**🧪 Prueba Técnica – Desarrollo de API REST + Interfaz Web (HTML + Bootstrap)**

**🎯 Objetivo:**

Desarrollar una API REST utilizando **Python y Django REST Framework**, que permita gestionar un sistema básico de inventario. Además, se debe implementar una **interfaz web sencilla (HTML + Bootstrap)** con un **menú estilo dashboard** que consuma dicha API.

**📝 Requerimientos funcionales de la API:**

**1. Usuarios**

* Registro y autenticación de usuarios.
* Implementar **autenticación con token (TokenAuth o JWT)**.
* Solo usuarios autenticados pueden acceder a clientes, productos e inventario.

**2. Clientes**

* CRUD (crear, listar, actualizar y eliminar).
* Campos: nombre, correo electrónico, teléfono y dirección.

**3. Productos**

* CRUD.
* Campos: nombre, descripción, precio y categoría.

**4. Inventario**

* CRUD.
* Relación con producto.
* Campos: producto, cantidad en stock, fecha de ingreso y observaciones (opcional).

**🖥️ Requerimientos de la interfaz web (frontend):**

**1. Dashboard en HTML + Bootstrap**

* Crear un menú lateral (dashboard estilo admin).
* Opciones en el menú: Inicio, Clientes, Productos, Inventario, Cerrar sesión.
* Mostrar nombre del usuario logueado.

**2. Consumo de la API:**

* Listar clientes, productos e inventario en tablas HTML.
* Formularios básicos para crear nuevos registros (no se requiere validación avanzada).
* Autenticación: permitir login con email y contraseña desde un formulario, y guardar el token para consumir la API.

**3. No se requiere JavaScript avanzado, pero se valora uso básico para manejar llamadas AJAX con fetch() o axios.**

**🔐 Restricciones técnicas:**

* Backend en **Python 3.x**, **Django**, y **Django REST Framework**.
* Interfaz HTML sencilla (puede ser Django templates o estática).
* Uso de **Bootstrap 4 o 5** para el diseño del dashboard.
* Separación clara de backend (API) y frontend (aunque sea en el mismo proyecto).
* