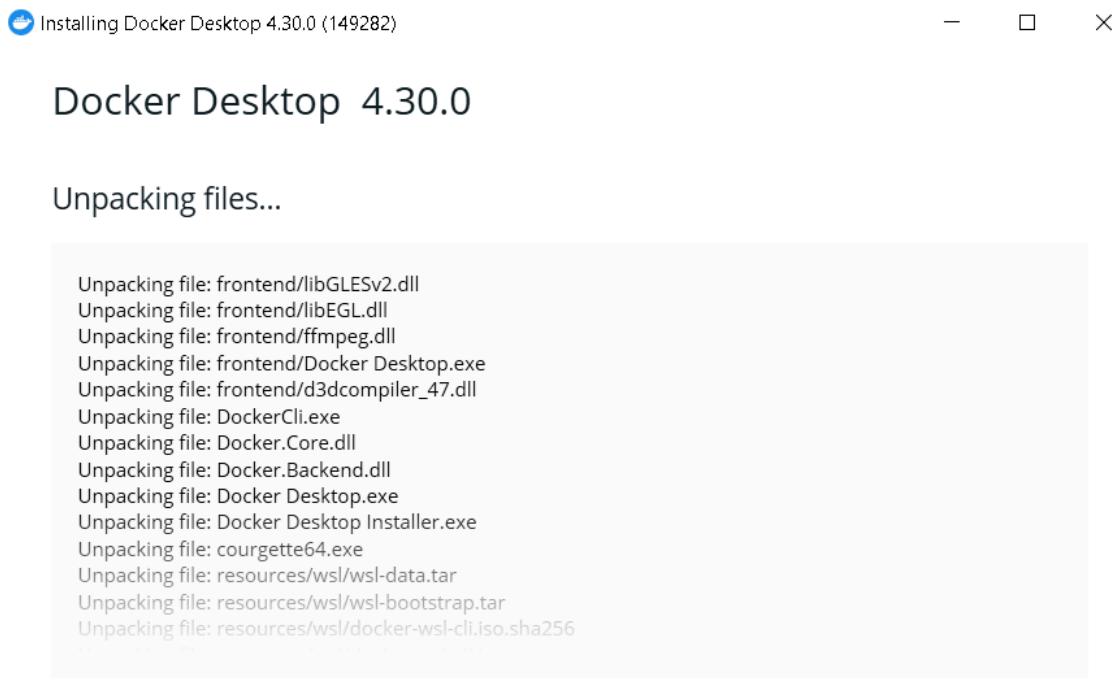
Abrimos y buscamos las opciones de **virtual machine platform o plataforma de máquina virtual y Windows Subsystems for Linux o Subsistema de Windows para Linux** y activamos las opciones si no están activadas:



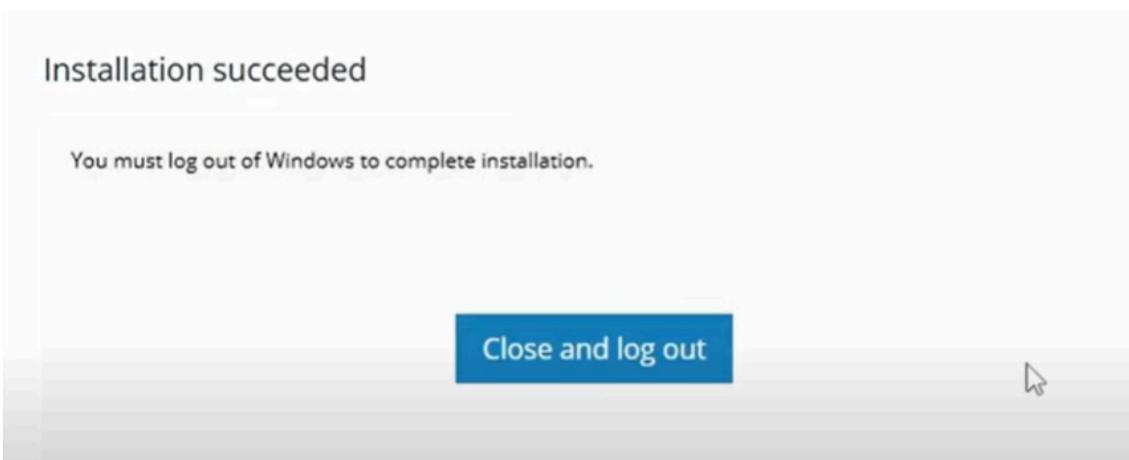
Después de esto ya podremos instalar docker desktop desde la página oficial descargamos el instalador:



Y lo ejecutamos:



Nos saldrá esto al terminar:



Pulsamos en **Close and log out** y se reiniciará la máquina una vez se reinicie nos muestra los términos de docker desktop los aceptamos y ya tendremos docker desktop y podremos ejecutar comandos de docker en la terminal.

Para comprobarlo podemos poner el siguiente comando para ver la versión de docker que tenemos:

docker –version

```
C:\Users\damag>docker --version
Docker version 26.1.1, build 4cf5afa
```

7.2.2 Preparación para el despliegue

Para preparar el despliegue de nuestra aplicación en docker comenzamos por crear una carpeta para docker en la que pusimos lo siguiente:

- Un readme con comandos de interés de docker a la hora de desplegar llamado **readme_comandos_docker.md**.

Ejecutar Docker Compose para Generar o Iniciar los Contenedores

```
docker-compose -p contenedor-gymtracker up --build
```

Este comando genera los contenedores de GymTracker y los inicia. Si es la primera vez que se ejecuta este comando o si se realizaron cambios en la configuración, se reconstruirán las imágenes de Docker antes de iniciar los contenedores.

Parar Contenedores

Para detener los contenedores de GymTracker, presiona **Ctrl + C** en la terminal donde se están ejecutando.

Ver Contenedores

```
docker ps -a
```

Este comando muestra una lista de todos los contenedores, incluidos los que están detenidos.

Ver Volúmenes

```
docker volume ls
```

Este comando muestra una lista de todos los volúmenes Docker en el sistema.

Acceder a la Base de Datos del Contenedor

Para acceder a la base de datos del contenedor de la base de datos de GymTracker, primero accede al shell del contenedor y luego ingresa a MySQL:

```
docker exec -it contenedor-gymtracker_database_1 /bin/bash  
mysql -u root -p
```

La contraseña predeterminada es "root_password".

Acceder a los Logs de los Contenedores de la app y la api

Puedes acceder a los logs de los contenedores de la aplicación y la API de GymTracker de la siguiente manera:

```
docker logs contenedor-gymtracker_app_1  
docker logs contenedor-gymtracker_api_1
```

Borrar Contenedores de GymTracker

```
docker rm contenedor-gymtracker_database_1 contenedor-gymtracker_app_1 contenedor-gymtracker_api_1
```

Este comando elimina los contenedores específicos de GymTracker.

Borrar Volumen de GymTracker

```
docker volume rm contenedor-gymtracker_db_data
```

Este comando elimina el volumen llamado "contenedor-gymtracker_db_data", utilizado por la base de datos de GymTracker.

- **Los jars**, el jar de la api de ejercicios y el jar de la app de GymTracker.

- Los **Dockerfile** para cada jar para crear una imagen a partir de la cuál crear el contenedor para la api y para la app.

- **Dockerfile.api:**

```

1 # Dockerfile para la API
2 FROM openjdk:17-jdk-slim
3 COPY Api-Ejercicios-0.0.1-SNAPSHOT.jar /api/api.jar
4 WORKDIR /api
5 CMD ["java", "-jar", "api.jar"]

```

FROM openjdk:17-jdk-slim:

Esta línea especifica la imagen base que se utilizará para construir la nueva imagen. En este caso, se utiliza una imagen de OpenJDK 17 en su versión "slim", que es una versión más ligera y optimizada.

COPY Api-Ejercicios-0.0.1-SNAPSHOT.jar /api/api.jar:

Esta línea copia el archivo Api-Ejercicios-0.0.1-SNAPSHOT.jar desde el sistema de archivos del host (donde se encuentra el Dockerfile) a la ruta /api/api.jar dentro de la imagen Docker. Este archivo es el ejecutable de la API.

WORKDIR /api:

Esta línea establece el directorio de trabajo dentro del contenedor en /api. Esto significa que cualquier comando que se ejecute después de esta línea se ejecutará dentro de este directorio.

CMD ["java", "-jar", "api.jar"]:

Esta línea especifica el comando que se ejecutará cuando se inicie un contenedor basado en esta imagen. En este caso, se ejecuta el comando java -jar api.jar para iniciar la API.

- **Dockerfile.app:**

```

1 # Dockerfile para la aplicación
2 FROM openjdk:17-jdk-slim
3
4 # Copia el archivo JAR de tu aplicación
5 COPY GymTracker-0.0.1-SNAPSHOT.jar /app/app.jar
6
7 # Copia el script wait-for-it
8 COPY wait-for-it.sh /app/wait-for-it.sh
9
10 # Para que el script wait-for-it.sh tenga permisos de ejecución
11 RUN chmod +x /app/wait-for-it.sh
12
13 # Establecer el directorio de trabajo
14 WORKDIR /app
15
16 # Comando para iniciar la aplicación, esperando a que la base de datos esté lista
17 CMD ["/wait-for-it.sh", "database:3306", "--", "java", "-jar", "app.jar"]

```

Este Dockerfile crea una imagen Docker para la aplicación Java GymTracker, asegurándose de que la aplicación espere hasta que la base de datos esté lista antes de iniciarse.

FROM openjdk:17-jdk-slim:

Usa una imagen base ligera de OpenJDK 17 para ejecutar la aplicación Java.

COPY GymTracker-0.0.1-SNAPSHOT.jar /app/app.jar:

Copia el archivo JAR de la aplicación al contenedor.

COPY wait-for-it.sh /app/wait-for-it.sh:

Copia el script wait-for-it.sh al contenedor para esperar a que la base de datos esté lista.

RUN chmod +x /app/wait-for-it.sh:

Hace que el script sea ejecutable.

WORKDIR /app:

Establece /app como el directorio de trabajo dentro del contenedor.

CMD ["./wait-for-it.sh", "database:3306", "--", "java", "-jar", "app.jar"]:

Usa wait-for-it.sh para esperar a que la base de datos esté disponible en database:3306 antes de iniciar la aplicación con java -jar app.jar.

Uso del script wait-for-it.sh

El script wait-for-it.sh asegura que la aplicación GymTracker no intente conectarse a la base de datos antes de que esta esté lista, evitando errores de conexión. Esto nos sirvió para que el Docker Compose pudiera ejecutar el contenedor de la base de datos y la aplicación simultáneamente.

- **Script wait-for-it:**

El script `wait-for-it` lo obtuvimos de GitHub, y puedes encontrarlo en el siguiente enlace: <https://github.com/vishnubob/wait-for-it>. Este script nos ayudó a solucionar el error que ocurría al hacer el `docker-compose up`, donde la aplicación no se iniciaba correctamente porque dependía de que la base de datos estuviera disponible primero. Utilizando `wait-for-it`, pudimos hacer que nuestra aplicación esperaría hasta que la base de datos estuviera lista antes de intentar iniciarse, asegurando así un inicio ordenado y sin errores.

- Archivo para la inicialización de la base de datos **init.sql**.

- **docker-compose.yml :**

Este archivo docker-compose.yml lo hemos creado para desplegar la aplicación Gymtracker. Configura tres contenedores: una base de datos MySQL, la API de ejercicios y la aplicación principal. A continuación, una explicación breve de cada sección:

Versión:

```
version: '3.8'
```

- Especificamos la versión del formato de **docker-compose**.

Servicios

Base de datos (database)

```
services:  
  database:  
    image: mysql:8.0  
    environment:  
      MYSQL_ROOT_PASSWORD: root_password  
      MYSQL_DATABASE: gymtracker  
      MYSQL_USER: dad  
      MYSQL_PASSWORD: padre  
    ports:  
      - "3307:3306"  
    volumes:  
      - db_data:/var/lib/mysql  
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql  
    networks:  
      - app-network
```

- Usamos MySQL 8.0 con configuraciones de entorno para la base de datos.
- Mapeamos el puerto 3306 del contenedor al 3307 del host.
- Definimos volúmenes para la persistencia de datos y la inicialización.
- Conectamos a la red app-network.

API (api)

```
api:  
  build:  
    context: .  
    dockerfile: Dockerfile.api  
  ports:  
    - "8081:8081"  
  networks:  
    - app-network
```

- Construimos la imagen usando Dockerfile.api.
- Mapeamos el puerto 8081 del contenedor al 8081 del host.

