

DREAMGYM

CFGS DAW

CURSO 2022/2023

RODRIGO

LÓPEZ SÁNCHEZ

COLEGIO CALASANZ SALAMANCA



1. Introducción	3
2. Descripción de la aplicación	4
3. Plan de Empresa	5
4. Tecnologías escogidas y justificación	6
5. Diseño de la aplicación	7
5.1. Diagramas y definición de casos de uso	7
5.2. Diagramas de clases.....	7
5.3. Modelo entidad-relación	7
6. Arquitectura de la aplicación	8
6.1. Estructura del proyecto	8
6.2. Librerías externas utilizadas.....	8
7. Manual de despliegue.....	9



1. Introducción

La aplicación web “DREAMGYM” es una aplicación creada para poder informatizar la inscripción a un gimnasio, dado que hoy en día muchos gimnasios no ofrecen inscribirse en los mismos a través de internet, y/o, peor aún, no ofrecer contenido fotográfico para así mostrar como es el gimnasio ni los servicios que ofrece el mismo.

Una aplicación web como DREAMGYM puede renovar la imagen de un gimnasio en su totalidad y atraer a más clientes gracias a un aspecto profesional y competente con el resto de gimnasios.

El objetivo del proyecto es crear una aplicación web lo más atractiva e intuitiva posible, a la par que ofrecer una herramienta valiosa y diferenciadora para los clientes del mismo.



2. Descripción de la aplicación

La aplicación web tendrá las siguientes funciones:

- Página de inicio llamativa para recibir al posible cliente con cierta profesionalidad y cercanía.
- Página de inscripción para usuarios ya registrados (clientes del gimnasio) y zona de inscripción para nuevos clientes.
- Página donde el usuario puede realizar login con su usuario y contraseña una vez registrado.
- Página donde el usuario puede ver las fotos del gimnasio. (Galería)
- Página de precios a seleccionar para que el cliente sienta que el gimnasio es accesible para cualquier perfil. Se calculará su edad y se le mostrará el descuento que se le va a realizar además del precio final que pagará en función de su edad. 16-30 años 15%, 30-60 años 5%, 60-80 años 40% y 80-100 años 60%.
- Página de noticias interesantes para los más curiosos, artículos sobre nutrición, realizar ejercicios correctamente y/o beneficios de la actividad deportiva.
- Página donde el usuario una vez registrado y logueado pueda ver qué ejercicios podría hacer hoy en el gimnasio. (Sugerencias de ejercicios)
- Página donde el usuario una vez registrado y logueado pueda consultar las clases que se ofrecen durante la semana en el gimnasio. (Calendario)
- Panel de administrador donde un usuario con el rol de administrador pueda ver los clientes y borrar todo lo relacionado con los mismos, es decir, borrar todas las suscripciones y compras que haya realizado el cliente.



3. Plan de Empresa

Nombre del negocio: Dreamgym

Misión: Brindar una plataforma en línea para un gimnasio que ofrezca servicios de entrenamiento personal, clases en línea y nutrición.

Objetivo: Aumentar el alcance del gimnasio, generando ingresos a través de membresías y servicios en línea.

Análisis de mercado:

Identificación del público objetivo: Tanto personas las cuales estén interesadas en la actividad física, la salud y el bienestar, como las personas que estén interesadas en dar un cambio en su vida y en su cuerpo.

Competencia: Buscar puntos de diferenciación a través de investigar a los gimnasios y servicios de entrenamiento en línea.

Tendencias de mercado: Analizar las tendencias en el mercado del fitness y la actividad física para adaptar los servicios ofrecidos a las necesidades del público objetivo.

Plan de Marketing:

Estrategia: Utilizar el marketing digital para llegar a los clientes potenciales y aumentar la presencia en línea del gimnasio.

Canales de Marketing: Publicidad en Google, redes sociales (Facebook, Instagram), publicidad en línea, emails con marketing.

Presencia en línea: Creación y gestión de redes sociales, blog del gimnasio, SEO en el sitio web.

