**INFORME DE TALLER PRÁCTICO EXPERIMENTAL**

|  |  |
| --- | --- |
| DATOS DE LA ASIGNATURA | |
| Nombre(s) del Estudiante(s): Sabine Vela, José Franco, Rosita Quezada. | **Nivel:**3ro |
| Carrera: Desarrollo de software. | **Docente:** Verónica Zapata. |
| Asignatura: Programación 3. | **Periodo académico:** |

1. **TEMA DEL TALLER**

 Desarrollar una aplicación web empresarial

1. **DESCRIPCIÓN DEL TALLER**

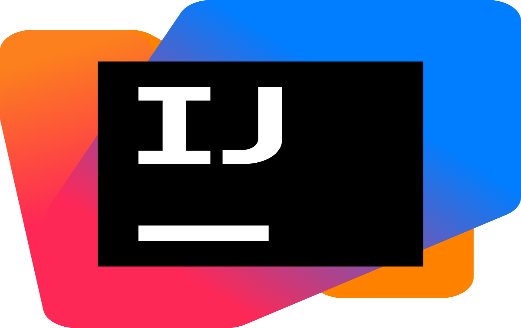
Elaboraruna aplicación web que permita gestionar información referente a matrículas vehiculares en el cual se debe contemplar la validación de campos

1. **RESULTADO DE APRENDIZAJE ATADO AL TALLER PRÁCTICO EXPERIMENTAL**

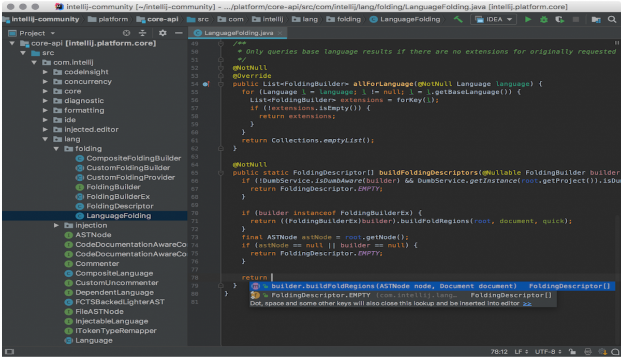
Prepara adecuadamente Spring Boot y el IDE Intellij IDEA para la construcción de aplicaciones web modernas en cohesión con tecnologías del backend y frontend.

1. **FUNDAMENTACIÓN TEÓRICA DE LA PRÁCTICA**

**¿Qué es intellij?**

IntelliJ IDEA es un entorno de desarrollo integrado (IDE) para Java y Kotlin diseñado para maximizar la productividad de los desarrolladores. Realiza las tareas rutinarias y repetitivas por usted al proporcionar finalización de código inteligente, análisis de código estático y refactorizaciones, y le permite concentrarse en el lado positivo del desarrollo de software, lo que lo convierte no solo en una experiencia productiva sino también agradable.

**IntelliJ IDEAVentajas para el desarrollo**

Todo desarrollador web necesita herramientas que faciliten y hagan evolucionar su trabajo, y los que trabajan con lenguajes como Java o Kotlin han encontrado en**IntelliJ IDEA** un entorno de desarrollo integrado (IDE) que cumple con todo lo necesario para convertirse en su software de cabecera.

Esta herramienta agiliza el proceso de codificación y ofrece funcionalidades que mejoran la productividad, pero también la calidad del software. ¡Descubre cómo exprimir al máximo sus propiedades!

**Spring Web**

 Es un módulo del framework Spring Framework que permite crear aplicaciones web y servicios web RESTful. Spring Web facilita el desarrollo de aplicaciones web al reducir la cantidad de código repetitivo y configuración.

**Thymeleaf**

Es un motor de plantillas de Java que permite crear documentos a partir de plantillas y modelos de datos. Se utiliza principalmente en entornos web, pero también puede funcionar en entornos no web.

**Spring Boot Devtools**

Es un conjunto de herramientas que ayudan a los desarrolladores a crear aplicaciones Spring. Devtools es la abreviatura de Developer Tool, que significa herramienta para desarrolladores.

**Lombok**

Es una biblioteca de código abierto para Java que facilita la escritura de código. Se utiliza para reducir el código repetitivo y mejorar la legibilidad del código.

**Inyección por campo con @Autowired**

Es un mecanismo de inyección de dependencias en **Spring Boot 3** que permite inyectar directamente las dependencias en los campos de una clase utilizando la anotación **@Autowired**. En este enfoque, el contenedor de **Spring** se encarga de asignar valores a los campos marcados con **@Autowired** sin necesidad de constructores o métodos setter.

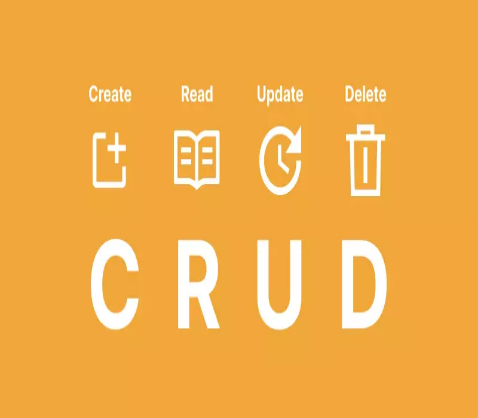


**¿Qué es Tailwind CSS?**

Tailwind, en pocas palabras, es un framework CSS que da prioridad a la utilidad sobre el propio estilo, pero además a diferencia de otros frameworks CSS como Bootstrap o Bulma, Tailwind no provee una serie de componentes predefinidos. En su lugar, este framework opera en un nivel inferior y te proporciona un conjunto de clases de ayuda para estructura y estilado, de forma que, usando dichas clases, puedas crear rápidamente diseños personalizados con facilidad. Además, no es opinionado y gracias a su flexibilidad te permite crear un diseño realmente único.

**CRUD**

Es el acrónimo de Create (Crear), Read (Leer), Update (Actualizar) y Delete (Borrar). Este concepto se utiliza para describir las cuatro operaciones básicas que pueden realizarse en la mayoría de las bases de datos y sistemas de gestión de información.

**Las ventajas del CRUD**

Facilita la creación y gestión de datos.

Proporciona una estructura coherente y fácil de entender para su manipulación.

Ayuda a minimizar los errores y garantiza la integridad de los datos.

Proporciona una base sólida para el desarrollo de aplicaciones.Y las desventajas del CRUD son que:

Puede ser demasiado simplista para aplicaciones complejas.

En ocasiones, es menos eficiente para aplicaciones de alta velocidad o de gran escala.

A veces, requiere una gran cantidad de código y configuración para implementarlo completamente.

1. **DESCRIPCIÓN DE LAS ACTIVIDADES DESARROLLADAS DURANTE LA PRÁCTICA.**

Actividades ordenadas y en secuencia a ser desarrolladas durante el taller práctico experimental.

|  |  |
| --- | --- |
| ACTIVIDAD | DESCRIPCIÓN |
| Creación del proyecto en la página spring initializr. | Se genera la estructura base del proyecto seleccionando el lenguaje, versión de Spring Boot y las dependencias necesarias. Luego, se descarga el proyecto en formato ZIP. |
| Abrir el proyecto en intelliJ IDEA | Se importa el proyecto descargado en IntelliJ IDEA, configurando el entorno y asegurando que todas las dependencias se resuelvan correctamente. |
| Realización de la página index ,html. | Se crea la página principal del proyecto utilizando HTML y, si es necesario, se integra con Tailwind CSS para el diseño. |
| Realización de los controladores | Se crean las clases controladoras en Java que gestionan las solicitudes HTTP y envían las respuestas adecuadas. Se utilizan anotaciones como `@Controller` y `@GetMapping`. |
| Realización de las entidades | Se definen las clases de entidad en Java que representan las tablas de la base de datos, utilizando anotaciones de JPA. |
| Realización de la validación de los formularios. | Se implementa la validación de los datos ingresados en los formularios utilizando anotaciones como `@NotNull`, `@Size`, `@Email`, etc., en las clases de entidad. |

1. **MATERIALES Y EQUIPOS**

Corresponde a los materiales, equipos, insumos, maquinaria, instrumental, etc. disponibles en los laboratorios institucionales o escenarios de aprendizaje asignados.

|  |  |  |
| --- | --- | --- |
|  | MATERIAL / EQUIPO/ INSUMO | UTILIDAD |
|  | Computadora | Dispositivo principal para desarrollar, compilar y ejecutar el código. |
|  | Internet | Permite la descarga de los proyectos, acceso a documentación y repositorios de código. |
|  | ItelliJ IDEA | Entorno de desarrollo integrado (IDE) para escribir, depurar y administrar proyectos en Java y TypeScript. |
|  | spring initializr | Herramienta para generar la estructura inicial de proyectos Spring Boot con las dependencias necesarias. |
|  | Dependencias | spring web (para realizar una página web.), Spring Boot Dev Tools (tiene herramientas para desarrollar), Thymeleaf (nos deja trabajar con vistas, html, nos permite trabajar con front), Lombok( Para trabajar con la data), validation (Para realizar validaciones). |

1. **IDENTIFICACIÓN DE RIESGOS Y EQUIPOS DE PROTECCIÓN PERSONAL (SI APLICA)**

Riesgos genéricos identificados por carrera y los equipos de protección personal específicos a ser utilizados por los estudiantes durante la ejecución de los talleres práctico experimentales.



1. **ANÁLISIS DE RESULTADOS**

Durante el desarrollo del taller, se logró generar correctamente el proyecto en Spring Initializr con las dependencias necesarias y abrirlo en IntelliJ IDEA sin errores. Se creó la página index.html para la interfaz del usuario, permitiendo la visualización y navegación adecuada. Los controladores fueron implementados con éxito, permitiendo la gestión de solicitudes HTTP para la visualización y carga de dietas. Las entidades se definieron correctamente en JPA, asegurando la persistencia de datos relacionados con los usuarios y sus planes de alimentación. Además, se incorporaron validaciones en los formularios para garantizar la correcta entrada de datos, como nombres, edades, restricciones alimenticias y objetivos nutricionales.

1. **CONCLUSIONES**

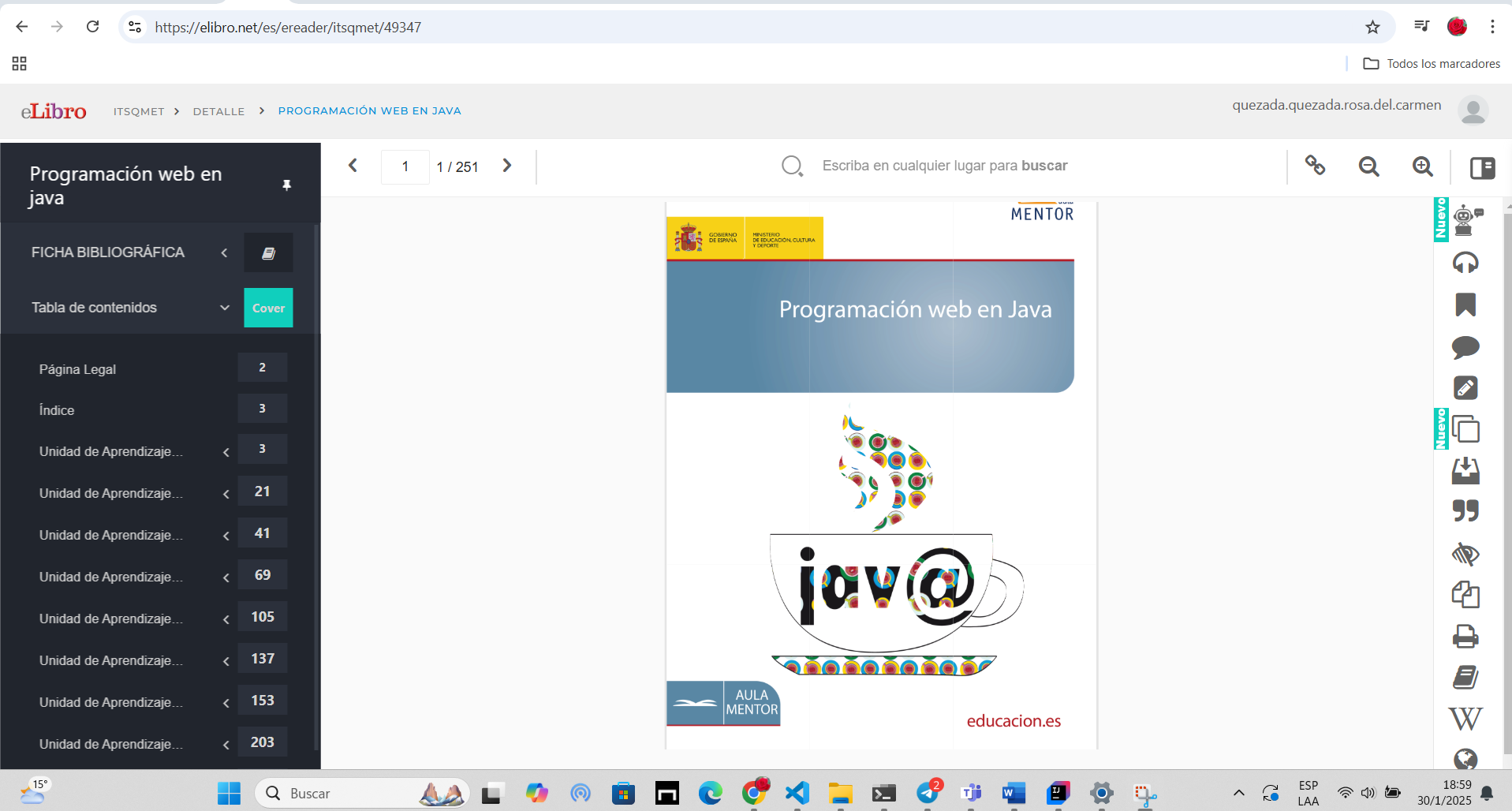
El proyecto alcanzó los objetivos propuestos, logrando crear una aplicación funcional para un nutricionista donde se pueden gestionar dietas de forma efectiva. La estructura del proyecto se desarrolló correctamente, con la integración de controladores, validaciones y la visualización de las dietas en una interfaz amigable. Además, la implementación de las entidades y los controladores permitió que la aplicación procesara correctamente las solicitudes del usuario. A pesar de que aún no se ha integrado una base de datos.Las validaciones en los formularios aseguran la calidad de los datos ingresados. Sin embargo, para mejorar la escalabilidad, seguridad y funcionalidad a largo plazo, se recomienda la integración de una base de datos externa.

