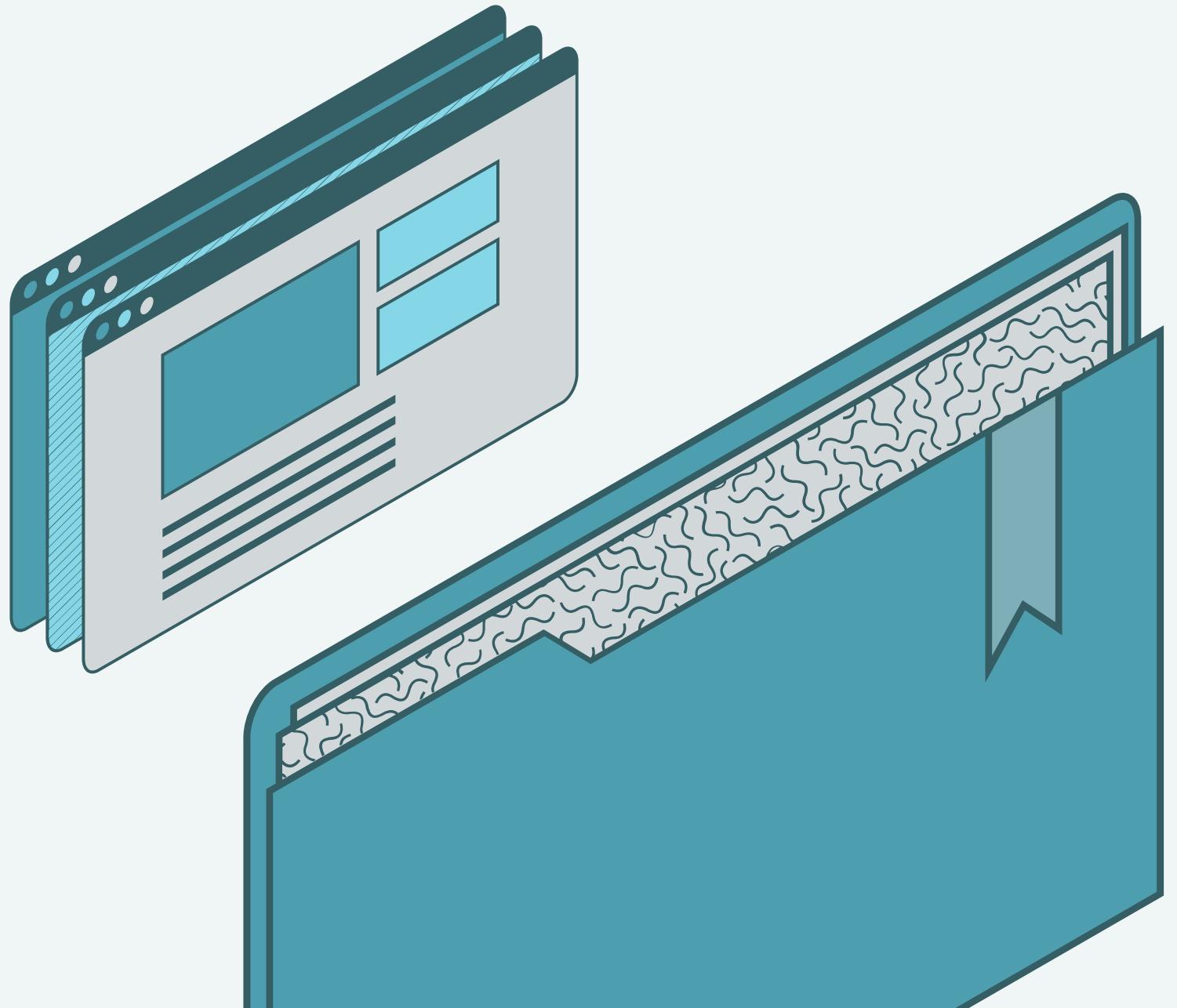


LOS TELETUBBIES

# SMARTBIT

# íNDICE



---

**01. Idea General del Proyecto**

---

**02. Avance Actual y Demostración**

---

**03. Organización y Roles del Equipo**

---

**04. Decisiones Técnicas y Desafíos**

---

**05. Tareas Pendientes y Plan**

---

**06. Reflexión y Aprendizajes**

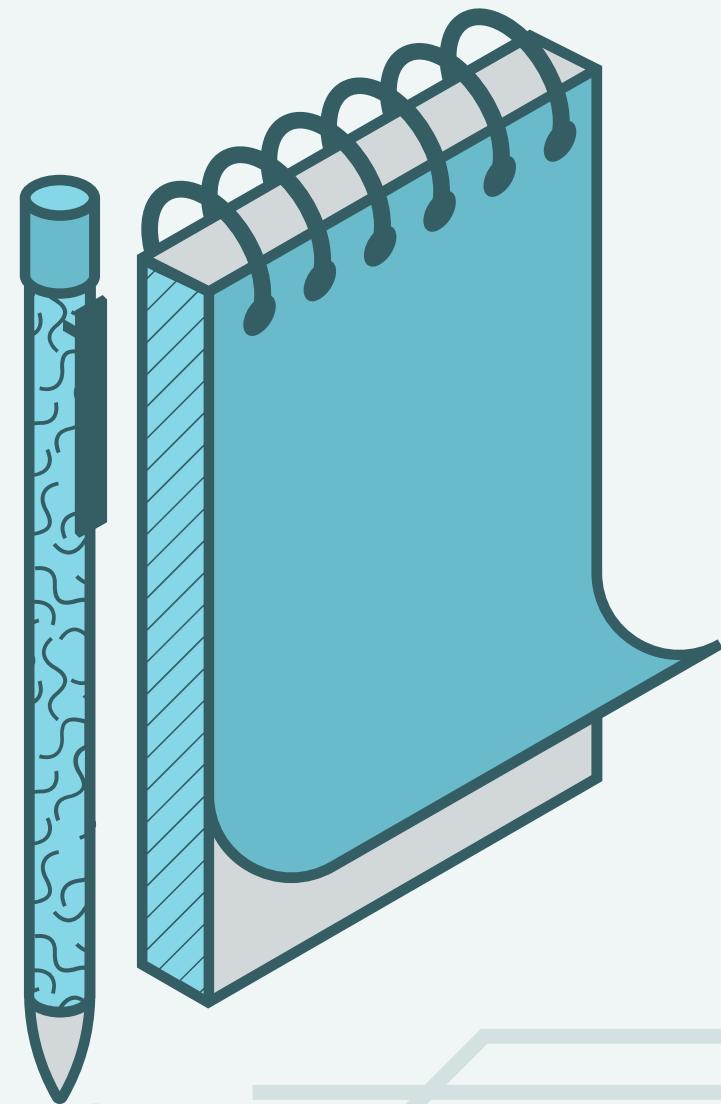
---

**07. Agradecimiento**

---