- Conectamos a la red app-network.

Aplicación (app)

```
app:
  build:
    context: .
    dockerfile: Dockerfile.app
  depends_on:
    - database
    - api
  environment:
    SPRING_DATASOURCE_URL: jdbc:mysql://database:3306/gymtracker
    SPRING_DATASOURCE_USERNAME: dad
    SPRING_DATASOURCE_PASSWORD: padre
    SERVER_PORT: 443
  ports:
    - "443:443"
  networks:
    - app-network
```

- Construimos la imagen usando Dockerfile.app.
- Establecemos dependencias para que la aplicación espere a que database y api estén listas.
- Configuramos variables de entorno para la conexión a la base de datos.
- Mapeamos el puerto 443 del contenedor al 443 del host.
- Conectamos a la red app-network.

Redes y Volúmenes

```
networks:
  app-network:
    driver: bridge
volumes:
  db_data:
```

- Definimos una red app-network para la comunicación entre servicios.
- Definimos un volumen db_data para la persistencia de datos de la base de datos.

- Con este docker-compose.yml, aseguramos que la aplicación GymTracker espere a que la base de datos y la API estén listas antes de iniciar, previniendo errores de dependencias.

Una vez hemos hecho la carpeta ya está todo listo para el despliegue.

7.2.3 Despliegue

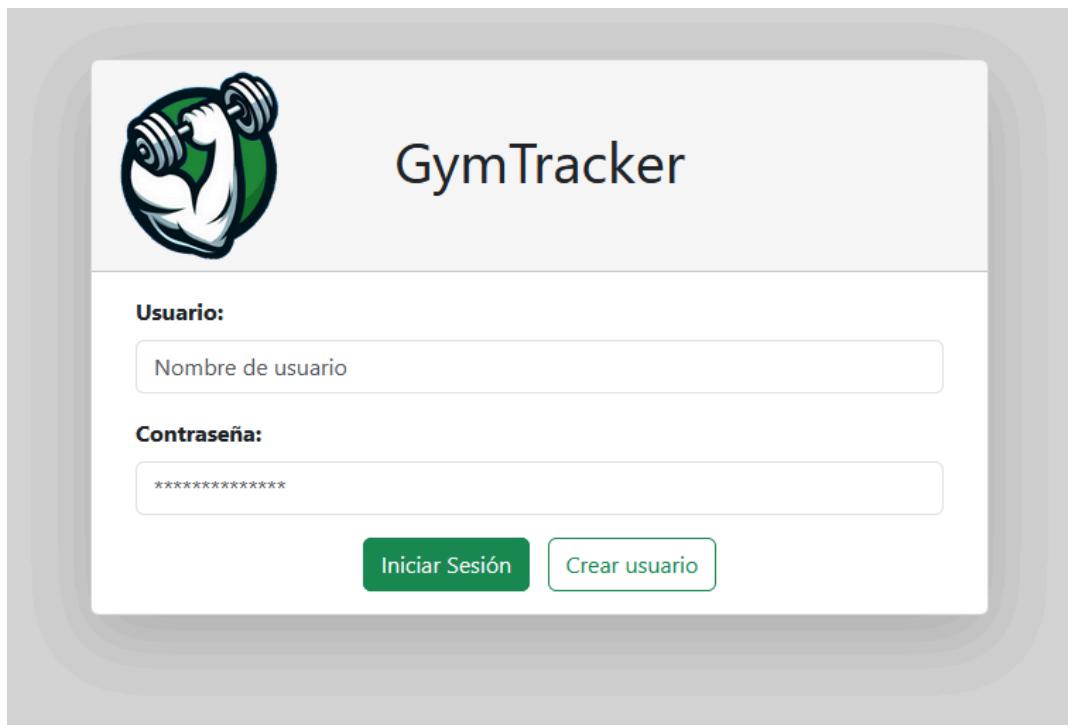
Tanto en linux como windows:

1. Entrar en el github de GymTracker:
[deiivvv/GymTracker: TFG de Alvaro Gallego, Daniel Marin y David Martin \(github.com\)](https://github.com/deiivvv/GymTracker)
2. Localiza el archivo ZIP llamado despliegue_docker.zip en el repositorio y descárgalo.
3. Descomprime el archivo ZIP en la ubicación deseada.
4. Abre la carpeta descomprimida y verás el archivo readme_comandos_docker.md en el que tienes los comandos para despliegue y mantenimiento del mismo.
5. Abre una terminal en la carpeta y ejecuta el siguiente comando:

docker-compose -p contenedor-gymtracker up --build

Y ya tendrás GymTracker listo para acceder en tu navegador desde la siguiente dirección:

<https://localhost>



7.3 Despliegue en local con Spring Tool Suite (STS)

7.3.1 Prerrequisitos

- 1. Java Development Kit (JDK):** Asegúrate de tener instalado JDK (Java Development Kit) 17 o superior.
- 2. Git:** Necesitas Git para clonar el repositorio desde GitHub.
- 3. Spring Tool Suite (STS):** Un IDE recomendado para el desarrollo de aplicaciones Spring.
- 4. MySQL:** MySQL es un sistema de gestión de bases de datos relacional de código abierto.

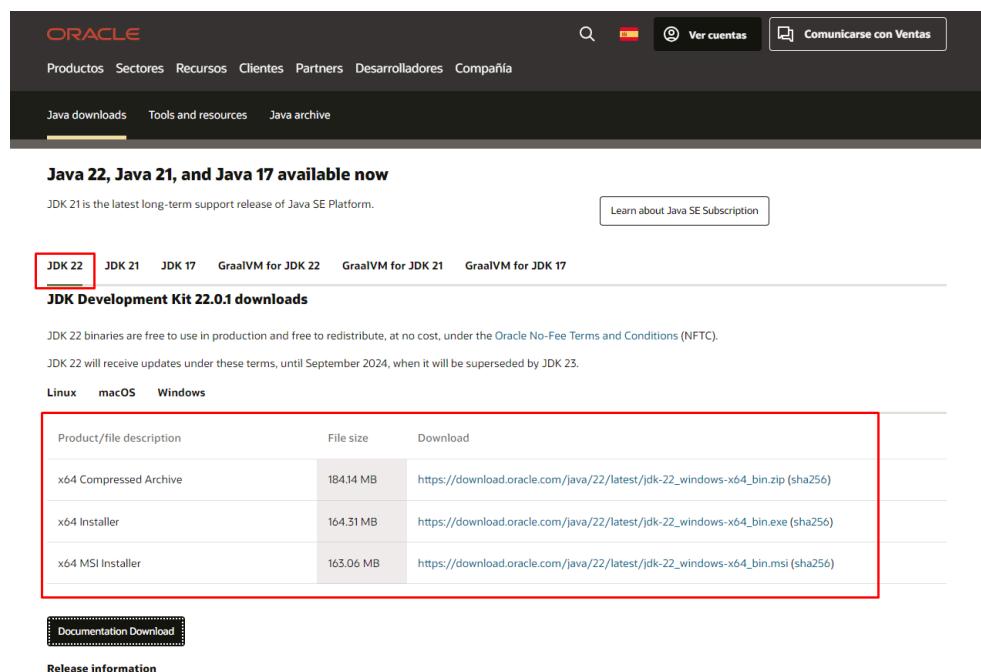
7.3.1.1 Instalación de Prerrequisitos

Windows

7.3.1.1.1 Instalar JDK

Descarga el JDK desde [Oracle JDK o OpenJDK](#).

Sigue las instrucciones del instalador.



The screenshot shows the Oracle Java Downloads page. At the top, there's a navigation bar with links for Productos, Sectores, Recursos, Clientes, Partners, Desarrolladores, and Compañía. Below that is a secondary navigation bar with links for Java downloads, Tools and resources, and Java archive. A banner at the top says "Java 22, Java 21, and Java 17 available now". Below the banner, it says "JDK 21 is the latest long-term support release of Java SE Platform." There are tabs for JDK 22, JDK 21, JDK 17, GraalVM for JDK 22, GraalVM for JDK 21, and GraalVM for JDK 17. The "JDK 22" tab is selected. A section titled "JDK Development Kit 22.0.1 downloads" lists three download options: "x64 Compressed Archive" (184.14 MB), "x64 Installer" (164.31 MB), and "x64 MSI Installer" (163.06 MB). Each option has a "Download" link next to it. A red box highlights the download links for the compressed archive, installer, and MSI installer. At the bottom of the section, there are "Documentation Download" and "Release information" links.

7.3.1.1.2 Instalar Git

[Descarga Git desde Git.](#)
Sigue las instrucciones del instalador.

The screenshot shows the official Git website's download section for Windows. It features a search bar at the top right. On the left, there are links for 'About', 'Documentation', 'Downloads' (which is highlighted in red), 'GUI Clients', 'Logos', and 'Community'. Below these are sections for the 'Pro Git book' and 'Other Git for Windows downloads'. A prominent red box highlights the 'Standalone Installer' link. To its right are links for '32-bit Git for Windows Setup.' and '64-bit Git for Windows Setup.'. Further down, there are links for 'Portable ("thumbdrive edition")', '32-bit Git for Windows Portable.', and '64-bit Git for Windows Portable.'. A note about winget tool installation follows, along with a command-line snippet. The 'Now What?' section includes three icons: a book, a mouse, and a speech bubble.

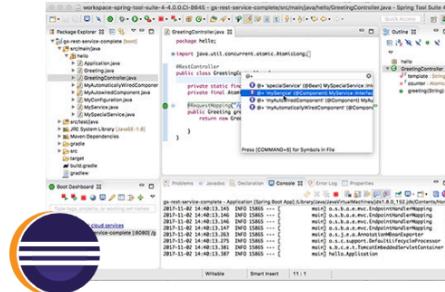
7.3.1.1.3 Instalar Spring Tool Suite (STS)

[Descarga STS desde Spring Tool Suite.](#)
Sigue las instrucciones del instalador.

Spring Tools 4 for Eclipse

The all-new Spring Tool Suite 4. Free.
Open source.

- 4.22.1 - LINUX X86_64
- 4.22.1 - LINUX ARM_64
- 4.22.1 - MACOS X86_64
- 4.22.1 - MACOS ARM_64
- 4.22.1 - WINDOWS X86_64**



7.3.1.1.4 Instalar MySql

Para instalar MySQL en Windows:

1. Descargar MySQL Installer desde [MySql](#).

④ MySQL Community Downloads

MySQL Installer

General Availability (GA) Releases Archives

MySQL Installer 8.0.37

Note: MySQL 8.0 is the final series with MySQL Installer. As of MySQL 8.1, use a MySQL product's MSI or Zip archive for installation. MySQL Server 8.1 and higher also bundle MySQL Configurator, a tool that helps configure MySQL Server.