Plan de operaciones:

Infraestructura: Contratar a un equipo de desarrollo web para mantener la página web y hacer cambios según las nuevas tendencias.

Servicios: Ofrecer entrenamiento personal en línea, clases en línea y planes de nutrición personalizados.

Plataformas de pago: Integrar las opciones de pago en línea para los servicios ofrecidos.

Soporte al cliente: Ofrecer soporte al cliente a través de la página web y las redes sociales.

Finanzas:

Ingresos: Generados a través de las membresías en línea y los servicios ofrecidos.

Costos: Inversión en el desarrollo web, marketing digital y pago a los entrenadores en línea.

Proyección de ingresos: Establecer una proyección de ingresos basada en el número de miembros y servicios adquiridos.

Retorno de inversión: Realizar un análisis de retorno de inversión para determinar el éxito financiero del proyecto.



4. Tecnologías escogidas y justificación

VUE CLI (parte **frontend**): Vue CLI es un framework de JavaScript muy popular para la creación de aplicaciones web interactivas y dinámicas. Vue CLI es una herramienta que te permite crear y configurar proyectos Vue de manera rápida y sencilla. Su facilidad de uso, su documentación detallada y su amplia comunidad de desarrolladores son algunas de las razones por las que Vue CLI es una elección sólida para el desarrollo de la parte del front-end de DreamGym.

Vue Router(parte **frontend**): Vue Router es un complemento de Vue que te permite crear y gestionar rutas en tu aplicación web. Con Vue Router, puedes crear una estructura de navegación coherente y fácil de usar para los usuarios de tu página web. Además, Vue Router es fácil de integrar con otras herramientas y frameworks, lo que lo convierte en una opción conveniente para el desarrollo del front-end de tu proyecto.

Spring Data JPA(parte **backend**): Spring Data JPA es una herramienta de Java que se utiliza para simplificar la creación de aplicaciones basadas en bases de datos. Te permite trabajar con bases de datos relacionales de manera más fácil y eficiente, ya que automatiza gran parte del código que de otra manera tendrías que escribir manualmente. Además, Spring Data JPA es compatible con una amplia variedad de bases de datos, lo que lo convierte en una herramienta versátil para el desarrollo del back-end de DreamGym, por si en algún momento se planteara utilizar otro sistema de BBDD.

1. **Spring Security JWT** (parte **backend**): Spring Security es un framework de seguridad de Java que se utiliza para proteger las aplicaciones web de posibles ataques. JWT (JSON Web Tokens) es una tecnología que se utiliza para la autenticación y autorización de usuarios en aplicaciones web. Spring Security JWT combina ambas tecnologías para crear un sistema de autenticación y autorización seguro y escalable para tu proyecto. En comparación con Spring Security normal, JWT ofrece una mayor flexibilidad y eficiencia, lo que lo convierte en una opción adecuada para la gestión de la seguridad de DreamGym. Hay varias razones por las cuales se ha utilizado Spring JWT frente a Spring Security (Con sesiones):
Escalabilidad: Spring Security JWT es más escalable que Spring Security convencional, especialmente en entornos distribuidos o en la nube. Con JWT, el servidor no necesita almacenar la sesión de usuario, lo que lo hace más fácil de escalar horizontalmente sin problemas de rendimiento.
2. **Eficiencia:** En comparación con Spring Security convencional, JWT puede ser más eficiente en términos de uso de recursos. Con JWT, no se requiere consultar la base de datos en cada solicitud para validar la sesión de usuario, lo que puede mejorar significativamente el rendimiento de la aplicación.



3. **Flexibilidad:** JWT ofrece más flexibilidad que Spring Security convencional. Con JWT, es fácil incluir información adicional en el token, como roles de usuario o información personalizada, lo que puede ser útil en aplicaciones más complejas.
4. **Integración:** JWT se integra bien con otras tecnologías, como OAuth2 y OpenID Connect, lo que permite una autenticación y autorización más seguras y escalables en aplicaciones empresariales.
5. **Estandarización:** JWT es un estándar abierto y ampliamente aceptado para la autenticación y autorización de usuarios en aplicaciones web, lo que lo hace una opción sólida y compatible con otras herramientas y tecnologías.