# TÍTULO E IDEÁ GENERAL DEL PROYECTO

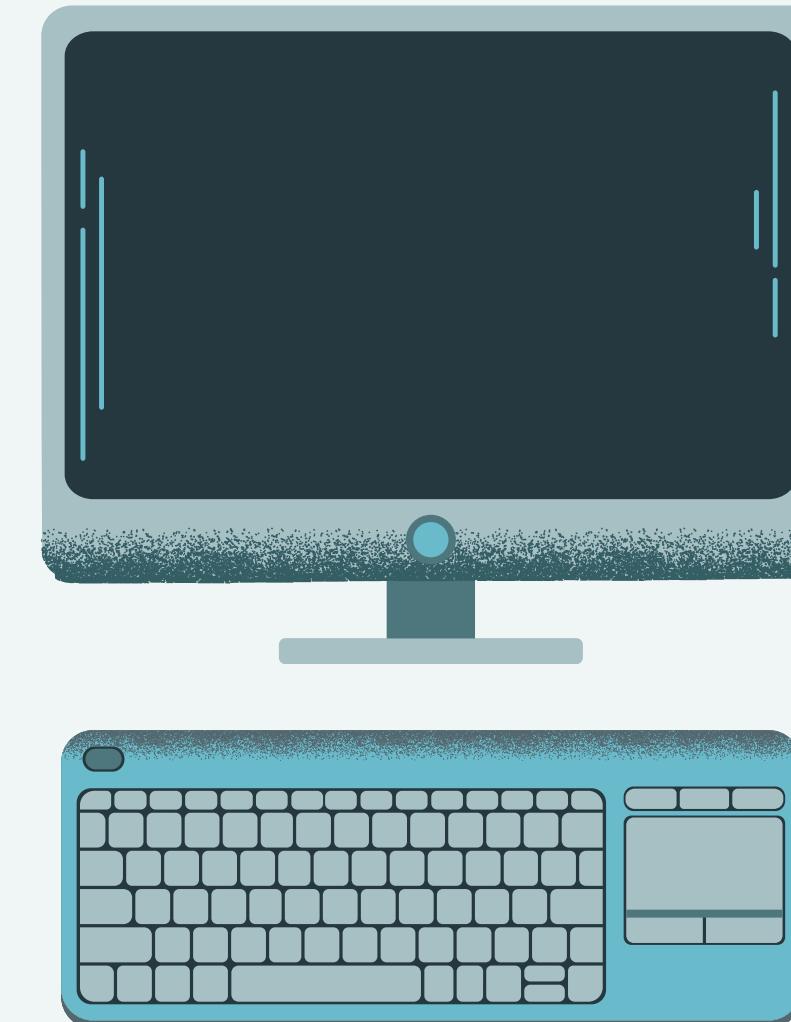
- Título: SmartBite: Tu Asistente Nutricional Inteligente.
- Idea Central: Una aplicación web que ofrece planes nutricionales personalizados utilizando un modelo de recomendación inteligente.
- Tecnologías Clave:
- Backend: Python con el framework Flask.
- Frontend: HTML y Tailwind CSS para un diseño moderno.
- Inteligencia Artificial: Un modelo de recomendación para generar las sugerencias.