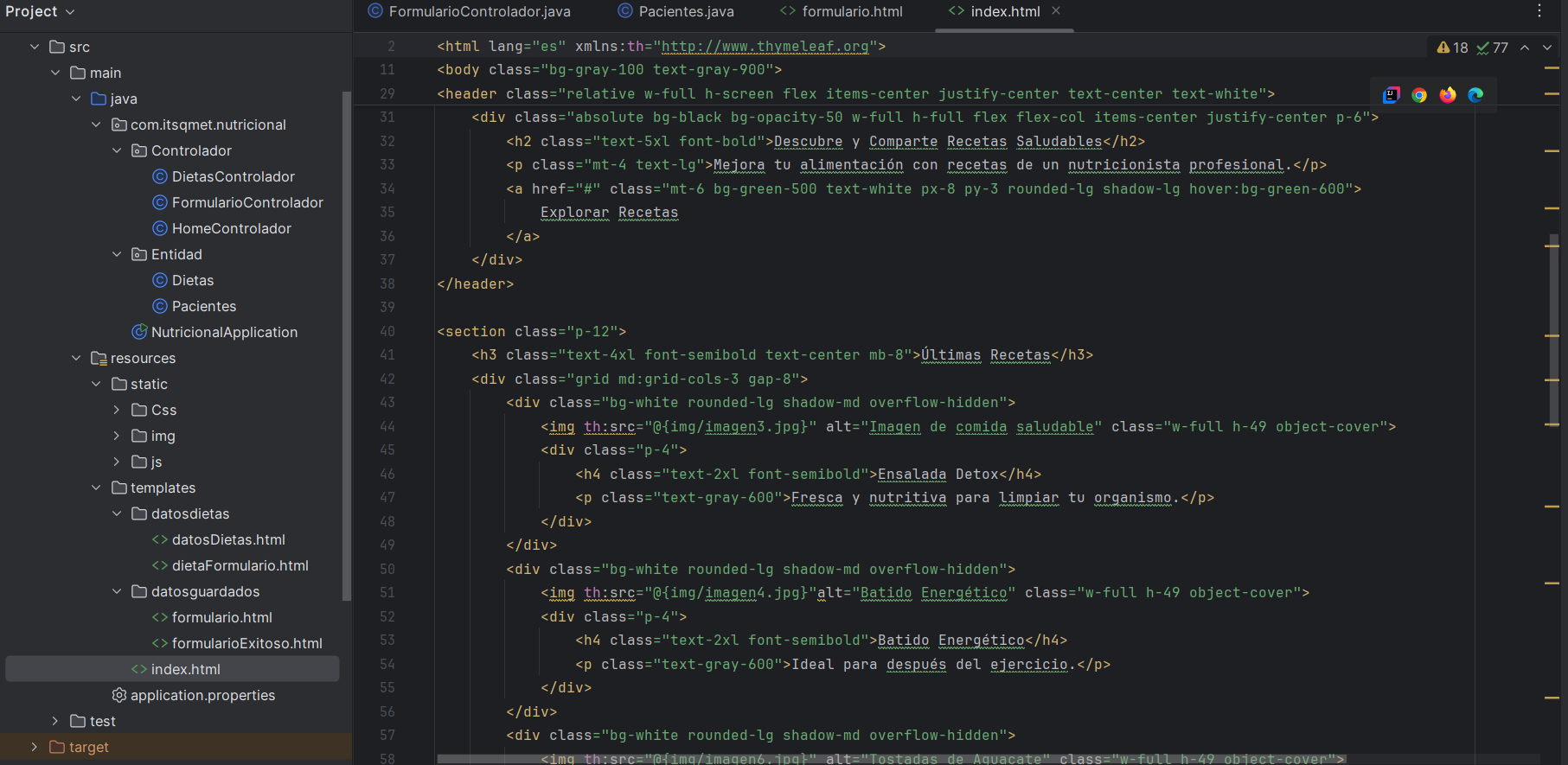
1. **RECOMENDACIONES**

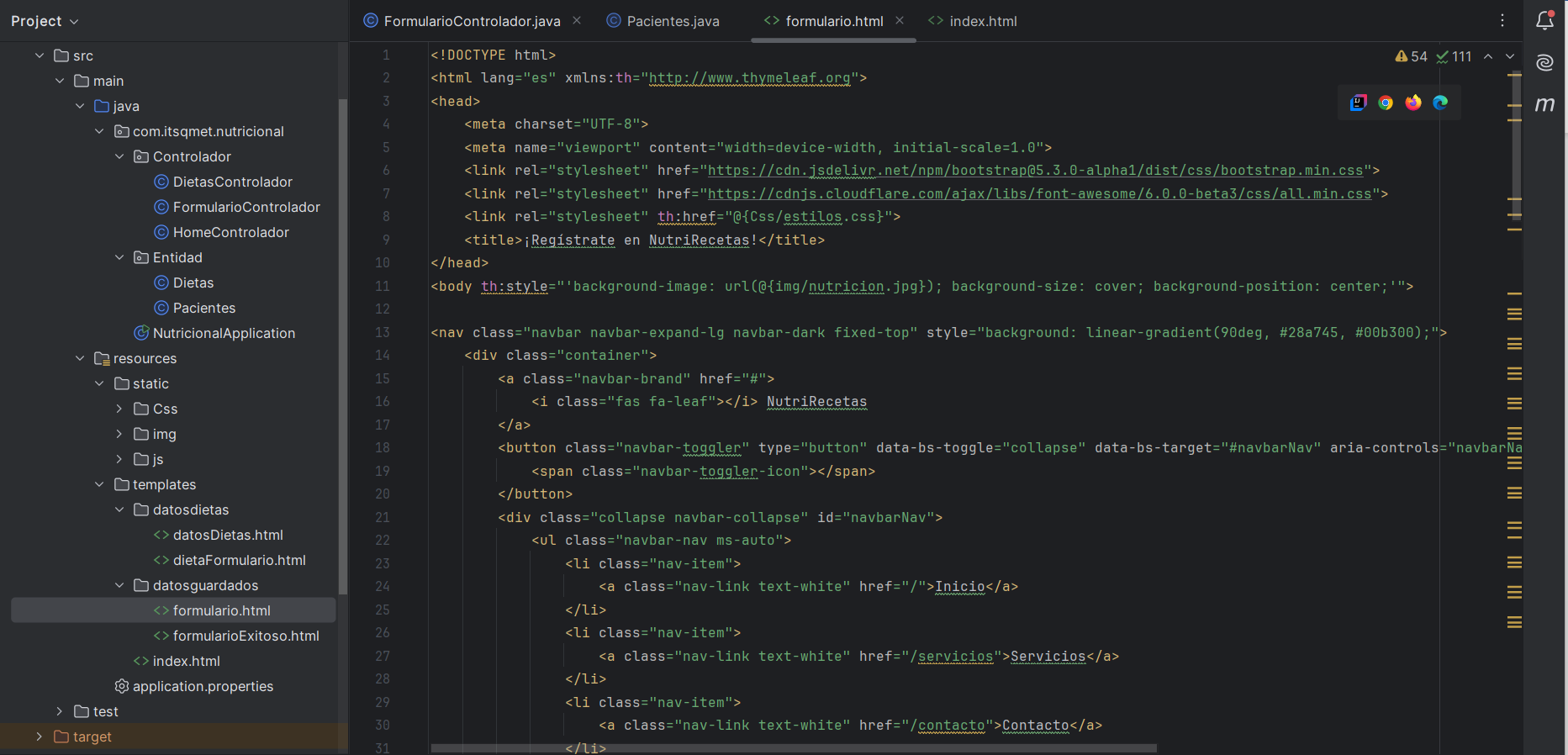
Se recomienda mejorar la interfaz de usuario utilizando más características de Tailwind CSS para hacerla más atractiva. También sería útil optimizar la gestión de dietas permitiendo su personalización según objetivos específicos, y agregar validaciones más avanzadas en los formularios para garantizar entradas más precisas. En el futuro, sería conveniente implementar una base de datos para almacenar las dietas y usuarios, y un sistema de autenticación con Spring Security para aumentar la seguridad. Además, permitir la subida de imágenes o archivos PDF relacionados con las dietas mejoraría la funcionalidad. Realizar pruebas automatizadas también garantizaría el correcto funcionamiento del sistema antes de su despliegue.

