Documentación del Sistema de Recomendación de Comidas

# 📌 Descripción general

Este sistema es una aplicación de escritorio desarrollada en Java con JavaFX y Maven. Su objetivo es recomendar comidas saludables y personalizadas a los usuarios según sus preferencias, historial de consumo y necesidades calóricas. Utiliza Neo4j como base de datos orientada a grafos para modelar las relaciones entre usuarios, comidas, preferencias y puntuaciones. El proyecto está dividido en clases modelo, lógicas de recomendación, vistas JavaFX y servicios de conexión.

# 💻 Requisitos del sistema

Sistema operativo compatible: macOS (probado), Windows 10/11, Linux

Requisitos de software:

- Java JDK 17 o superior

- Maven 3.6.0+

- IntelliJ IDEA (Community o Ultimate)

- Neo4j Desktop 5.x (con soporte Bolt habilitado)

# ⚙️ Instalación y ejecución

1. Clonar el repositorio: https://github.com/MrMenth0l/ProyectoDietas

2. Abrir Neo4j Desktop, crear una base de datos con URI: bolt://localhost:7687, usuario: neo4j, contraseña: Estructuras123

3. Ejecutar la clase `FoodAdd.java` para poblar la base de datos con comidas desde archivos CSV e imágenes.

4. Desde IntelliJ, correr el proyecto con el comando: mvn clean javafx:run

# 📦 Componentes principales del sistema

• Comida.java: Clase modelo que representa una comida con sus valores nutricionales, tipo, imagen, receta y puntuaciones.

• ComidaService.java: Servicio para consultar comidas desde Neo4j (todas o muestras iniciales).

• Enums.java: Define `RespuestaUsuario` (votos) y `MomentoDelDia` (desayuno, almuerzo, cena).

• FoodAdd.java: Utilidad para cargar comidas desde archivo o consola en la base de datos.

• Neo4JConnect.java: Singleton que gestiona la conexión al driver de Neo4j.

• RecomendadorAlgo.java: Motor de recomendaciones, filtra comidas, ordena por puntuación y proximidad calórica.

• RegistroComida.java: Registra el consumo real de comidas por parte del usuario en Neo4j.

• RegistroPuntuacion.java: Representa votos individuales con fecha y puntuación.

• Usuario.java: Clase modelo para usuarios, sus datos, historial y preferencias.

# 🖼️ Interfaz de usuario (JavaFX)

• LoginView.java: Permite iniciar sesión. Verifica credenciales y redirige a preferencias o recomendaciones.

• RegistroView.java: Formulario para registrar un nuevo usuario, con hash seguro de contraseña (BCrypt).

• PreferenciasInicialesView.java: Vista para que usuarios nuevos voten comidas y definan sus preferencias.

• PanelPrincipalView.java: Muestra recomendaciones con imagen, info nutricional y botones para votar.

• MainApp.java: Clase principal que lanza la aplicación desde `org.example.View` con JavaFX.

# 🎨 Estilos visuales

El archivo `estilos.css` define los estilos visuales usados por las vistas:

- `.form-background`, `.label`, `.text-field`, `.button`: estilo visual uniforme y limpio.

- `.titulo`, `.card-comida`, `.imagen-comida`: aplicados para mejorar la experiencia visual del usuario.

# 📄 Archivo pom.xml

El archivo `pom.xml` contiene las siguientes dependencias:

- `neo4j-java-driver`: para conectar con Neo4j.

- `javafx-controls`: para construir la interfaz.

- `jbcrypt`: para encriptar contraseñas.

También configura el plugin `javafx-maven-plugin` para lanzar la aplicación desde la clase `MainApp.java`.

# 🧠 Motor de recomendación

La clase `RecomendadorAlgo` implementa el algoritmo principal de sugerencias. Utiliza los siguientes criterios:

- El momento del día (desayuno, almuerzo, cena)

- Las preferencias iniciales del usuario

- La puntuación acumulada de cada comida

- La cercanía calórica al historial del usuario

El motor permite integración externa llamando a: RecomendadorAlgo.recomendar(usuario, historial, cantidad)

# ✅ Conclusión

El sistema de recomendación de comidas desarrollado en JavaFX y Neo4j permite una experiencia personalizada, segura y ampliable. Su arquitectura modular y conectividad con base de datos lo hacen ideal para extenderlo a otras plataformas o usarlo como motor embebido en sistemas mayores.