MySQL (parte **backend**): MySQL o una amplia gama de características y funcionalidades para el almacenamiento y manipulación de datos. MySQL es de código abierto y cuenta con una amplia comunidad de desarrolladores, lo que garantiza su actualización y mantenimiento continuos.

5. Diseño de la aplicación

5.1. Diagramas y definición de casos de uso

5.2. Diagramas de clases

5.3. Modelo entidad-relación

Diagrama de casos de uso:

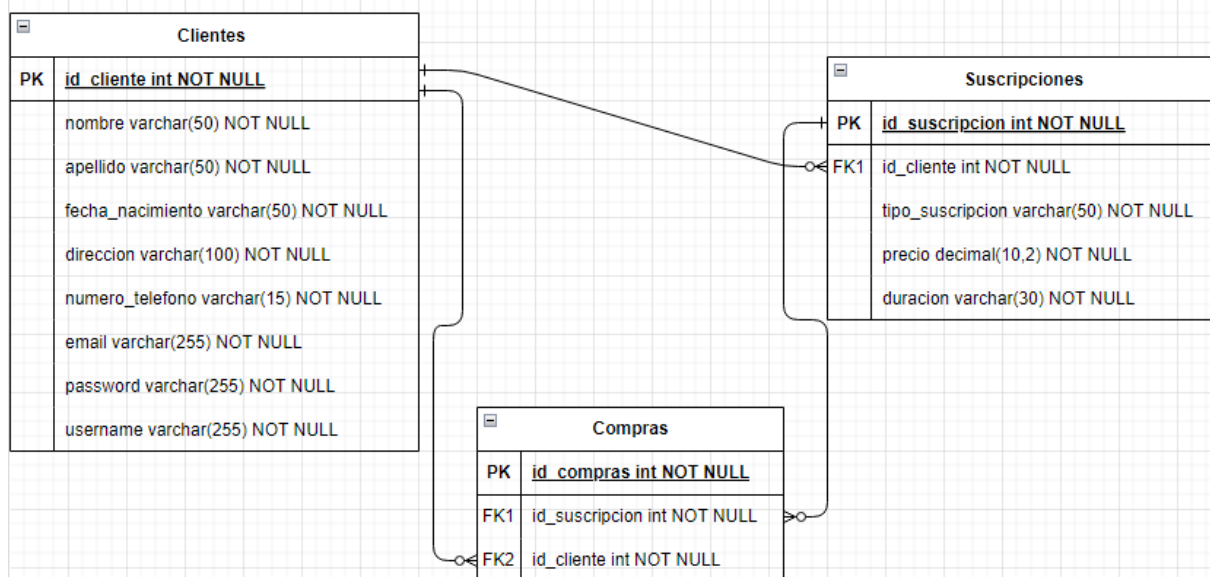


[illegible]

Para ver mejor el diagrama, se encuentra en la raíz de la carpeta “ProyectoFinalBack”, y, dentro de la carpeta “src/main/java” se encuentra el archivo .clic llamado “dreamgym-diagram.cld”, el cual se puede abrir en Eclipse para hacer zoom y verlo en detalle.



Diagrama entidad-relación:





6. Arquitectura de la aplicación

6.1. Estructura del proyecto

Con respecto a la estructura del proyecto, la parte backend se basa en un proyecto escrito en JAVA. Está realizada con el framework Spring Data JPA, donde se tiene un árbol de directorios propio de un proyecto de dichas características.