# AVANCE ACTUAL Y DEMOSTRACIÓN VISUAL

## ESTADO GENERAL

El proyecto está en una fase avanzada de desarrollo, con los componentes más complejos ya finalizados.



## MODELO DE RECOMENDACIÓN

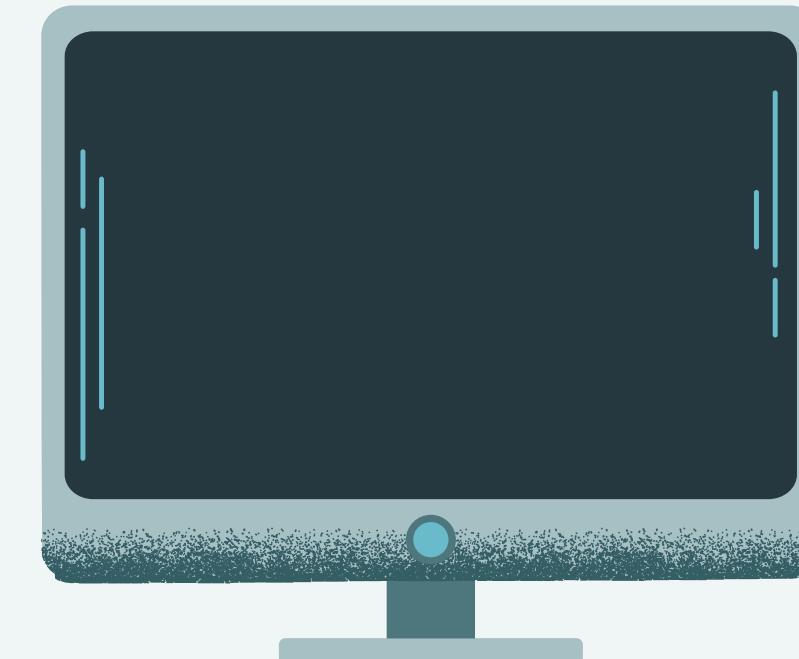
¡Terminado y funcional! Esta fue la parte más desafiante y que más tiempo consumió, pero es el corazón de SmartBite y ya está operativo.

# AVANCE ACTUAL Y DEMOSTRACIÓN VISUAL

## FRONTEND

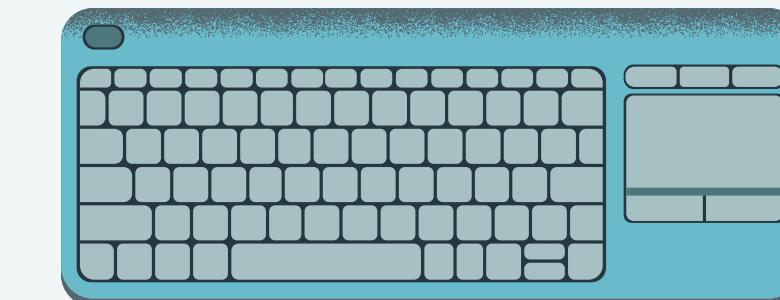
La interfaz de usuario está completamente diseñada.

