DOCUMENTACIÓN DEL PROYECTO – HEALTHY BITE

---

1. Modelo ambiental

Healthy Bite es una aplicación web pensada para permitir que los clientes puedan hacer pedidos de comida saludable y simultáneamente recibir recomendaciones de nutricionistas. El sistema se desarrolló en Python usando Flask.

Entorno de desarrollo: se trabajó con Python 3, VS Code, y SQLite como base de datos. Para el deploy se usó Render. Además, se usa Google Auth para el inicio de sesión

---

2. Estructura del proyecto

Para mantener el proyecto ordenado se aplicó la estructura de blueprints. La aplicación está organizada de esta forma:

- **healthy_bite/auth/**
  Maneja todo lo relacionado al login, logout y el inicio de sesión con Google.

- **healthy_bite/clientes/**
  Contiene las rutas y vistas que puede usar un cliente: hacer pedidos, ver recomendaciones, etc.

- **healthy_bite/nutricionistas/**
  Se encarga de la parte del nutricionista: ver clientes, validar pedidos, dejar recomendaciones, modificar planes y platos.

- **healthy_bite/main/**
  Tiene las rutas generales como la página de inicio.

- **templates/**
  Acá están todas las páginas HTML.

- **static/**
  Archivos CSS e imagen de portada.

- **db.py**
  Archivo donde se abre la base, se crean las tablas y se maneja la conexión.

- **init.py**
  Acá está el App Factory, donde se crea la aplicación, se cargan las variables de entorno y se registran los blueprints.

- **run.py**
- Archivo que ejecuta la aplicación cuando se corre de manera local.

Fuera de la carpeta principal están:

- requirements.txt

- Procfile

- README

- .gitignore

La idea fue dejar todo separado para que sea fácil de leer.

---

3. Librerías y dependencias

- **Flask**: framework principal para manejar rutas, plantillas, sesiones y blueprints.

- **python-dotenv**: para cargar las variables del archivo .env (secret key, credenciales de Google, etc).

- **google-auth-oauthlib**: para implementar el inicio de sesión con Google.

- **requests**: para completar el flujo de autenticación.

- **gunicorn**: para poder hacer el deploy en Render.

- **sqlite3**: base de datos incluida en Python.

---

4. Instrucciones de ejecución (cómo correr el proyecto)

Para levantar la app localmente, los pasos son estos:

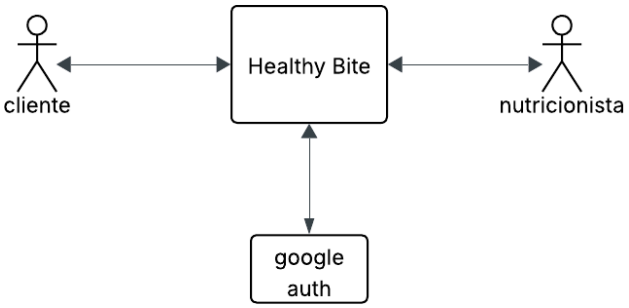
1. Clonar el repositorio desde GitHub.
2. Crear un entorno virtual (por ejemplo, en Windows: `py -m venv .venv`).
3. Activar el entorno virtual.
4. Instalar las dependencias con `pip install -r requirements.txt`.
5. Crear un archivo `.env` en la raíz del proyecto con las siguientes variables:
    - SECRET_KEY
    - GOOGLE_CLIENT_ID
    - GOOGLE_CLIENT_SECRET
    - GOOGLE_REDIRECT_URI
6. Ejecutar el archivo run.py con `py run.py`.
7. Abrir el navegador en `http://localhost:5000`.

La base de datos SQLite se crea automáticamente la primera vez que se ejecuta la aplicación.

---

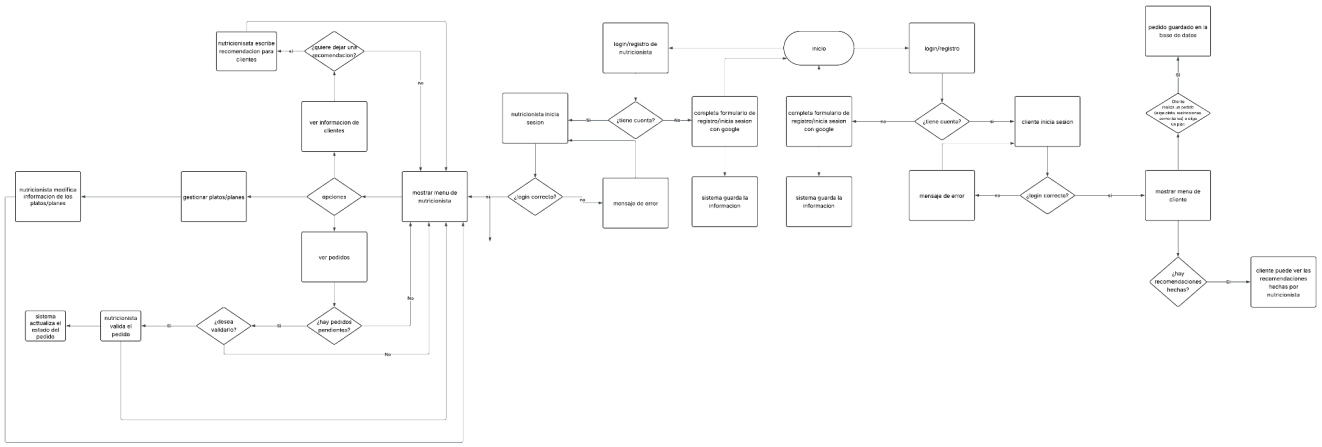
5. Diagramas

5.1 Diagrama de contexto



El cliente interactúa con la app para hacer pedidos, el sistema los guarda y el nutricionista puede verlos y validarlos. También existe la comunicación con Google Auth solo para el primer inicio de sesión.

5.2 Diagrama de flujo



https://lucid.app/lucidchart/7b7dd638-42e6-4032-9ce7-b9885ebfdd86/edit?beaconFlowId=9B11BC3F3B0C8671&invitationId=inv_d420ec1e-2fbd-4779-9a72-880b3eee9b5b&page=0_0#

---

6. Resultados y reflexión

El proyecto quedó funcionando correctamente tanto de manera local como en Render. La implementación de Google Auth fue la parte donde hubieron más complicaciones, sobre todo por el tema de las variables de entorno y las URLs de redirección. El resto fue más directo.

Lo que más costó fue la organización de los blueprints al principio y entender bien cómo dividir las rutas de cliente y nutricionista, ya que en el proyecto de java del que hice la migración no tenía todo separado.

Como posibles mejoras a futuro:

- Agregar validaciones más estrictas en los formularios.
- Mostrar estadísticas o reportes para nutricionistas.
- Agregar imágenes a los platos..
- Que el cliente tenga un chat directo con el nutricionista para seguir el progreso hacia su objetivo.