El proyecto está basado en el modelo-vista-controlador, en ese caso, la entidad **controlador** estaría representada por el código Java que maneja la lógica del negocio y se encarga de procesar las solicitudes entrantes y enviar respuestas apropiadas. El **modelo** estaría representado por la capa de persistencia, que se encarga de interactuar con la base de datos MySQL y manejar la información. Finalmente, la **vista** estaría representada por la parte del frontend, que en este caso se desarrolló utilizando Vue CLI. En la Vista, los usuarios interactúan con la interfaz gráfica de la aplicación para enviar solicitudes al Controlador. El Controlador luego procesa las solicitudes y actualiza el Modelo según sea necesario. La Vista se actualiza para mostrar los cambios en el Modelo, lo que completa el ciclo de la arquitectura MVC.

Con respecto a la parte frontend, está desarrollada con VUE CLI utilizando router para el mapping de cada vista.

La estructura básica de un proyecto con Vue CLI que utiliza el enrutador (router) consta de varios archivos y carpetas importantes. A continuación, describiré cada uno de ellos:

1. Carpeta "node_modules": Esta carpeta contiene todas las dependencias de Node.js que se utilizan en el proyecto, incluyendo Vue.js, Vue Router y otras bibliotecas.
2. Carpeta "public": Esta carpeta contiene archivos estáticos que se servirán directamente al navegador, como el archivo index.html, imágenes, hojas de estilo (CSS), etc.
3. Carpeta "src": Esta es la carpeta principal del proyecto y contiene los archivos de código fuente. Algunos de los archivos y carpetas importantes dentro de esta carpeta son:
 - Archivo "main.js": Este archivo es el punto de entrada principal de la aplicación y se utiliza para inicializar Vue.js y configurar las opciones globales.
 - Carpeta "assets": Esta carpeta contiene archivos como imágenes, iconos, fuentes, etc., que se utilizan en la aplicación.
 - Carpeta "components": Esta carpeta contiene los componentes de Vue.js, que son bloques de código reutilizables que se utilizan para construir la interfaz de usuario.
 - Carpeta "views": Esta carpeta contiene las vistas principales de la aplicación, que se utilizan para enrutar y mostrar diferentes páginas.
 - Carpeta "router": Esta carpeta contiene el archivo "index.js", que define las rutas y configuraciones del enrutador de Vue.js.
4. Archivo "package.json": Este archivo contiene la información del proyecto, así como las dependencias y los scripts de comandos para construir, ejecutar y depurar la aplicación.
5. Archivo "babel.config.js": Este archivo se utiliza para configurar Babel, una herramienta que se utiliza para compilar el código fuente de JavaScript a un formato que pueda ser entendido por todos los navegadores web modernos.
6. Archivo "vue.config.js": Este archivo se utiliza para configurar la compilación y el empaquetado del proyecto. Se pueden configurar opciones como el título de la aplicación, la URL base, la ruta de salida, etc.



6.2. Librerías externas utilizadas

En el proyecto se han utilizado diversas librerías para implementar diferentes funcionalidades tanto en la parte frontend como en la parte backend.

En la parte frontend, se ha utilizado la librería Axios para realizar peticiones HTTP al servidor. Además, para la maquetación de la página se ha utilizado en algunas vistas Bootstrap, que permite un diseño responsive y fácil adaptación a diferentes dispositivos. Para la decodificación de tokens JWT se ha utilizado la librería jwt-decode. Por último, se ha utilizado la librería date-fns para la manipulación de fechas.

En la parte backend, se ha utilizado la librería MySQL Connector para la conexión a la base de datos MySQL. Para el manejo de herramientas de desarrollo se ha utilizado Devtools. Para la implementación de la seguridad se ha utilizado la librería io.jsonwebtoken y Spring Security Web.



7. Manual de despliegue

Clonar el repositorio de git: <https://github.com/Roo55/ProyectoFinalDAW.git>

BBDD

Abrir el script de la bbdd y ejecutar línea a línea para evitar fallos.

FRONTEND

Abrir la carpeta “proyectofinalfront” con VSCode.

Abrir una consola e introducir por la línea de comandos:

“npm install”

“npm run serve”

BACKEND

Abrir eclipse y abrir el proyecto desde dicho programa.

Click derecho encima del proyecto > Run as > Spring Boot App