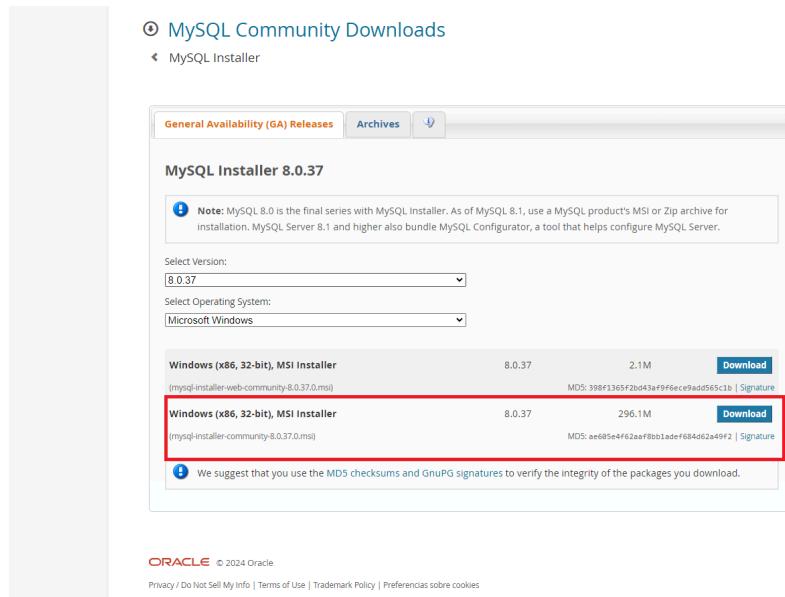
Select Version: 8.0.37

Select Operating System: Microsoft Windows

Windows (x86, 32-bit), MSI Installer	8.0.37	2.1M	Download
(mysql-installer-web-community-8.0.37.0.msi)		MD5: 198f1345f2bd43af9f6ce9ad0d5651cb Signature	
Windows (x86, 32-bit), MSI Installer	8.0.37	296.1M	Download
(mysql-installer-community-8.0.37.0.msi)		MD5: ae605e4f62aa8bb1adef684d62a49f2 Signature	

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

ORACLE © 2024 Oracle
Privacy / Do Not Sell My Info | Terms of Use | Trademark Policy | Preferencias sobre cookies

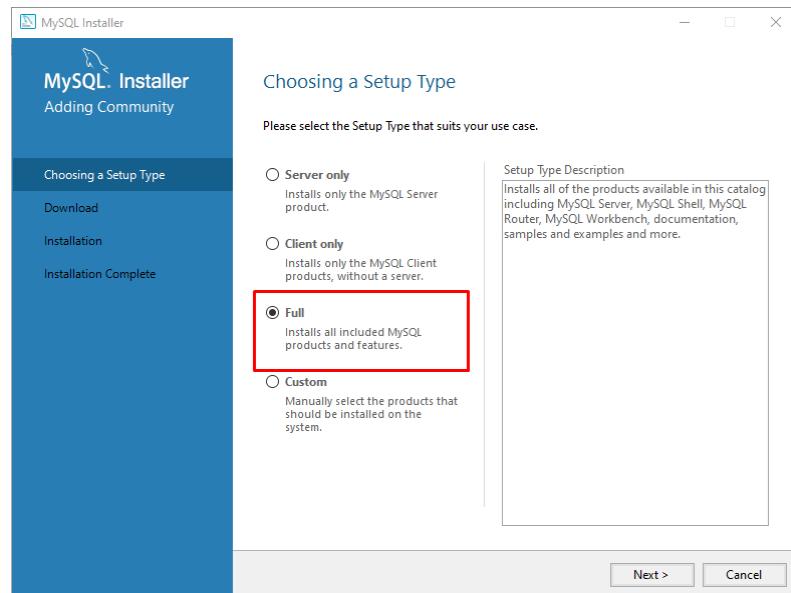


2. Ejecutar el instalador:

- Abre el archivo descargado.

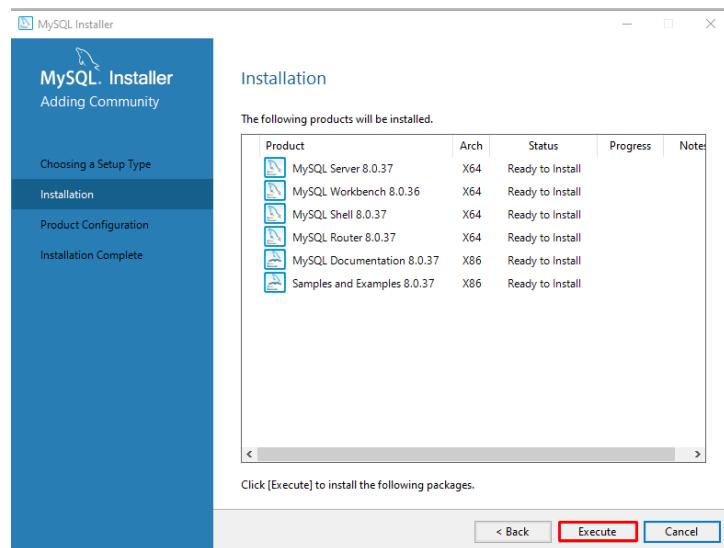
3. Seleccionar el tipo de instalación.

- Elige "Full" para una instalación completa.
- Haz clic en "Next".



4. Instalar las dependencias.

- El instalador verificará y descargará las dependencias necesarias.
- Haz clic en "Execute" para proceder.

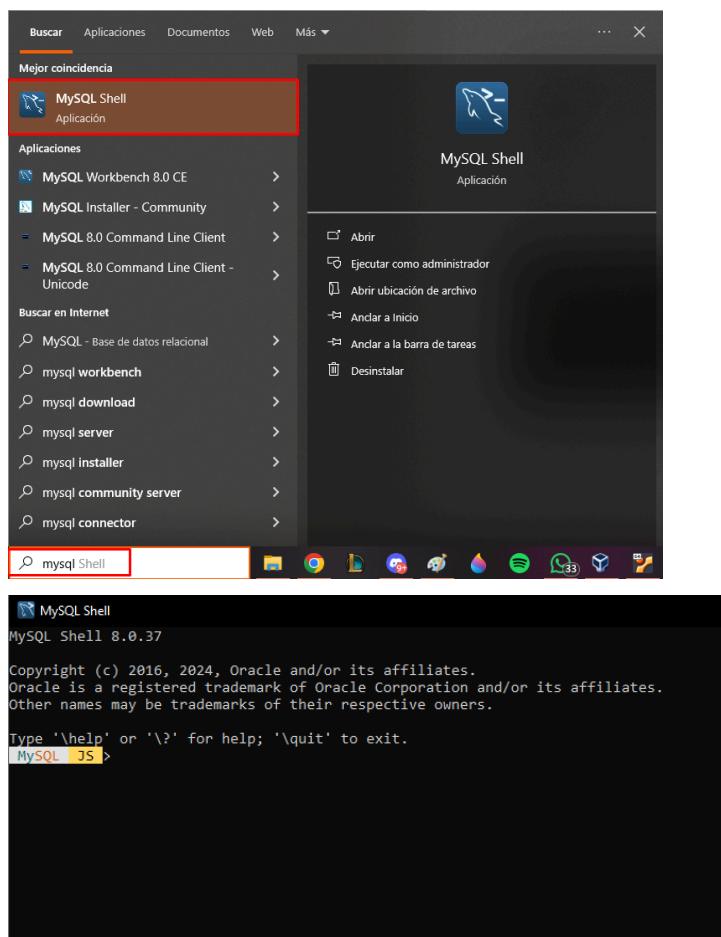


5. Configuración del servidor MySQL.

- Elige el tipo de configuración (General o Server Only).
- Configura el tipo y puerto de red (generalmente, puedes dejar las opciones predeterminadas).
- Establece una contraseña para el usuario `root` y, opcionalmente, crea usuarios adicionales.
- Configura el servicio de Windows si deseas que MySQL se inicie automáticamente con el sistema.

6. Finalizar la instalación.

- Una vez finalizada la instalación, puedes abrir MySQL Workbench o MySQL Shell para empezar a trabajar con MySQL.



Linux

7.3.1.2.1 Instalar JDK

Compruebe la versión de java que tiene

java -version

En caso de que no tenga o tenga una inferior al jdk-17, instale y/o actualice su jdk

sudo apt update

sudo apt install openjdk-19-jdk (última versión disponible)

7.3.1.2.2 Instalar Git

sudo apt install git

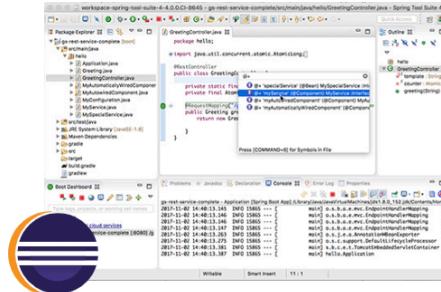
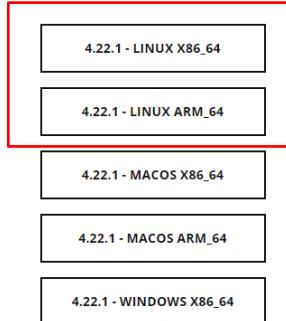
7.3.1.2.3 Instalar Spring Tool Suite (STS)

Descarga STS desde [Spring Tool Suite](#).

Extrae el archivo y sigue las instrucciones de instalación correspondientes para tu distribución.

Spring Tools 4 for Eclipse

The all-new Spring Tool Suite 4. Free.
Open source.

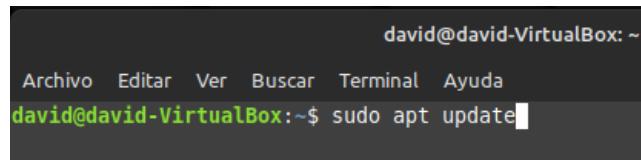


7.3.1.2.4 Instalar MySql

Para instalar MySQL en Linux (Debian)

1. Actualizar los repositorios:

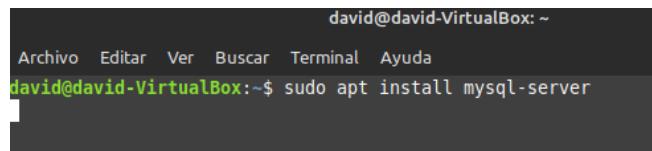
```
sudo apt update
```



A screenshot of a terminal window titled "david@david-VirtualBox: ~". The window has a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The command "david@david-VirtualBox:~\$ sudo apt update" is typed at the prompt.

2. Instalar MySQL:

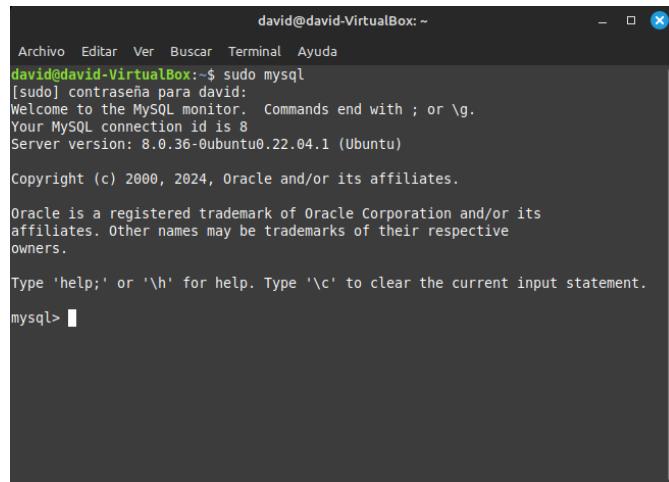
```
sudo apt install mysql-server
```



A screenshot of a terminal window titled "david@david-VirtualBox: ~". The window has a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The command "david@david-VirtualBox:~\$ sudo apt install mysql-server" is typed at the prompt.

3. Iniciamos MySQL para comprobar que funciona:

```
sudo mysql
```



A screenshot of a terminal window titled "david@david-VirtualBox: ~". The window has a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The command "david@david-VirtualBox:~\$ sudo mysql" is typed at the prompt. The terminal then displays the MySQL monitor interface, showing connection details and copyright information.

Estos pasos deberían permitirte instalar y configurar MySQL en tu sistema Linux.

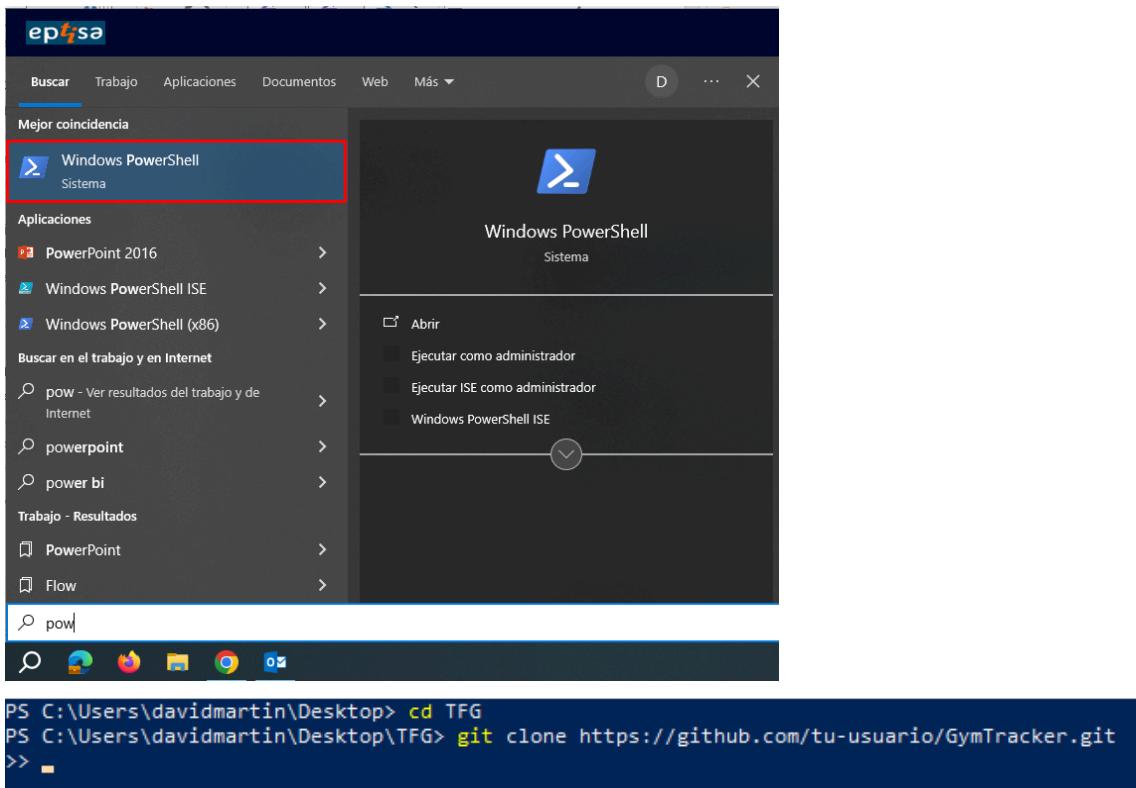
7.3.2 Clonar el Repositorio desde GitHub

Windows

En una terminal de powershell (pulsamos botón de windows y escribimos powershell), nos posicionamos en el directorio donde queremos que se clone el repositorio, podemos movernos con el comando **cd**, una vez estemos situados en nuestro destino, ejecutamos el siguiente comando para clonar el repositorio:

```
git clone https://github.com/tu-usuario/GymTracker.git
```

Reemplaza `tu-usuario` con el nombre de tu usuario en GitHub.