Tenemos las pantallas listas para que el usuario interactúe con la aplicación.



## BACKEND Y BASE DE DATOS

La arquitectura base con Flask y Docker está implementada.



**Diet Recommendations**

Meals per day: 3 | Plan: Maintain weight

BREAKFAST	LUNCH	DINNER
South of the Border Orzo Salad #RSC	Baked Pork Chop Goulash	Southwestern Egg Rolls
Pasta With Chicken and Vodka Sauce	No-Peek Beef Stew With Red Wine	Grown up Goat Cheese Mac
Mediterranean Pasta Toss	Vermicelli With Chicken and Chorizo	Awesome Crock-Pot Chicken Pot Pie Soup
30 Minute Cincinnati Chili	One Skillet Italian Lasagna	Double Decker Fish Tacos
Steak House Fajitas	Ww Louisiana Chicken.	Chipotle Pork Tacos With Radish Coleslaw

Recommendation generated successfully!

Gender  
 Male  Female

Choose your weight plan

Activity  
 Little/no exercise Light exercise (1-2 days/wk) Moderate exercise (3-5 days/wk) Active (6-7 days/wk) Extra active (very physical job)

Selected: Moderate exercise (3-5 days/wk)

Meals per day: 3

**BMI Calculator**  
**22.23 kg/m<sup>2</sup>**  
Normal

**MAINTAIN WEIGHT**  
**2501 Calories/day**  
Maintain current weight

**MILD WEIGHT LOSS**  
**2251 Calories/day**  
-0.25 kg/week

**WEIGHT LOSS**  
**2001 Calories/day**  
-0.5 kg/week

**EXTREME WEIGHT LOSS**  
**1501 Calories/day**  
-1.0 kg/week

**Diet Recommendations**

Meals per day: 3 | Plan: Maintain weight.

BREAKFAST	LUNCH	DINNER
South of the Border Orzo Salad #RSC	Baked Pork Chop Goulash	Grown up Goat Cheese Mac
Pasta With Chicken and Vodka Sauce	No-Peek Beef Stew With Red Wine	One Skillet Italian Lasagna
Mediterranean Pasta Toss	Vermicelli With Chicken and Chorizo	Awesome Crock-Pot Chicken Pot Pie Soup
30 Minute Cincinnati Chili	Ww Louisiana Chicken.	Double Decker Fish Tacos

### Automatic Diet Recommendation

Modify the values and click the Generate button.

Age	Height (cm)	Weight (kg)
31	180.0	70.0

Gender  
 Male  Female

Choose your weight plan

Activity  
 Little/no exercise Light exercise (1-2 days/wk) Moderate exercise (3-5 days/wk) Active (6-7 days/wk) Extra active (very physical job)

Selected: Moderate exercise (3-5 days/wk)

