TEST TECNICO SOFTWARE ENGINEER I

Bienvenido al desafío LiquiVerde de Grupo Lagos

🎯 Descripción del Desafío

Debes crear una plataforma de retail inteligente que ayude a los consumidores a ahorrar dinero mientras toman decisiones de compra sostenibles, optimizando presupuesto e impacto ambiental/social.

📱 Qué Debes Desarrollar

Una aplicación full-stack que incluya:

Backend API:

- Sistema de análisis de productos y sostenibilidad
- Optimización de listas de compras multi-criterio
- Cálculo de ahorros e impacto ambiental (bonus)
- Sistema de recomendaciones de sustitución (bonus)

Frontend Web/Móvil:

- Escáner de productos (código de barras o búsqueda)
- Dashboard de ahorros e impacto (bonus)
- Generador de listas de compras optimizadas
- Comparador de productos y alternativas (bonus)
- Mapa de tiendas y rutas eficientes (bonus)

Los requisitos marcados como bonus no son obligatorios, pero si te darán mayor puntaje al momento de realizar la evaluación. También puedes crear libremente features que consideres necesarias, así como APIs backend.

🛠️ Stack Tecnológico Permitido

Frontend (Debes elgir uno):

- React + Vite
- Angular

Backend (Debes elegir uno):

- Python (FastAPI/Flask/Django)
- Node.js (Express/NestJS)
- Go
- Java (Spring Boot)

Base de Datos (Elige una o las necesarias si lo requieres):

- SQLite
- PostgreSQL
- Redis (para cache)
- MongoDB
- Cassandra

Opcionales (Bonus):

- Docker + Docker Compose
- Despliegue en servidor gratuito
- PWA (Progressive Web App)

🌐 APIs Públicas a Utilizar

Puedes utilizar las que creas convenientes siempre y cuando sean publicas.

Datos de Productos:

- **Open Food Facts API**

  https://world.openfoodfacts.org/api/v0/product/{barcode}.json

  https://world.openfoodfacts.org/api/v2/search?countries=chile&categories=food

- **USDA FoodData Central**

  https://api.nal.usda.gov/fdc/v1/foods/search?api_key=YOUR_API_KEY

Precios y Comercio:

- **Tesco API** (Ejemplo - usar alternativa disponible)

  https://dev.tescolabs.com/grocery/products/?query={product}

- **Carbon Footprint API**

  https://www.carboninterface.com/api/v1/estimates

Geolocalización:

- **OpenStreetMap Nominatim**

  https://nominatim.openstreetmap.org/search?format=json&q={address}

🧠 **Algoritmos a Implementar**

Obligatorios (mínimo 2):

1. **Algoritmo de Mochila Multi-objetivo** para optimización de lista de compras
2. **Sistema de Scoring de Sostenibilidad** (económico, ambiental, social)
3. **Algoritmo de Sustitución Inteligente** de productos

Opcionales (Bonus):

- Algoritmo de Planificación Temporal de Compras
- Optimización de Rutas de Tiendas
- Sistema de Recompensas por Sostenibilidad

📦 Entregables Obligatorios

1. **Repositorio Git** (GitHub/GitLab) con:
    - Código fuente completo
    - README.md con instrucciones de despliegue
    - Dataset de ejemplo con productos
2. **README.md debe incluir:**
    - Instrucciones claras para ejecutar localmente
    - Configuración de APIs y variables de entorno
    - Explicación de algoritmos implementados
    - **Sección "Uso de IA"** detallando asistencia recibida
3. **Aplicación Funcional** que cumpla con:
    - Análisis de productos escaneados/buscados
    - Generación de listas optimizadas
    - Cálculo de ahorros e impacto

🏆 Criterios de Evaluación

Funcionalidad (50%):

- Análisis de productos funcionando
- Generación de listas optimizadas
- Cálculo de ahorros preciso (bonus)
- Interfaz de usuario clara

Algoritmos (30%):

- Implementación correcta de optimización
- Cálculos de sostenibilidad
- Calidad del código algorítmico

Calidad de Código (20%):

- Estructura y organización
- Manejo de errores
- Documentación

Bonus (+10%):

- Dockerización
- Despliegue en cloud
- Tests
- Funcionalidades extra

💡 **Tips y Recomendaciones**

- **Enfócate en el core**: Optimización de compras es lo principal
- **Usa datos realistas**: Crea un dataset de ejemplo convincente
- **Explica tus fórmulas**: Documenta cómo calculas sostenibilidad y ahorro
- **Piensa en el usuario**: La interfaz debe hacer complejidad simple

¡La innovación en sostenibilidad y ahorro será altamente valorada!

📝 **Instrucciones entrega**

🕒 **Tiempo Estimado: 24 horas (8-12 horas de trabajo efectivo)**