MENÚPLANNER

NAHIA CASTAÑEDA FRANCO

OBJETIVO

Esta aplicación permite a los usuarios crear, gestionar y organizar recetas, consultar recetas de una API externa y generar menús semanales.

FUNCIONALIDADES

- Permitir la identificación del usuario
- Crear recetas personalizadas con ingredientes, pasos y tiempos de preparación.
- Consultar recetas desde una API externa y guardarlas.
- Marcar recetas como favoritas para un acceso rápido.
- Generar menús semanales organizados para una mejor planificación.

APIS (BACKEND PYTHON)

- Autenticación
  - POST /api/login->Permite a los usuarios iniciar sesión.
  - POST /api/register->Permite registrar nuevos usuarios.
- Gestión de Recetas
  - POST /api/recipes/imported/add ->Agrega una receta importada asociada a un usuario.
  - POST /api/recipes/add -> Agrega una receta personalizada.
  - GET /api/recipes -> Obtiene todas las recetas asociadas a un usuario.

APIS (BACKEND NODE)

- Gestión de Favoritos
  - POST /api/favorites -> Añade una receta a los favoritos del usuario.
  - GET /api/favorites -> Obtiene las recetas favoritas de un usuario.

MODELO DE DATOS RELACIONAL

- Usuarios (usuarios):
  - id: INT, PK – Identificador único del usuario.
  - nombre: VARCHAR(255) – Nombre del usuario.
  - correo: VARCHAR(255) – Correo electrónico del usuario (único).
  - contrasena: VARCHAR(255) – Contraseña del usuario.
- Recetas (recetas):
  - id: INT, PK – Identificador único de la receta.
  - usuario_id: INT, FK – Relación con el usuario que creó la receta.
  - nombre: VARCHAR(255) – Nombre de la receta.
  - ingredientes: TEXT – Lista de ingredientes separados por comas.
  - instrucciones: TEXT – Lista de instrucciones separadas por punto.
  - tiempo_preparacion: VARCHAR(20) – Tiempo estimado de preparación.
  - imagen_url: VARCHAR(255) – URL de la imagen de la receta.

MODELO DE DATOS NO RELACIONAL

- Usuarios (User):
  - _id: ObjectId. PK – Identificador único del usuario.
  - email: String. Unique – Correo electrónico del usuario.
  - favoriteRecipes: [ObjectId] – Referencias a recetas favoritas.
  - weeklyMenu: Array – Lista de días y sus comidas asignadas.
    - weeklyMenu.day: String – Día de la semana (e.g., "Monday").
    - weeklyMenu.meals: [String] – Títulos de recetas asignadas a ese día.
- Recetas (Recipe):
  - _id: ObjectId. PK – Identificador único de la receta.
  - title: String. Unique – Título único de la receta.
  - ingredientes: [String] – Lista de ingredientes.
  - instrucciones: [String] – Lista de instrucciones.
  - imagenUrl: String – URL de la imagen de la receta.

ARQUITECTURA SOLUCIÓN

- Tipo: Microservicios
  - Separación de responsabilidades en servicios independientes para recetas, favoritos y menús semanales.
- Coordinación centralizada a través de un API Gateway, lo que facilita la escalabilidad y la integración.
- Tecnologías Seleccionadas:
  - Frontend:
    - React.js: Cliente SPA (Single Page Application) moderno.
    - Extensivo uso de HTML5, CSS y JavaScript para una experiencia interactiva y dinámica.
- Backend:
  - Node.js: Manejo de funcionalidades de favoritos y menús semanales.
  - Python: Gestión de recetas creadas e importadas desde APIs externas.
- Bases de Datos:
  - MySQL: Datos relacionales como usuarios y recetas.
  - MongoDB: Datos no relacionales como favoritos.

LECCIONES APRENDIDAS

- Lecciones Aprendidas:
  - Diseñar una arquitectura modular con microservicios.
  - Integrar tecnologías variadas como React, Python y Node.js.
- Aspectos Destacados:
  - Uso de API Gateway para centralizar solicitudes.
  - Bases de datos híbridas (MySQL y MongoDB) para eficiencia y flexibilidad.
  - Consultas a APIs externas en tiempo real para enriquecer contenido.