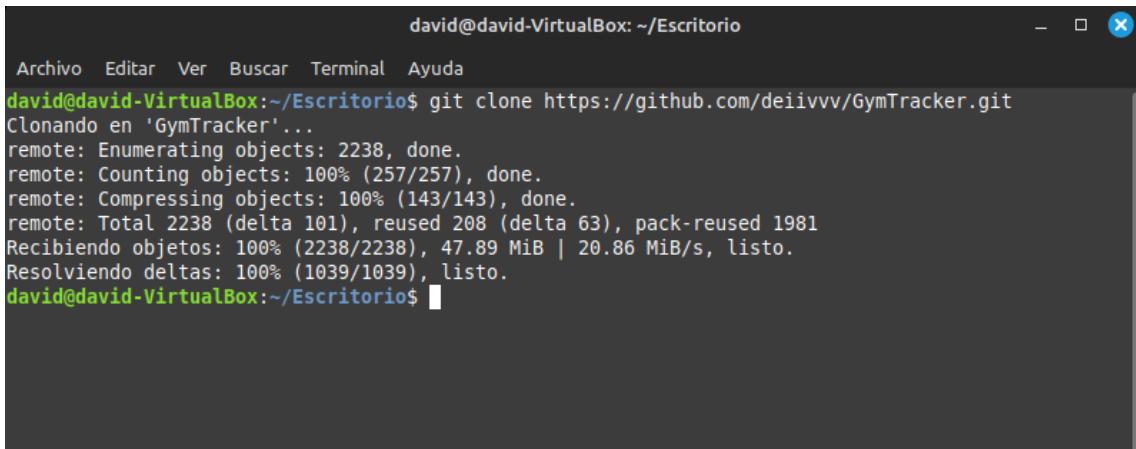


Linux

Abrimos una terminal, nos posicionamos en el directorio donde queremos que se clone el repositorio, podemos movernos con el comando **cd**, una vez estemos situados en nuestro destino, ejecutamos el siguiente comando para clonar el repositorio:

```
git clone https://github.com/tu-usuario/GymTracker.git
```

Reemplaza `tu-usuario` con el nombre de tu usuario en GitHub.

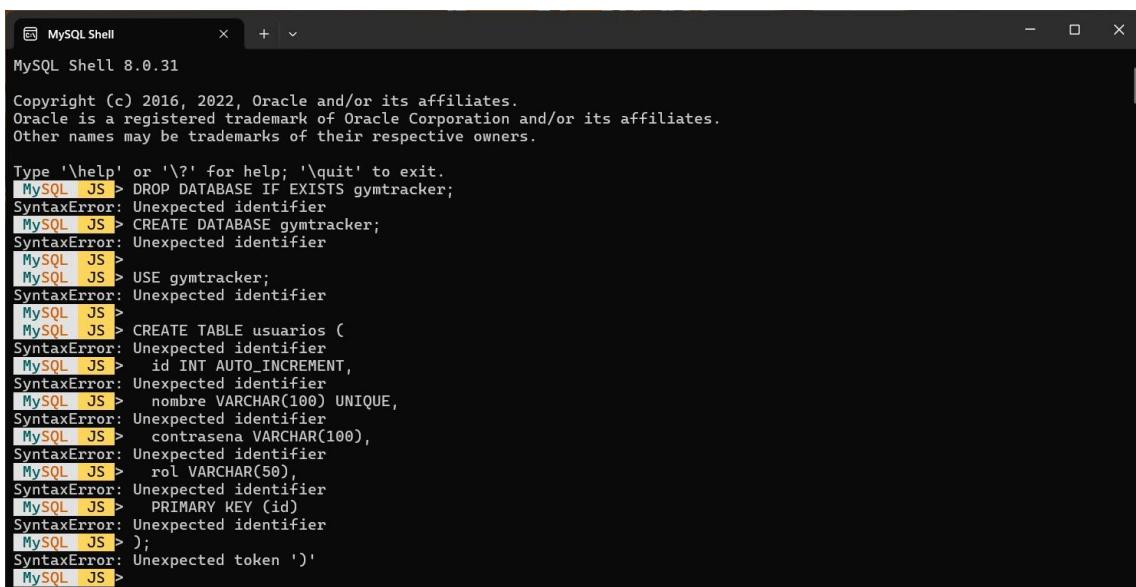


A screenshot of a terminal window titled "david@david-VirtualBox: ~/Escritorio". The window shows the command "git clone https://github.com/deiivvv/GymTracker.git" being run. The output indicates the cloning process is complete, showing object enumeration, counting, compressing, and receiving data from the remote repository. The final message says "Resolviendo deltas: 100% (1039/1039), listo." The terminal prompt "david@david-VirtualBox:~/Escritorio\$" is visible at the bottom.

7.3.3 Instalar la base de datos

Windows

Abrimos la carpeta clonada de Github, entramos en la carpeta BBDD ya abrimos el archivo Gymtracker.sql y copiamos todo el contenido. Pegamos el contenido en nuestro MySql Shell.



A screenshot of the MySQL Shell interface, version 8.0.31. The window title is "MySQL Shell". The shell prompt is "MySQL>". The user is executing the file "Gymtracker.sql", which contains SQL commands to create a database and a table named "usuarios". The shell displays several "SyntaxError: Unexpected identifier" messages, indicating that the SQL syntax used in the file is not valid according to the MySQL parser. The command "CREATE TABLE usuarios (" is shown, followed by various column definitions and constraints.

Ya tendríamos nuestra base de datos creada.

Linux

Abrimos la carpeta clonada de Github, entramos en la carpeta BBDD y abrimos el archivo Gymtracker.sql y copiamos todo el contenido.

```
tfg@tfg-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
tfg@tfg-VirtualBox:~$ sudo mysql;
[sudo] contraseña para tfg:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> DROP DATABASE IF EXISTS gymtracker;
routine, 33, @id_serie_flexiones_diamQuery OK, 6 rows affected (0,34 sec)

mysql> CREATE DATABASE gymtracker;
routine, 33, @id_serie_flexiones_diamQuery OK, 6 rows affected (0,34 sec)

mysql>
mysql> USE gymtracker;
Database changed
mysql>
mysql> CREATE TABLE usuarios (
    ->     id INT AUTO_INCREMENT,
    ->     nombre VARCHAR(100) UNIQUE,
    ->     contrasena VARCHAR(100),
```

Ya tendríamos nuestra base de datos creada.

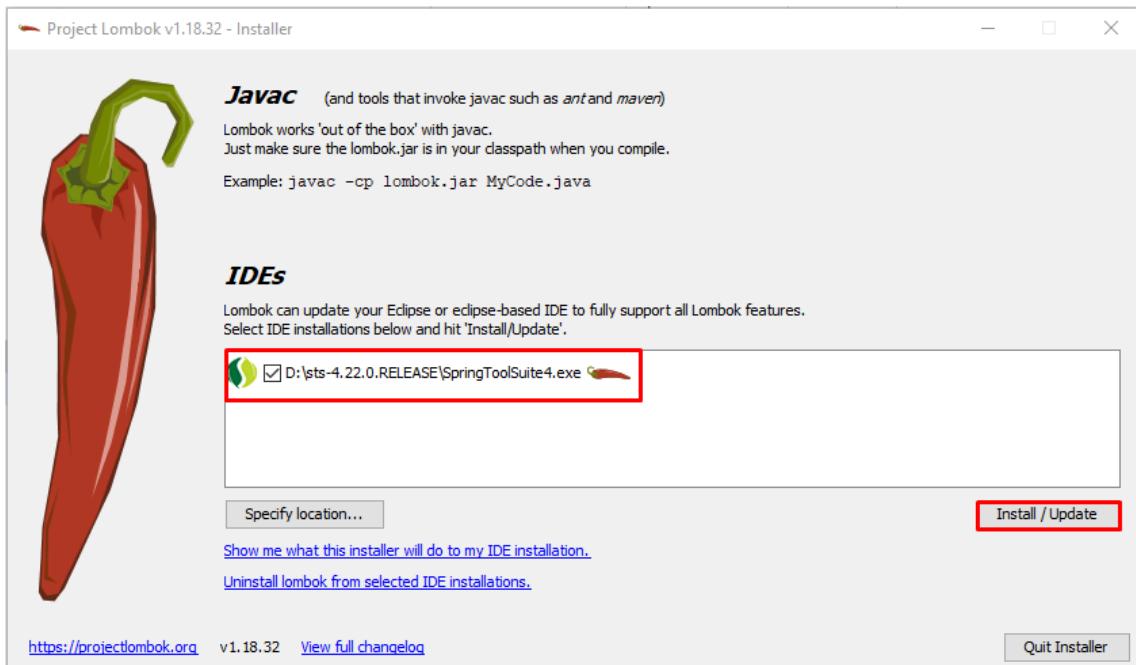
7.3.4 Ejecutar la Aplicación

Windows

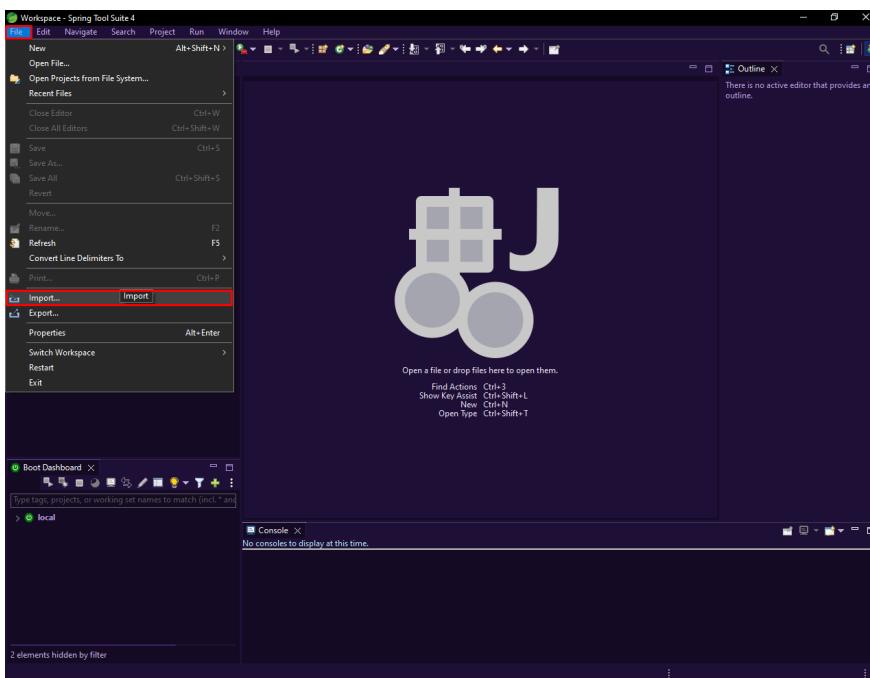
Una vez que la clonación sea exitosa, primero vamos a instalar **lombok** (Biblioteca Java diseñada para minimizar el código repetitivo que se escribe en las clases Java.). Ahora ya podemos ejecutar la aplicación desde SpringBoot.

