

# Prueba Técnica – Backend Developer (Python/Go)

## Objetivo

Desarrollar una aplicación web estilo **Password Manager** (ejemplo: LastPass o 1Password), en la que un usuario pueda gestionar de manera segura sus credenciales.

El foco principal de la prueba está en el **backend** (API, seguridad, arquitectura, buenas prácticas). El frontend es opcional, pero se valorará si se incluye una interfaz básica para interactuar con la aplicación (puede ser en React u otra tecnología).

## Tecnologías

- **Backend obligatorio:** Python (Django/FastAPI/Flask) o Go (Gin/Fiber/Chi).
- **Base de datos:** SQLite o PostgreSQL.
- **Frontend opcional:** React u otra librería/framework de preferencia.
- **Extras (opcionales y valorados):**
  - Contenedorización con Docker/Docker Compose.
  - Despliegue en GitHub Codespaces o similar.

## Requisitos funcionales mínimos

1. **Autenticación con token (JWT o similar)**
  - Registro de usuario.
  - Login con emisión de token.
  - Un usuario no puede ver ni editar cuentas de otro.
2. **Gestión de contraseñas (Vault):**
3. Cada cuenta debe incluir los siguientes campos:
  - **username**
  - **password** (cifrada)
  - **url**
  - **notes**
  - **icon**
4. Se deben implementar operaciones **CRUD** (Create, Read, Update, Delete).
5. Seguridad:
  - La contraseña del usuario debe guardarse hasheada (bcrypt o similar).
  - Las contraseñas gestionadas deben almacenarse **cifradas**.

6. Recuperación de contraseña olvidada: Flujo para que un usuario pueda restablecer su contraseña mediante email/token.
7. Búsqueda y filtrado:
  - Búsqueda por texto libre en `username`, `url` o `notes`.
  - Filtrado (ejemplo: por dominio en la URL).
8. Estructura de páginas (mínimo):
  - Página/Login para acceso a la aplicación.
  - Página principal con listado de contraseñas.
  - Vista detallada de una contraseña.



## Entregables

- Código fuente en un repositorio **GitHub/GitLab**.
- **README.md** con:
  - Instrucciones claras de instalación y ejecución.
  - Dependencias necesarias.
  - Variables de entorno requeridas (ej: claves de cifrado, credenciales DB).
  - Ejemplos de endpoints y cómo probarlos (curl, Postman o frontend si existe).
- Implementación de un frontend básico (React, Vue, Angular, etc.) para consumir la API.
- **Deploy de la aplicación.**



## Criterios de evaluación

- Claridad, limpieza y organización del código.
- Correcta implementación de la funcionalidad requerida.
- Uso de buenas prácticas de seguridad.
- Optimización y rendimiento.
- Conocimiento del framework/librería backend elegido (Django, FastAPI, Go, etc.).
- Documentación clara (README + comentarios cuando sea necesario).



## Tiempo estimado

- **Duración máxima:** 1 semana.



## Extras valorados

- Tests automatizados (unitarios/integración).
- Uso de Docker para levantar el entorno completo.
- Despliegue en **GitHub Codespaces** o cualquier entorno cloud accesible.