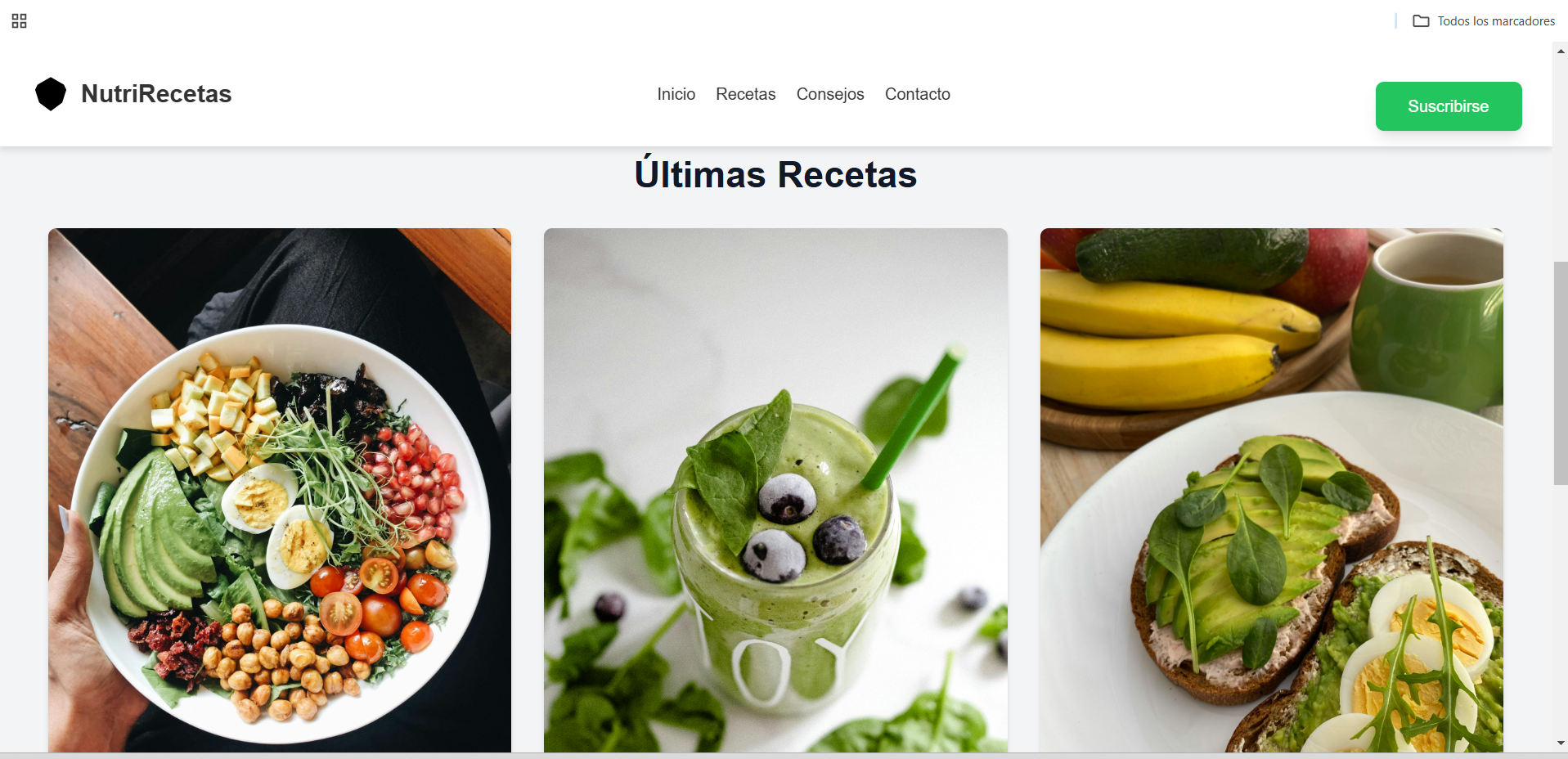
**EVIDENCIA FOTOGRÁFICA DE LA PRÁCTICA**