Meals per day: 3

**BMI Calculator**  
**21.6 kg/m<sup>2</sup>**  
Normal

**MAINTAIN WEIGHT**  
**2596 Calories/day**  
Maintain current weight

**MILD WEIGHT LOSS**  
**2346 Calories/day**  
-0.25 kg/week

**WEIGHT LOSS**  
**2096 Calories/day**  
-0.5 kg/week

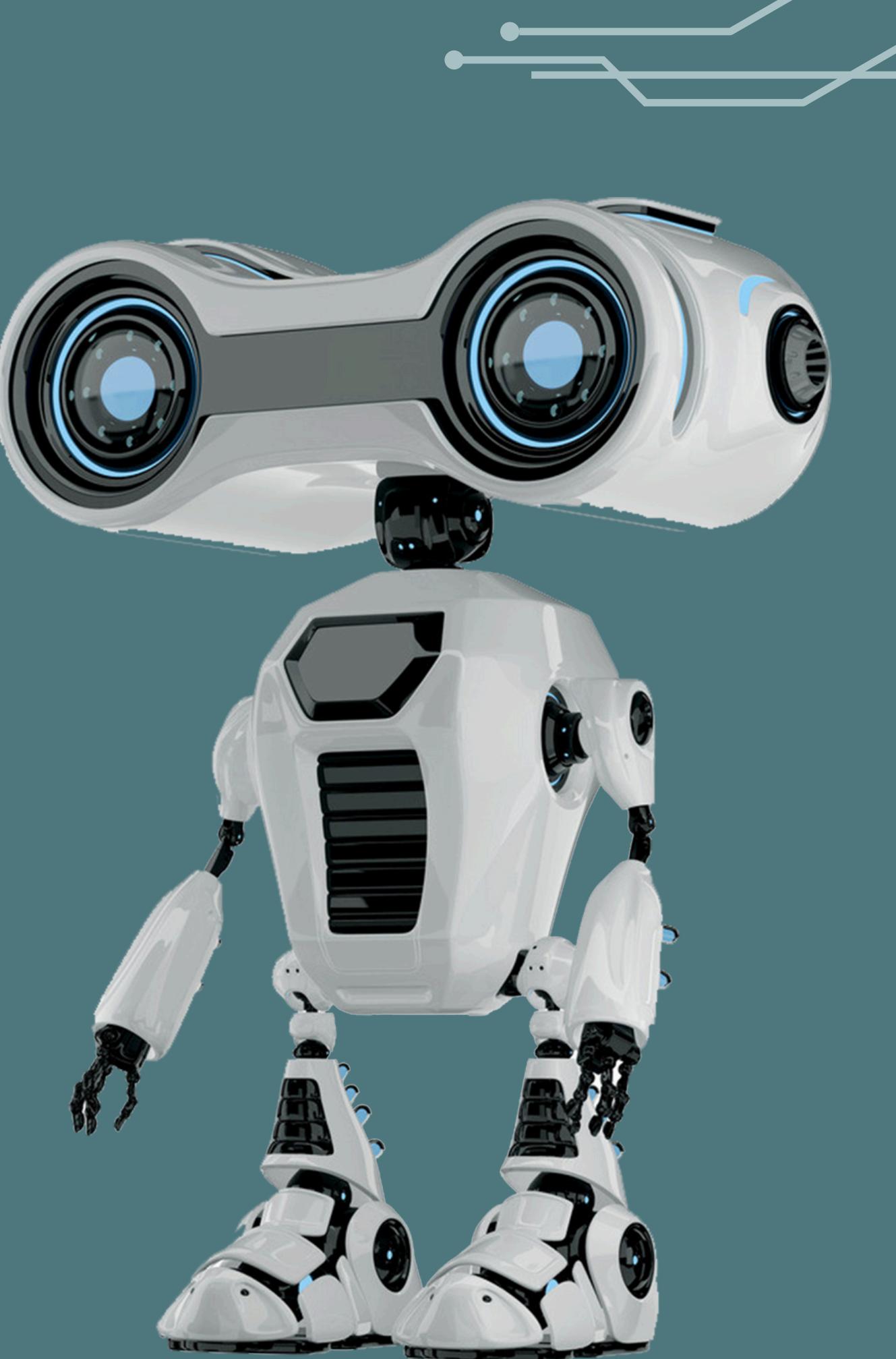
**EXTREME WEIGHT LOSS**  
**1596 Calories/day**  
-1.0 kg/week

# ORGANIZACIÓN Y ROLÉS

Distribución de Roles:

- Jefe de Producto y Backend: Deiber
- Científico de Datos / Ingeniero de IA: Camilo
- Desarrollador Frontend: Yeiner
- Arquitecto DevOps y de Base de Datos:  
Nicole

Herramientas: Usamos GitHub para el control de versiones, Docker para estandarizar nuestro entorno y herramientas de comunicación directa para mantenernos sincronizados.



# DECISIONES TÉCNICAS Y DESAFÍOS SUPERADOS

- Desafío Principal Superado: El Modelo de Recomendación.
  - Problema: Crear un sistema que diera recomendaciones verdaderamente personalizadas y relevantes no era trivial.
  - Solución: Tras investigar varias opciones, implementamos un modelo basado en K-Nearest Neighbors que demostró ser eficaz. Invertimos una cantidad significativa de tiempo en su entrenamiento y validación.