Nombre	Fecha de modificación	Tipo	Tamaño
Api-Ejercicios	24/04/2024 8:03	Carpeta de archivos	
BBDD	16/05/2024 13:28	Carpeta de archivos	
GymTracker-AppWeb	16/05/2024 13:28	Carpeta de archivos	
Anteproyecto.pdf	24/04/2024 7:53	Documento Adob...	96 KB
README.md	16/05/2024 13:28	Archivo de origen ...	2 KB
lombok.jar	29/04/2024 8:00	Executable Jar File	1.999 KB

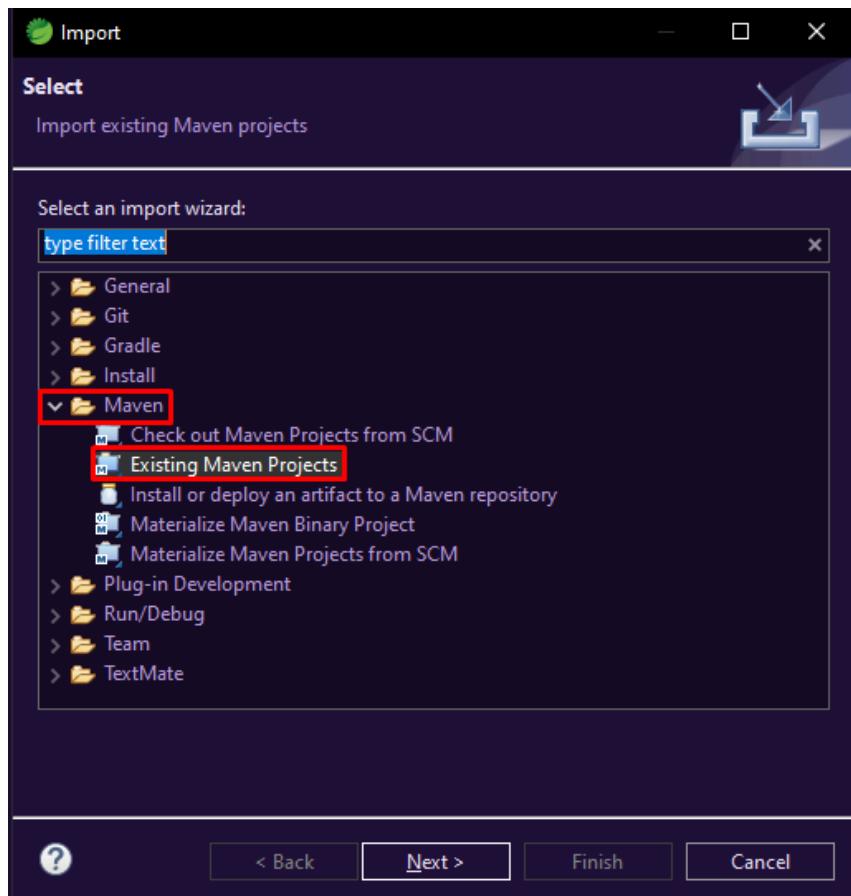
Elegimos la ruta donde se colocar el JAR, lo ideal es dentro de la carpeta sts-4.20.1.RELEASE, seleccionas el archivo SpringToolSuite4.exe (es posible que te salga sin la extension .exe entonces eliges el que no es .ini) y procedemos a instalarlo.



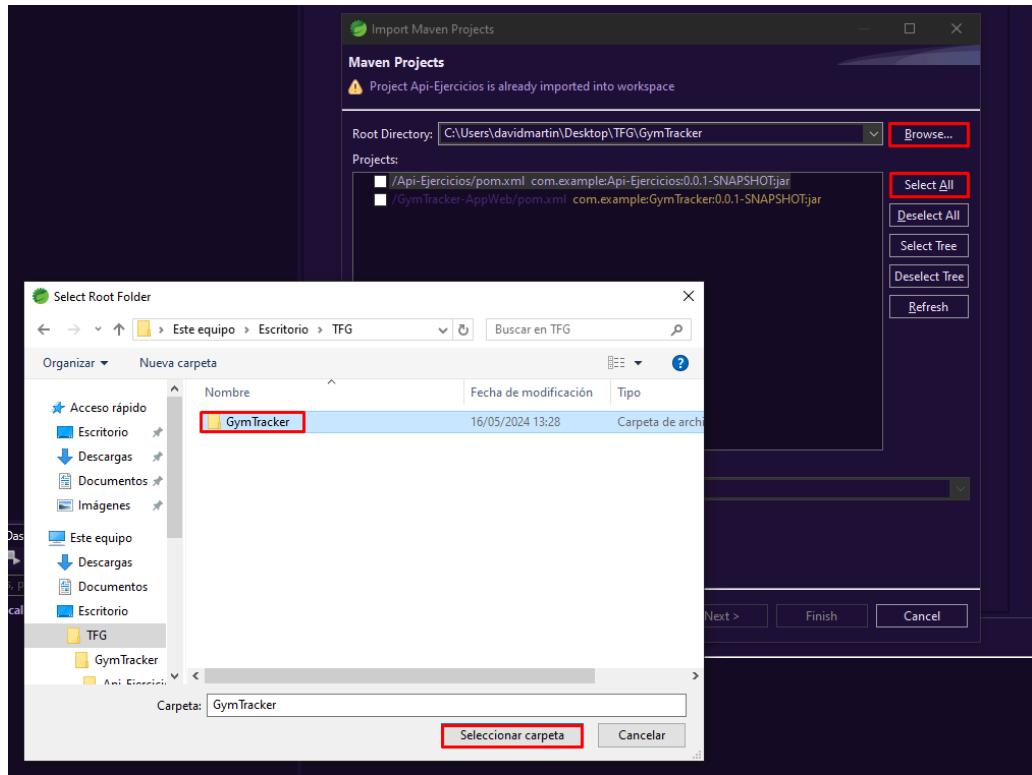
Una vez instalado lombok, iniciamos nuestro STS y lo primero que haremos será hacer click en **file, import**.



Se nos abrirá una nueva pestaña en la que seleccionaremos **maven, Existing Maven Projects**.

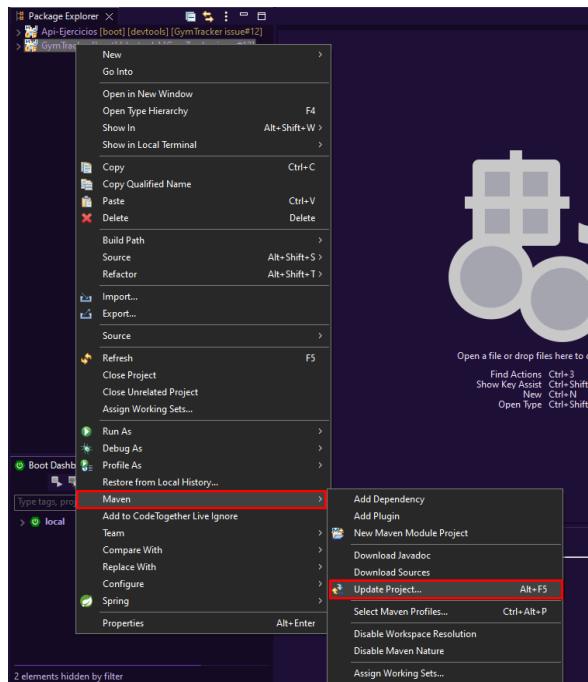


Nos abrirá otra nueva ventana en la buscaremos nuestro proyecto clonado de GitHub. Para ellos pulsamos en **Browse**, nos abrirá una ventana en la que buscaremos nuestra carpeta donde hemos clonado el proyecto. Una vez encontrado pulsamos en **seleccionar carpeta**. En la ventana que nos queda abierta pulsamos **Select All** y **Finish**.



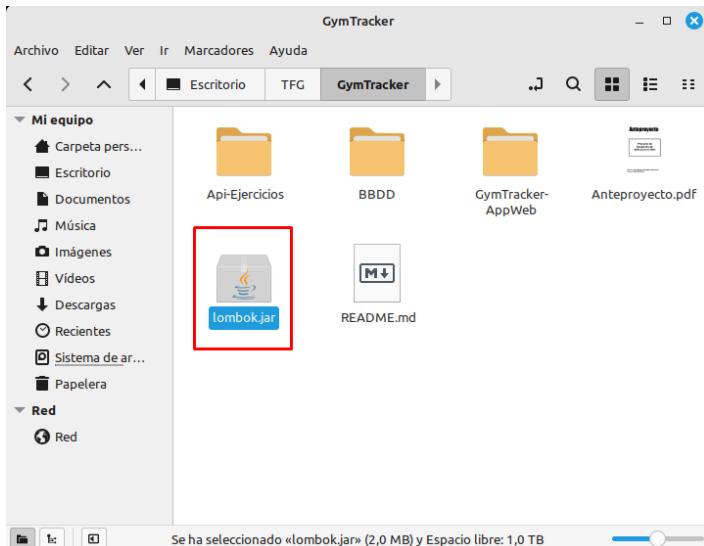
Por último, una vez nos aparezcan los dos proyectos, tenemos que hacer el siguiente paso en los dos.

Hacemos click derecho, vamos a **Maven, Update Project**. Esto debería actualizar el proyecto para que no de problemas con Lombok.



Linux

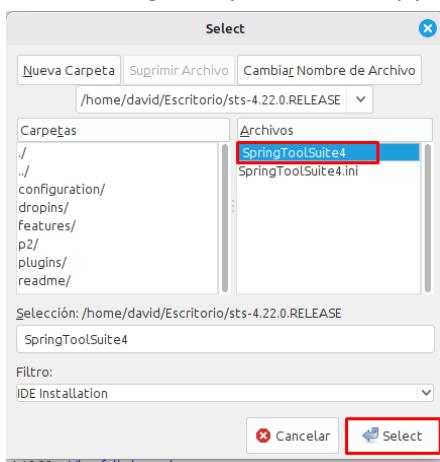
Una vez que la clonación sea exitosa, primero vamos a instalar **lombok** (Biblioteca Java diseñada para minimizar el código repetitivo que se escribe en las clases Java.). Ahora ya podemos ejecutar la aplicación desde SpringBoot.



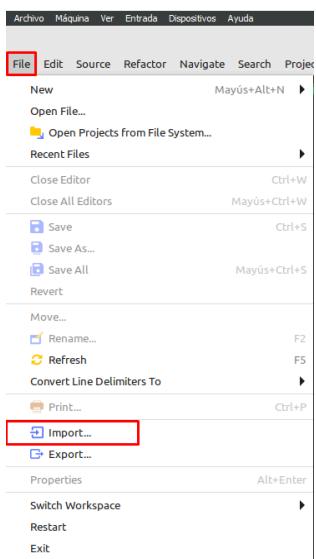
Abrimos una terminal en la carpeta que tenemos el archivo lombok y escribimos el siguiente comando para darle los permisos y pueda ejecutarse. **sudo chmod +x lombok.jar**

```
david@david-VirtualBox:~/Escritorio/TFG/GymTracker
Archivo Editar Ver Buscar Terminal Ayuda
david@david-VirtualBox:~/Escritorio/TFG/GymTracker$ sudo chmod +x lombok.jar
[sudo] contraseña para david:
david@david-VirtualBox:~/Escritorio/TFG/GymTracker$
```

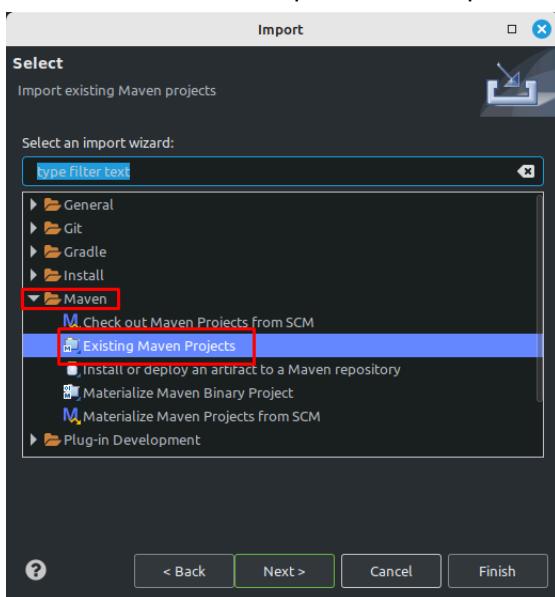
Elegimos la ruta donde se colocar el JAR, lo ideal es dentro de la carpeta sts-4.20.1.RELEASE, seleccionas el archivo SpringToolSuite4.exe (es posible que te salga sin la extensión .exe entonces eliges el que no es .ini) y procedemos a instalarlo.



