

# Prueba Senior SW Engineer <> Different Health

---

## **Contexto:**

Montar el frontal del chatbot y mejorar el del backend.

---

## **Frontend:**

Diseñar un frontal básico para el chatbot, dentro del "**Gadget**" chatbot.

➡ Usad los mismos patrones que hay en componentes reutilizables, como:

- Separación de estilos usando **CSS Modules (BEM)**
  - Carpetas de configuración organizadas
  - Código limpio y reutilizable
- 

## **Backend:**

Usar el código existente y **mejorarlo**.

La idea es poder preguntar cualquier **dato de salud, trends, insights**, de la mejor manera posible.

➡ Seguir los patrones actuales:

- Ruta
  - Controlador
  - Servicio
-

## Instrucciones:

- ❶ Crear un **cluster free de Mongo Atlas** (sino la búsqueda vectorial no funcionará) y añadir la URL en `Mongo_URL`
  - ❷ Añadir varias colecciones usando los JSON de ejemplo que hay en el servidor: **Dexa**, **Vo2Max**, **VHR**
    - Cread muchos más para poder hacer comparaciones entre años, tendencias, etc.
    - Si queréis crear más colecciones, adelante.
  - ❸ En el root del proyecto, instalar dependencias:  
`yarn install`
    - (Usar `--ignore-engines` en caso de incompatibilidad con vuestra versión de Node)
    - Si hay problemas, usad `nvm` y cambiad la versión a **18.3**
  - ❹ Arrancar frontend y backend
  - ❺ Usad **Copilot**, **Cursor** o **CodeWhisperer**, lo que necesitéis.
    - No perdáis tiempo en tareas manuales  
Cread una interfaz **sencilla pero elegante**
  - ❻ Aplicad **buenas prácticas** en todo momento 🍓
- 

## Qué se valorará:

- ✅ Calidad del código
- ✅ Limpieza y orden
- ✅ Calidad de la interfaz del frontend (sencilla, no os preocupéis por lo visual)
- ✅ Mejora del código backend:
  - Añadir **memoria**
  - **Optimización** para encontrar datos de forma efectiva
  - Soporte para **diversas colecciones**
  - Saber identificar **tendencias**

- Simular datos de **laboratorio** y **wearables** (cambian hora a hora)

---

🙏 ¡Muchas gracias!