# DECISIONES TÉCNICAS Y DESAFÍOS SUPERADOS

- Decisión Técnica Clave: Arquitectura de Doble Base de Datos.
  - ¿Qué hicimos? Decidimos separar los datos en dos bases de datos independientes:

**BD del Modelo:** Contiene el gran volumen de recetas y datos nutricionales, utilizada exclusivamente por el algoritmo.

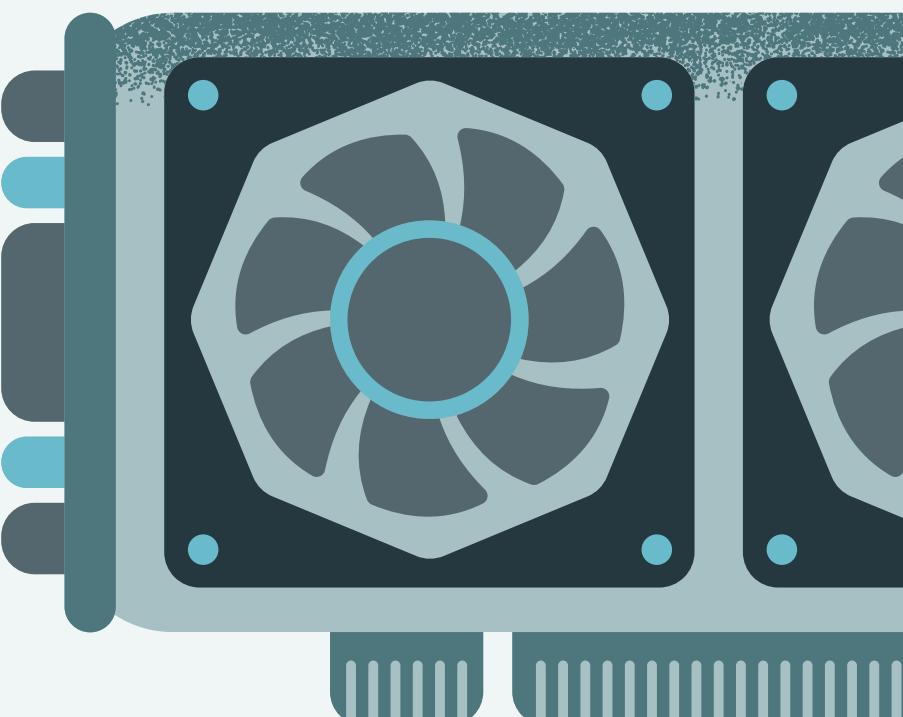
**BD de Usuarios:** Almacena la información sensible y personal de los usuarios (perfils, medidas, planes guardados, etc.).

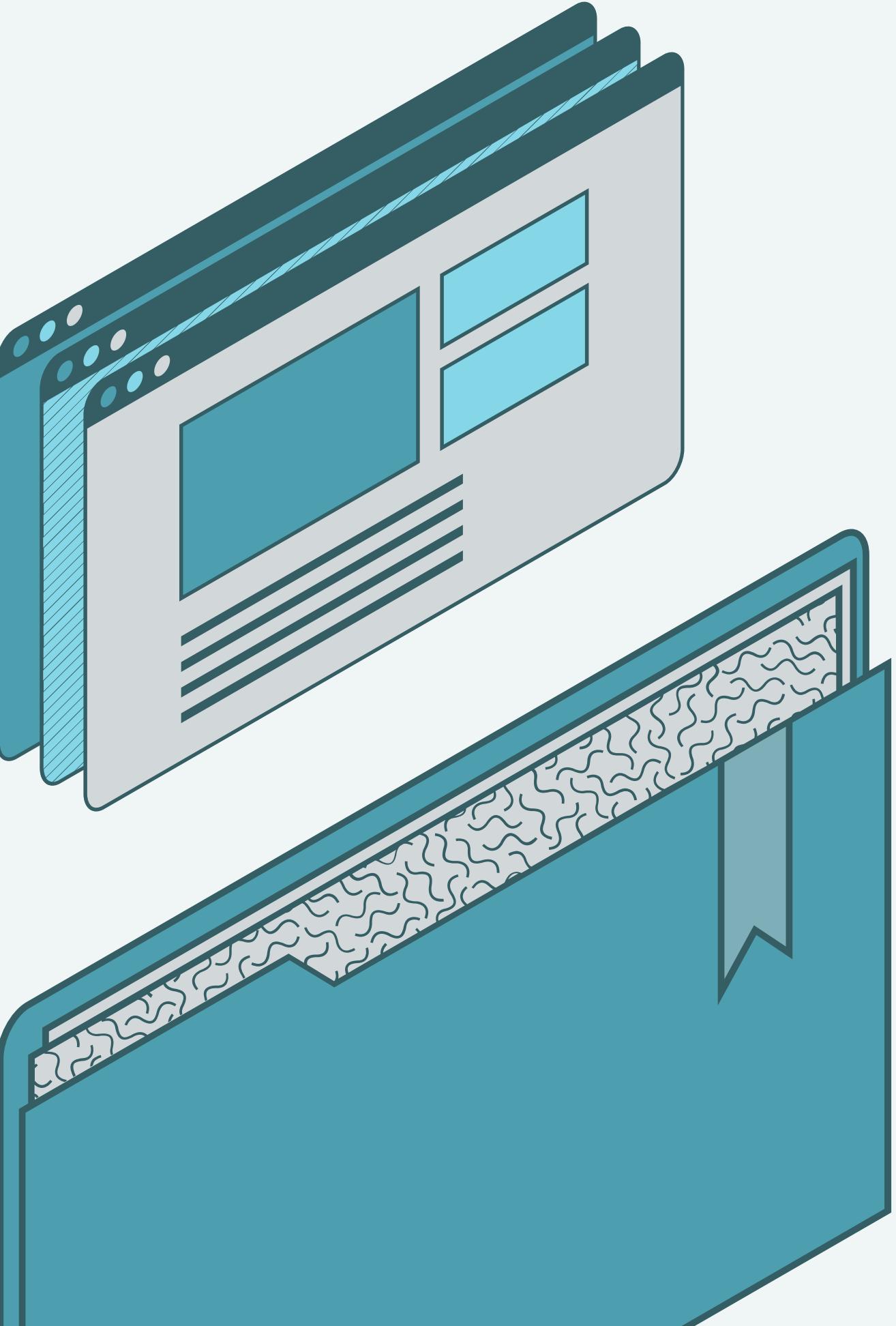
- ¿Por qué? Esta separación simplifica la gestión, mejora la seguridad al aislar los datos del usuario y permite que el modelo y la aplicación principal escalen de forma independiente.

# TAREAS PENDIENTES Y PLAN DE ACCIÓN (PRÓXIMAS 4 SEMANAS)

¿Qué falta? La lógica de integración final y la gestión completa de usuarios.

- **Módulo de Usuarios:** La tarea prioritaria es implementar el sistema de Login y Registro.
- **Persistencia de Datos:** Desarrollar la funcionalidad para guardar y gestionar los datos de cada usuario (sus medidas, objetivos, alergias y los planes que se le han generado).
- **Integración Final:** Conectar los datos del usuario que iniciará sesión con el modelo para que las recomendaciones se muestren en el frontend.





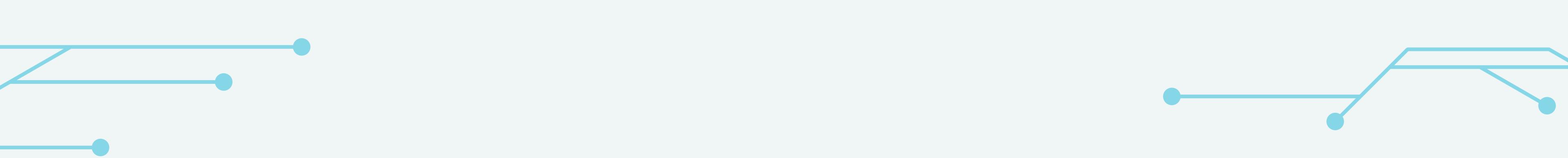
# REFLEXIÓN Y APRENDIZAJES

## 01. Principales Aprendizajes

- La importancia de priorizar el componente más complejo (el modelo), ya que desbloqueó el resto del proyecto.
- El valor de tomar decisiones de arquitectura pragmáticas, como separar las bases de datos, para simplificar el desarrollo y mantenimiento a largo plazo.
- Reforzamos conocimientos prácticos en Flask, Docker, y librerías de Machine Learning, aplicando la teoría a un caso de uso real.

## 02. ¿Qué haríamos diferente?

- "Si pudiéramos volver a empezar, definiríamos desde el principio un contrato de API claro entre el frontend y el backend. Esto nos habría permitido probar la integración mucho antes, incluso antes de que el frontend estuviera 100% terminado."



**MUCHAS  
GRACIAS**