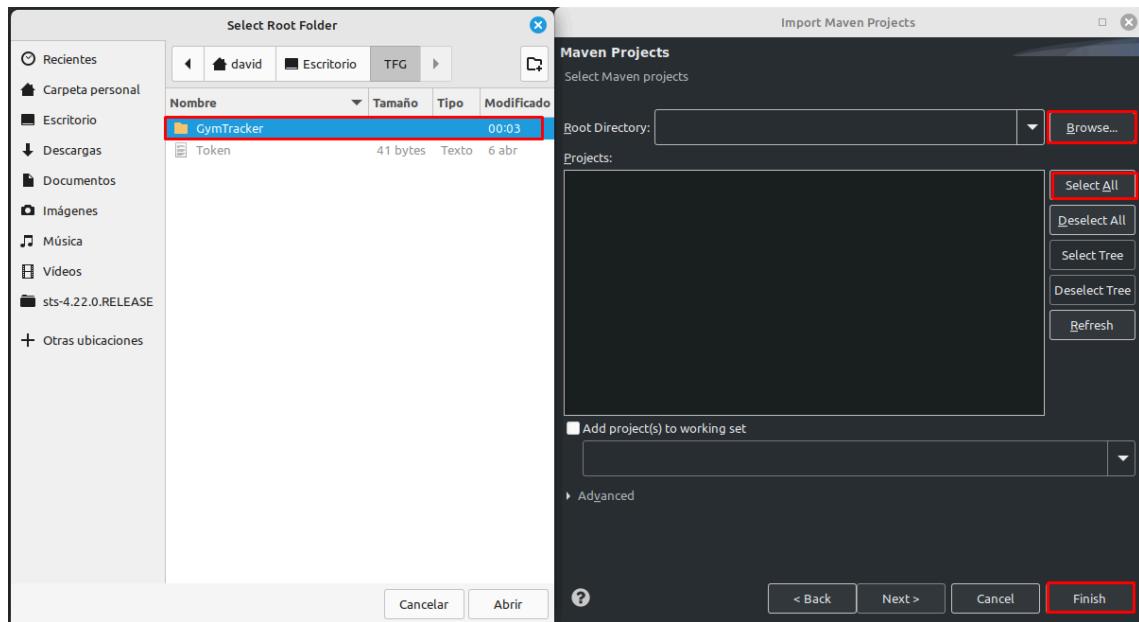
Una vez instalado lombok, iniciamos nuestro STS y lo primero que haremos será hacer click en **file, import**.



Se nos abrirá una nueva pestaña en la que seleccionaremos **maven, Existing Maven Projects**.

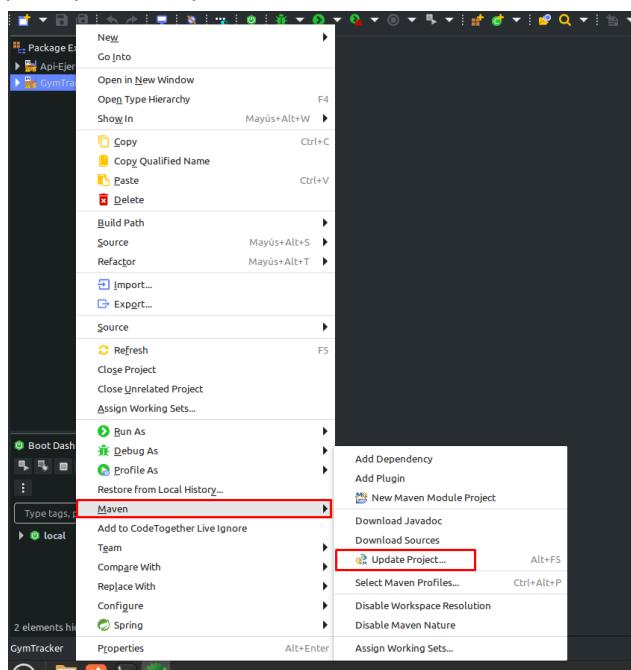


Nos abrirá otra nueva ventana en la buscaremos nuestro proyecto clonado de GitHub. Para ellos pulsamos en **Browse**, nos abrirá una ventana en la que buscaremos nuestra carpeta donde hemos clonado el proyecto. Una vez encontrado pulsamos en **seleccionar carpeta**. En la ventana que nos queda abierta pulsamos **Select All** y **Finish**.



Por último, una vez nos aparezcan los dos proyectos, tenemos que hacer el siguiente paso en los dos.

Hacemos click derecho, vamos a **Maven, Update Project**. Esto debería actualizar el proyecto para que no de problemas con Lombok.

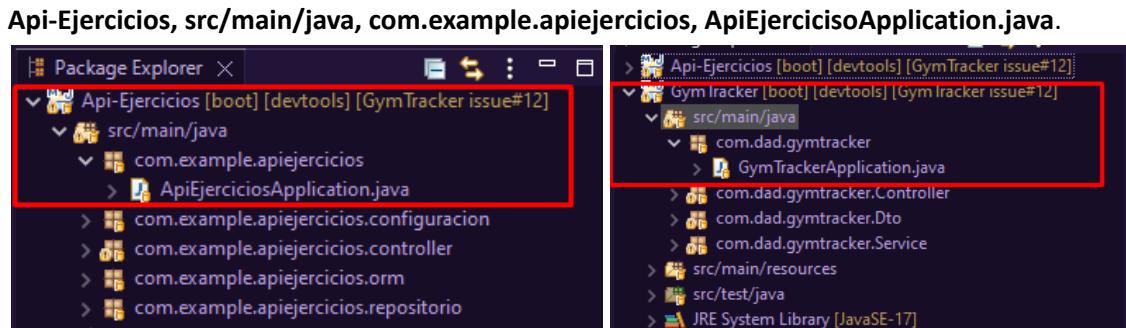


7.3.5 Acceder a la Aplicación

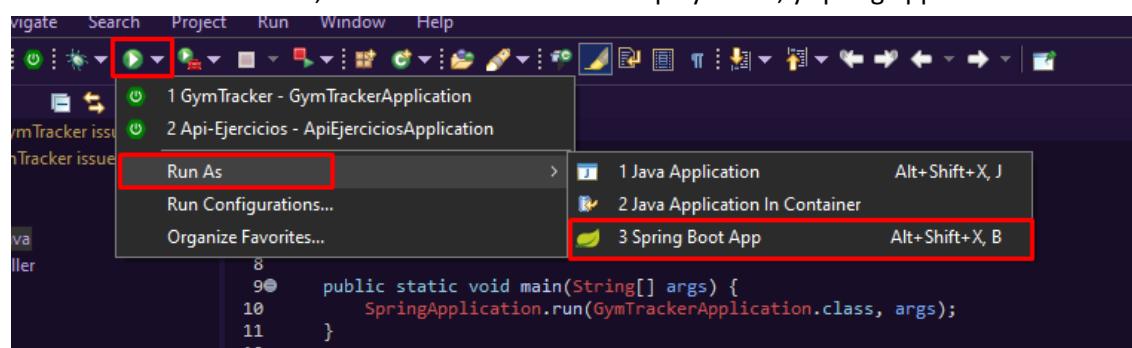
Una vez tengamos todos los pasos anteriores hechos y nuestro proyecto configurado en STS, ya podemos iniciar nuestra aplicación web.

Primero vamos a lanzar los dos proyectos. Debemos hacer el mismo paso en ambos proyectos.

Hacer doble click en **Gymtracker, src/main/java, com.dad.GymtrackerApplication.java**.

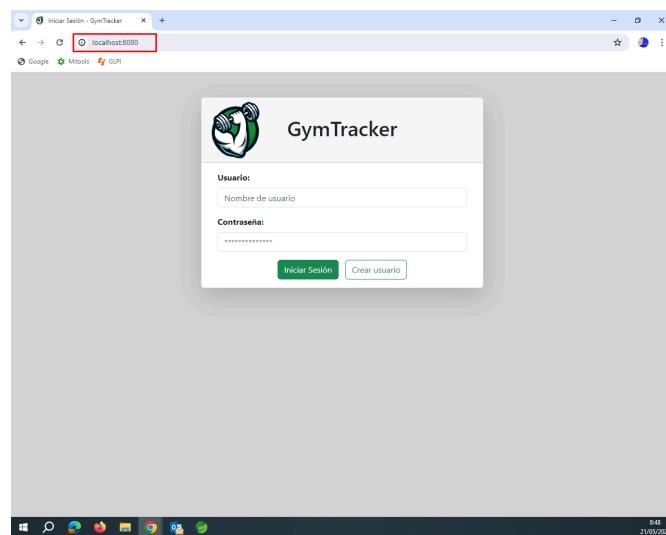


Una vez abierto el archivo, clickaremos en el botón de play verde, y spring app.



Puedes acceder a la aplicación abriendo un navegador web y visitando la siguiente URL:

<https://localhost>



7.4 Despliegue en local desde Github

7.4.1 Descarga de los archivos JAR

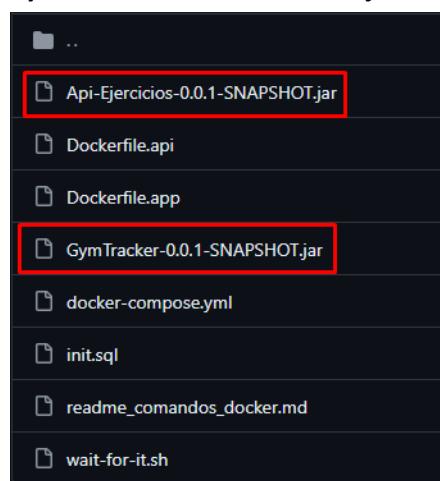
7.4.1.1 Linux:

Lo primero es descargar los archivos JAR desde el siguiente enlace de GitHub.

<https://github.com/deiivvv/GymTracker/tree/main/docker>

Descargaremos los archivos **Api-Ejercicios-0.0.1-SNAPSHOT.jar** y

GymTracker-0.0.1-SNAPSHOT.jar



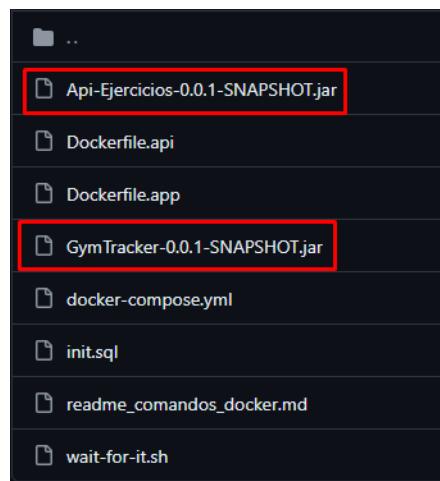
7.4.1.2 Windows:

Lo primero es descargar los archivos JAR desde el siguiente enlace de GitHub.

<https://github.com/deiivvv/GymTracker/tree/main/docker>

Descargaremos los archivos **Api-Ejercicios-0.0.1-SNAPSHOT.jar** y

GymTracker-0.0.1-SNAPSHOT.jar



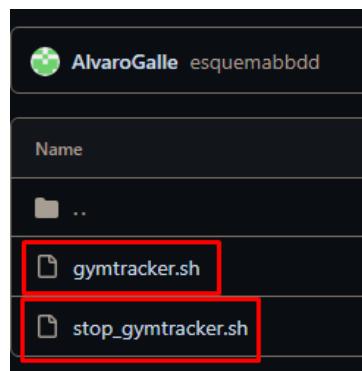
7.4.2 Preparación para el despliegue

Para preparar el despliegue de nuestra aplicación en local comenzamos por descargar un script que va a ejecutar automáticamente nuestros archivos JAR.

Para ello entraremos al siguiente enlace y nos descargamos los dos archivos que hay:

gymtracker.sh y **stop_gymtracker.sh**.

<https://github.com/dejivv/GymTracker/tree/main/aws/Scripts>



Lo más cómodo será guardar tanto los JAR como los scripts en una carpeta para no perderlos.

Este equipo > Escritorio > JARs y Scripts				
Nombre	Fecha de modificación	Tipo	Tamaño	
Api-Ejercicios-0.0.1-SNAPSHOT.jar	03/06/2024 22:58	Archivo WinRAR	45.765 KB	
gymtracker.sh	03/06/2024 23:35	Archivo de origen ...	1 KB	
GymTracker-0.0.1-SNAPSHOT.jar	03/06/2024 22:58	Archivo WinRAR	60.112 KB	
stop_gymtracker.sh	03/06/2024 23:35	Archivo de origen ...	1 KB	

En segundo lugar deberemos instalar MySQL para instalar nuestra base de datos.