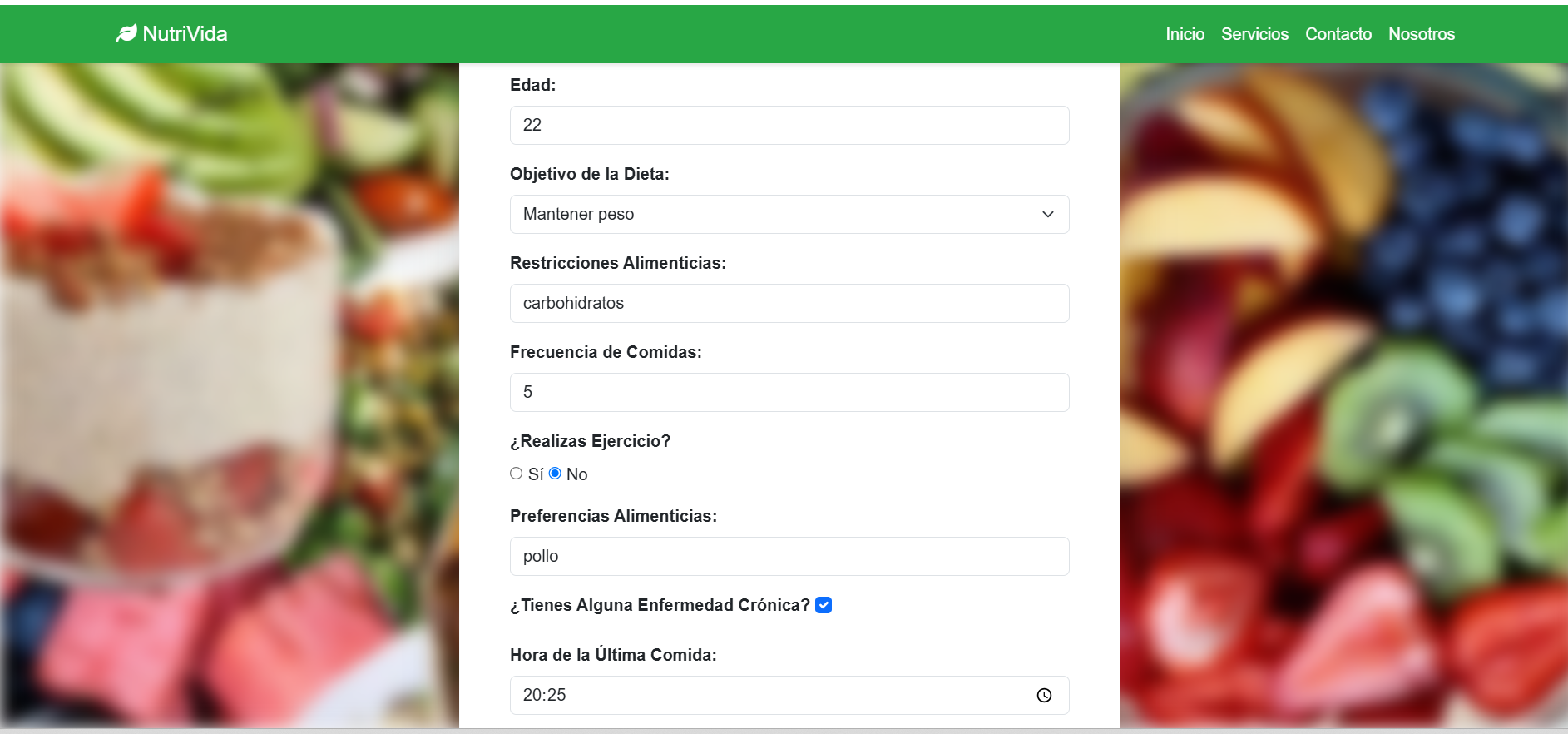


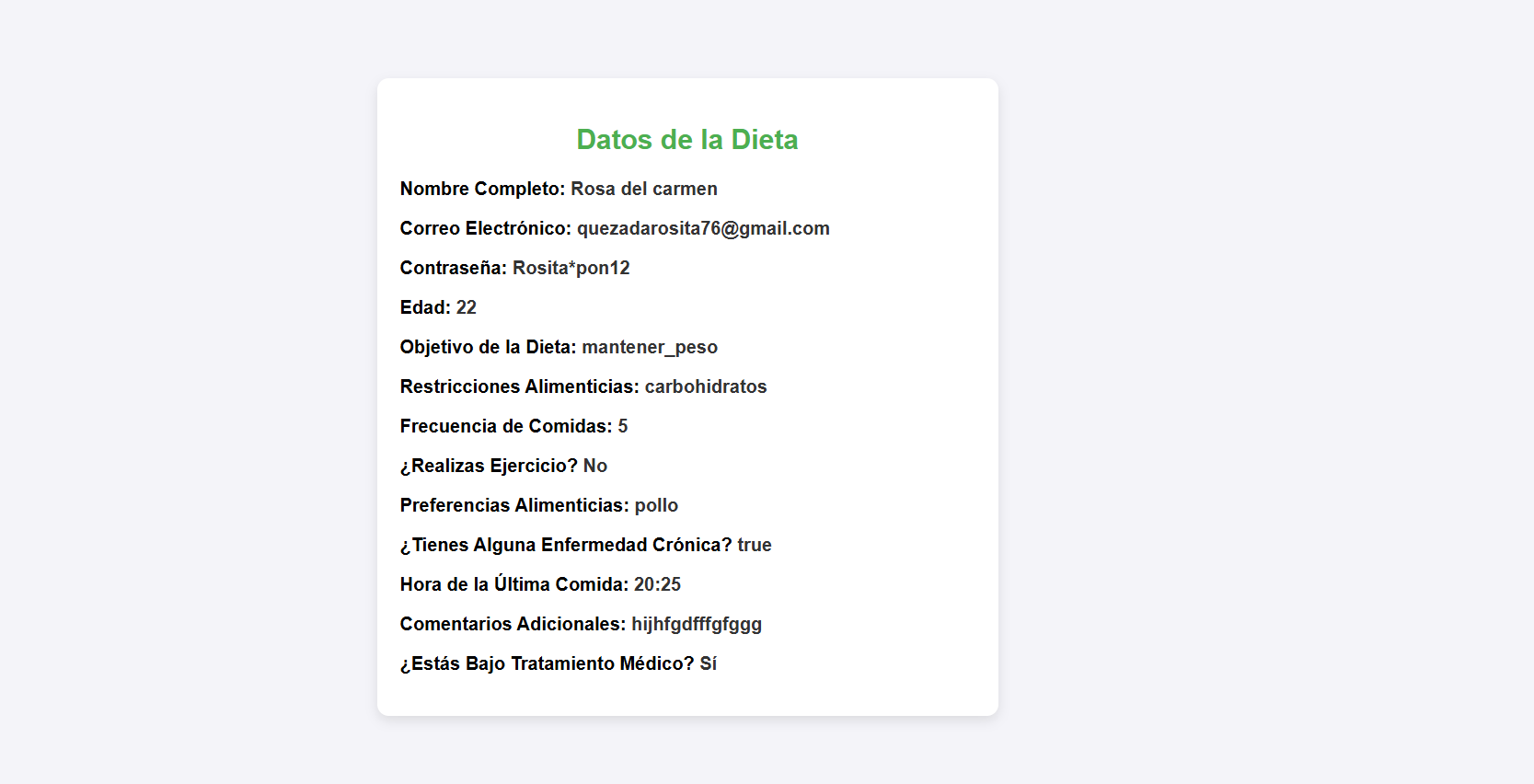














1. **BIBLIOGRAFÍA**

.https://blog.hubspot.es/website/que-es-crud.

<https://certidevs.com/tutorial-spring-boot-inyeccion-de-dependencias>

<https://openwebinars.net/blog/que-es-lombok/>

<https://www.arquitecturajava.com/spring-boot-devtools-y-recarga-de-aplicaciones/>

<https://openwebinars.net/blog/que-es-thymeleaf/>

https://elibro.net/es/lc/itsqmet/titulos/49347