7.4.2.1 Windows:

1. Descargar MySQL Installer desde MySql

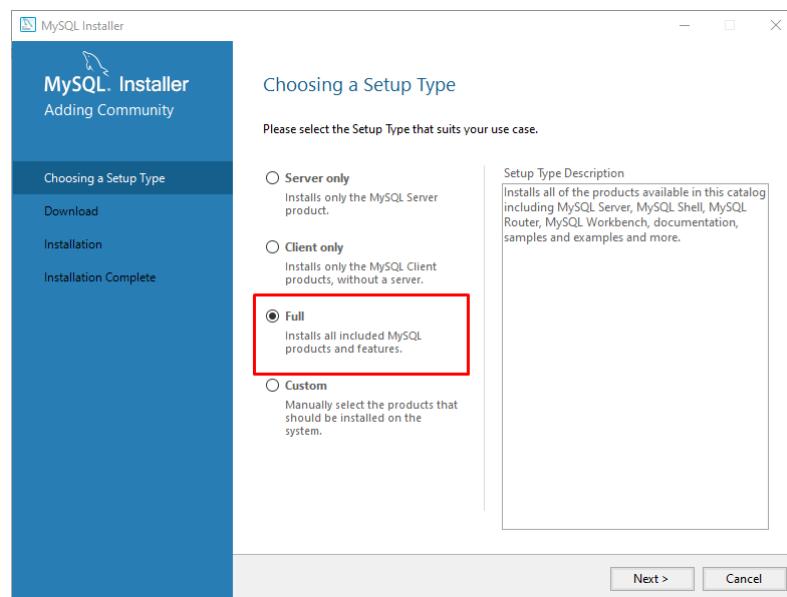
The screenshot shows the MySQL Community Downloads page. At the top, there are tabs for 'General Availability (GA) Releases' and 'Archives'. Below this, the MySQL Installer 8.0.37 page is displayed. It includes a note about MySQL 8.0 being the final series. A dropdown menu for 'Select Version' shows '8.0.37'. Another dropdown for 'Select Operating System' shows 'Microsoft Windows'. Two download links are listed: 'Windows (x86, 32-bit), MSI Installer' (mysql-installer-community-8.0.37.0.msi) and 'Windows (x86, 32-bit), MSI Installer' (mysql-installer-community-8.0.37.0.msi). Both links have 'Download' buttons. Below the links, a note says: 'We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.' The Oracle logo and copyright information are at the bottom.

2. Ejecutar el instalador:

- Abre el archivo descargado.

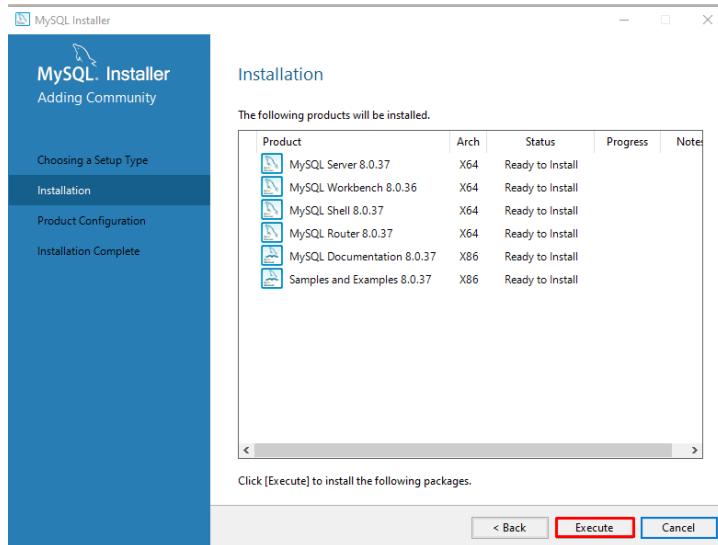
3. Seleccionar el tipo de instalación.

- Elige "Full" para una instalación completa.
- Haz clic en "Next".



4. Instalar las dependencias.

- El instalador verificará y descargará las dependencias necesarias.
- Haz clic en "Execute" para proceder.

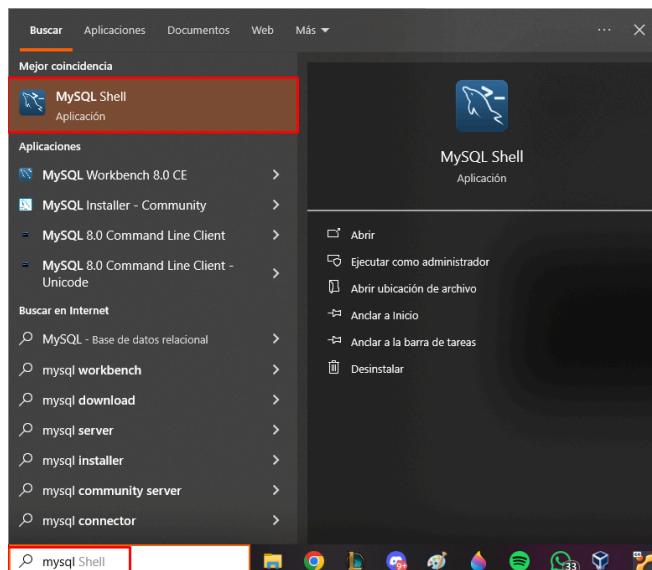


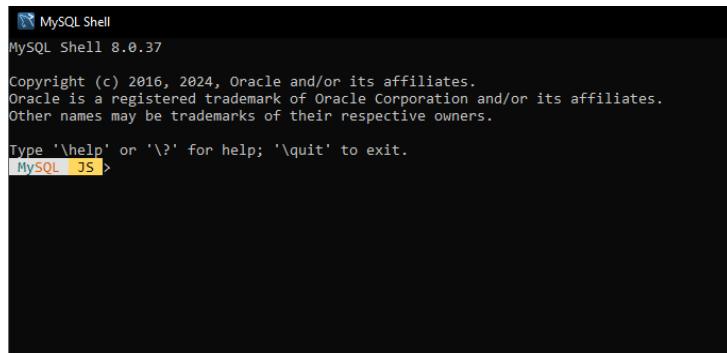
5. Configuración del servidor MySQL.

- Elige el tipo de configuración (General o Server Only).
- Configura el tipo y puerto de red (generalmente, puedes dejar las opciones predeterminadas).
- Establece una contraseña para el usuario `root` y, opcionalmente, crea usuarios adicionales.
- Configura el servicio de Windows si deseas que MySQL se inicie automáticamente con el sistema.

6. Finalizar la instalación.

- Una vez finalizada la instalación, puedes abrir MySQL Workbench o MySQL Shell para empezar a trabajar con MySQL.





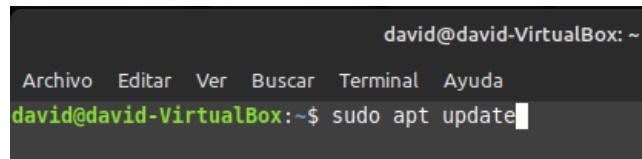
MySQL Shell
MySQL Shell 8.0.37
Copyright (c) 2016, 2024, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.
Type '\help' or '\?' for help; '\quit' to exit.
MySQL>

7.4.2.1 Linux:

Para instalar MySQL en Linux (Debian)

1. Actualizar los repositorios:

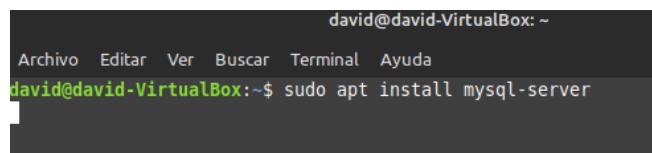
```
sudo apt update
```



david@david-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
david@david-VirtualBox:~\$ sudo apt update

2. Instalar MySQL:

```
sudo apt install mysql-server
```



david@david-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
david@david-VirtualBox:~\$ sudo apt install mysql-server

3. Iniciamos MySQL para comprobar que funciona:

```
sudo mysql
```

```

david@david-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
david@david-VirtualBox:~$ sudo mysql
[sudo] contraseña para david:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

Estos pasos deberían permitirte instalar y configurar MySQL en tu sistema Linux.

7.4.2.1.1 Clonar repositorio de Github para windows

En una terminal de powershell (pulsamos botón de windows y escribimos powershell), nos posicionamos en el directorio donde queremos que se clone el repositorio, podemos movernos con el comando **cd**, una vez estemos situados en nuestro destino, ejecutamos el siguiente comando para clonar el repositorio:

git clone https://github.com/tu-usuario/GymTracker.git

Reemplaza `tu-usuario` con el nombre de tu usuario en GitHub.

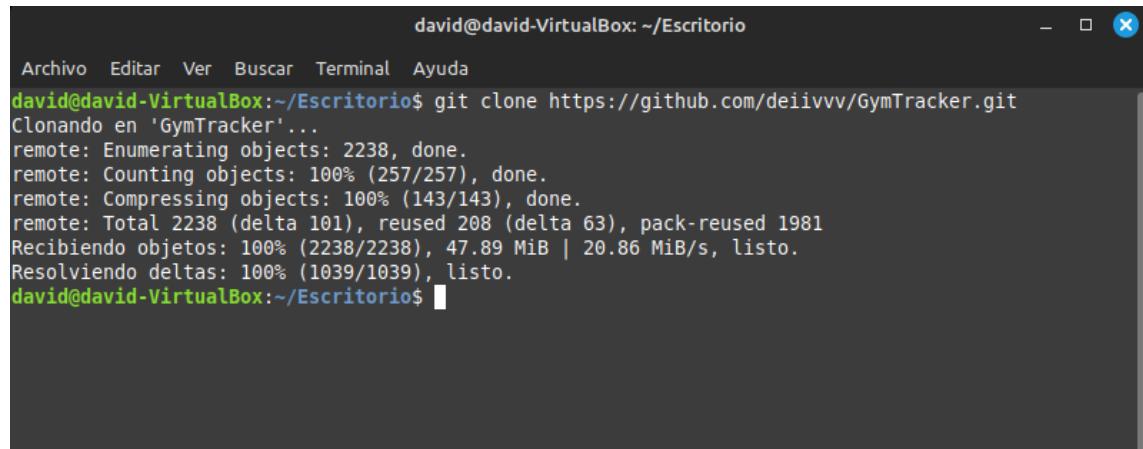
PS C:\Users\davidmartin\Desktop> cd TFG
PS C:\Users\davidmartin\Desktop\TFG> git clone https://github.com/tu-usuario/GymTracker.git
>> -

7.4.2.1.2 Clonar repositorio de GitHub para linux

Abrimos una terminal, nos posicionamos en el directorio donde queremos que se clone el repositorio, podemos movernos con el comando **cd**, una vez estemos situados en nuestro destino, ejecutamos el siguiente comando para clonar el repositorio:

```
git clone https://github.com/tu-usuario/GymTracker.git
```

Reemplaza `tu-usuario` con el nombre de tu usuario en GitHub.

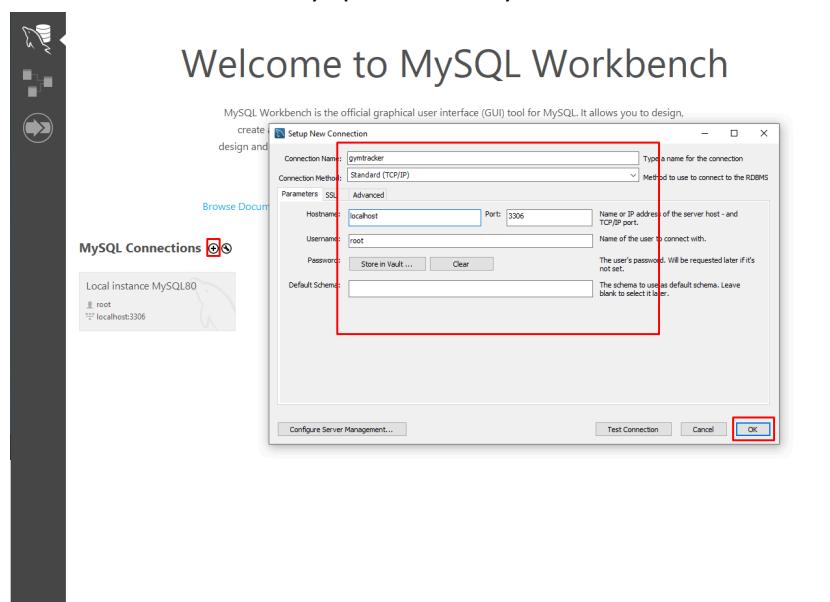


```
david@david-VirtualBox: ~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
david@david-VirtualBox:~/Escritorio$ git clone https://github.com/deiivv/GymTracker.git
Clonando en 'GymTracker'...
remote: Enumerating objects: 2238, done.
remote: Counting objects: 100% (257/257), done.
remote: Compressing objects: 100% (143/143), done.
remote: Total 2238 (delta 101), reused 208 (delta 63), pack-reused 1981
Recibiendo objetos: 100% (2238/2238), 47.89 MiB | 20.86 MiB/s, listo.
Resolviendo deltas: 100% (1039/1039), listo.
david@david-VirtualBox:~/Escritorio$
```

7.4.2.2.1 Volcar base de datos en windows

Abrimos la carpeta clonada de Github, entramos en la carpeta BBDD ya abrimos el archivo Gymtracker.sql y copiamos todo el contenido.

Ahora abrimos nuestro mysql workbench y creamos la conexión.



Pegamos el contenido en nuestro script y ya tendríamos nuestra base de datos creada.

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'gymtracker' is selected. The 'Schemas' pane on the left lists 'sakila', 'sys', and 'world'. The 'Query 1' pane contains the SQL script for creating the database and its tables. The 'Output' pane at the bottom shows the execution log with several successful insert operations. A red box highlights the 'Action Output' section of the log.

```
1 • DROP DATABASE IF EXISTS gymtracker;
2 • CREATE DATABASE gymtracker;
3 • 
4 • DROP USER IF EXISTS 'dad'@'localhost';
5 • CREATE USER 'dad'@'localhost' IDENTIFIED BY 'padre';
6 • GRANT ALL PRIVILEGES ON gymtracker.* TO 'dad'@'localhost';
7 • 
8 • USE gymtracker;
9 • 
10 • CREATE TABLE usuarios (
11     id INT AUTO_INCREMENT,
12     nombre VARCHAR(100) UNIQUE,
13     contrasena VARCHAR(100),
14     rol VARCHAR(50),
15     PRIMARY KEY (id)
16 );
17 )
18 
19 • CREATE TABLE perfil (
20     id_usuario INT AUTO_INCREMENT,
21     genero VARCHAR(100),
22     edad INT,
23     altura FLOAT,
24     peso FLOAT,
25     FOREIGN KEY (id_usuario) REFERENCES usuarios(id)
26     ON DELETE CASCADE
27 );
28 
29 • CREATE TABLE rutinas /
```

#	Time	Action	Message	Duration / Fetch
48	14:16:42	INSERT INTO series (peso, repes) VALUES (100, 1)	1 row(s) affected	0.000 sec
49	14:16:42	SET @id_serie_press_banca := LAST_INSERT_ID()	0 rows(s) affected	0.000 sec
50	14:16:42	INSERT INTO rutinas (id_usuario, nombre, fecha) VALUES ((SELECT id FROM usuarios WHERE id = 1), 'press banca', '2024-01-01')	1 row(s) affected	0.000 sec
51	14:16:42	SET @id_rutina := LAST_INSERT_ID()	0 rows(s) affected	0.000 sec
52	14:16:42	INSERT INTO rutinas_ejercicios_series (id_rutina, id_ejercicio, id_serie) VALUES (@id_rutina, 1, @id_serie_press_banca)	2 rows(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.000 sec
53	14:16:42	INSERT INTO rutinas_ejercicios_series (id_rutina, id_ejercicio, id_serie) VALUES (@id_rutina, 2, @id_serie_press_banca)	2 rows(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.000 sec

7.4.2.2 Volcar base de datos en linux

Abrimos la carpeta clonada de Github, entramos en la carpeta BBDD y abrimos el archivo Gymtracker.sql y copiamos todo el contenido.

En una terminal iniciamos mysql y pegamos todo el contenido.

The terminal session shows the creation of the 'gymtracker' database and its tables. It starts with the MySQL prompt, then executes the 'Gymtracker.sql' script. The output shows the creation of the database, the creation of the 'usuarios' and 'perfil' tables, and the creation of the 'rutinas' table. It also shows the insertion of data into the 'series' table and the creation of the 'rutinas_ejercicios_series' table.

```
tfg@tfg-VirtualBox:~$ sudo mysql;
[sudo] contraseña para tfg:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> DROP DATABASE IF EXISTS gymtracker;
routine, 33, @id_serie_press_bancaQuery OK, 6 rows affected (0,34 sec)

mysql> CREATE DATABASE gymtracker;
routine, 33, @iQuery OK, 1 row affected (0,01 sec)

mysql>
mysql> USE gymtracker;
d_serie_press_bancaDatabase changed
mysql>
mysql> CREATE TABLE usuarios (
->     id INT AUTO_INCREMENT,
->     nombre VARCHAR(100) UNIQUE,
->     contrasena VARCHAR(100),
```

Ya tendríamos nuestra base de datos creada.

7.4.3 Despliegue

7.4.3.1 Despliegue en windows:

Por último vamos a ejecutar los JAR para que nuestra app funcione, para ello hay que abrir dos terminales de powershell como administrador, haciendo click derecho + ejecutar como administrador. Una vez dentro vamos a navegar con el comando cd hasta nuestra carpeta donde se encuentra el JAR.

Por último vamos a ejecutar el comando java -jar [nombre del jar] y ya habríamos acabado.

Vamos a comprobar que todo va bien yendo al navegador y escribiendo nuestra url.

<https://localhost>

8. Pruebas (!IMPORTANT puntúa bien)

Suceso	Supuesto	Resultado	Solución	Conclusión
Conexión a la base de datos	Queremos consumir la tabla "usuarios" de la base de datos.	Error al conectarse a la base de datos y la aplicación no se levanta correctamente.	Configurar la URL y las credenciales de la base de datos en el archivo application.properties.	Es imprescindible configurar correctamente los detalles de la conexión a la base de datos en el archivo application.properties para que una aplicación de Spring Boot funcione correctamente.
Crear nuevo usuario en la aplicación	El usuario se guarda correctamente en la base de datos y también se crea como usuario de Spring Security con acceso	Los usuarios se guardan en la base de datos correctamente, pero no se crean automáticamente como usuarios de Spring Security con acceso.	Usar InMemoryUserDetailsManager que permita actualizar los usuarios de Spring Security.	Al combinar JPA e InMemoryUser Details Manager, podemos gestionar tanto la persistencia de los usuarios en la base de datos como su integración en Spring Security.
Implementar un sidebar dinámico en la aplicación web.	El sidebar debe cambiar su contenido o posición dependiendo de la pantalla en la que te encuentres.	El sidebar funciona correctamente.	Uso de javascript para cambiar los estilos del sidebar.	Un sidebar dinámico mejora la usabilidad y la experiencia del usuario al proporcionar contenido relevante según el contexto.
Acceso a la aplicación mediante login	Todos los usuarios podrán acceder a la aplicación a través de un login excepto los que tengan el rol de "bloqueado".	Todos los usuarios acceden sin excepciones.	Comprobar el rol de los usuarios y capturar el rol "bloqueado" para denegar el acceso..	La autenticación debe verificar y bloquear a los usuarios con el rol "bloqueado" para que no puedan acceder a la aplicación.
Consumo de la	Se deben listar y	No se encuentran	Crear un archivo	Crear un archivo

API para el listado y creación de ejercicios.	crear los ejercicios consumidos de la API.	ejercicios. No hay conexión con la api	de configuración y el método addCorsMappings para regular las direcciones y métodos de acceso.	de configuración y el método addCorsMappings para regular las direcciones y métodos de acceso.
Verificar el acceso a la aplicación de usuarios con el mismo nombre de usuario.	Verificar el acceso a la aplicación de usuarios con el mismo nombre de usuario.	Problemas al reconocer la contraseña de cada usuario. .	Exclusividad del nombre de usuario al crear un nuevo usuario.	Asegurar que el nombre de usuario sea único al crear un nuevo usuario para evitar problemas de autenticación.
Guardado de rutinas y ejercicios en la base de datos	Tras crear la rutina se guarda en la base de datos junto con los ejercicios y series de esta	Problemas al guardarlo y luego tratar de recuperarlo	Modificaciones en la estructura de la base de datos. tabla intermedia de relación ternaria	La importancia de una estructura de base de datos bien diseñada y adaptable para el correcto funcionamiento de la aplicación.
Editar perfil	Actualizar los cambios dentro de la base de datos	Los datos se actualizan correctamente.	Consulta la base de datos con la sentencia UPDATE	Las consultas de actualización deben ser precisas para mantener la integridad de los datos en la aplicación.
Hashear contraseñas	Almacenar las contraseñas hasheadas y comprobarlas en el login	Las funciones de javaScript permiten hashear pero no logramos hacer comprobaciones	Hashear las contraseñas a través del encriptado de Spring Security	Spring Security proporciona mecanismos seguros para hashear y verificar contraseñas.
Acceder a los perfiles como administrador	Poder acceder a los perfiles los usuarios sin saber sus contraseñas	El la sesión no cambia	Recuperar el id de usuario al que queremos acceder	Implementar un mecanismo para que el administrador pueda cambiar temporalmente el contexto de seguridad y actuar como otro

				usuario.
Crear gráfica de barras	Se creará un diagrama de barras con los datos devueltos por la petición axios	El java script no funciona como se esperaba	Uso de una librería de javaScript, Chart.	El uso de la librería Chart.js permitió la creación exitosa de un diagrama de barras a partir de los datos obtenidos. Esto resalta la utilidad de las librerías de JavaScript para el desarrollo.
Implementar funcionalidad de cierre de sesión.	Tras cerrar sesión, el usuario no debe poder volver a acceder a las pantallas de la aplicación.	Spring Security elimina la sesión y restringe el acceso.	Configurar Spring Security para manejar el cierre de sesión.	El cierre de sesión elimina la sesión del usuario y restringe el acceso a las pantallas de la aplicación, asegurando que solo los usuarios autenticados puedan acceder.

9. Conclusiones

10. Bibliografía

10.1 Enlaces de descarga

10.1.1 Oracle JDK



<https://www.oracle.com/java/technologies/downloads>

10.1.2 OpenJDK



<https://jdk.java.net>

10.1.3 Git

<https://git-scm.com/download/win>



10.1.4 Spring Tool Suite

<https://spring.io/tools>



10.1.5 MySql

<https://dev.mysql.com/downloads/installer>



10.1.6 Lombok

<https://projectlombok.org/download>



10.2 Documentación Oficial



10.2.1 Spring Boot

<https://spring.io/projects/spring-boot>



10.2.2 Java

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



10.2.3 MySql

<https://dev.mysql.com/doc/>



10.2.4 JavaScript

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>



10.2.5 Axios

<https://axios-http.com/docs/intro>



10.2.6 Bootstrap

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>



10.2.7 HTML

<https://developer.mozilla.org/en-US/docs/Web/HTML>



10.2.8 CSS

<https://developer.mozilla.org/en-US/docs/Web/CSS>



10.2.9 AWS

<https://docs.aws.amazon.com/>



10.2.10 Docker

<https://docs.docker.com/>



10.2.11 Git

<https://www.git-scm.com/doc>



10.2.12 Lombok

<https://projectlombok.org/features/>



10.2.13 Apache Tomcat

<https://tomcat.apache.org/>

10.3 Otras páginas de consulta



10.3.1 Stack Overflow

<https://stackoverflow.com/>



10.3.2 W3Schools

[schools](https://www.w3schools.com/)

<https://www.w3schools.com/>



10.3.3 ChatGPT

<https://chatgpt.com/>



10.3.4 Microsoft designer

<https://designer.microsoft.com/>

10.4 Páginas importantes en el desarrollo



10.4.1 Duck DNS

<https://www.duckdns.org/>



10.4.2 AWS Academy

<https://awsacademy.instructure.com/>



10.4.3 Github

<https://github.com/>

10.5 Enlaces para el despliegue en Docker

10.5.1 Tutorial para instalar docker

<https://www.youtube.com/watch?v=ZO4KWQfUBBc>

10.5.2 Instalar WSL

<https://learn.microsoft.com/en-us/windows/wsl/install>

10.5.3 Wait-for-it

<https://github.com/vishnubob/wait-for-it>

10.5.4 Instalar docker

<https://docs.docker.com/desktop/install/windows-install/>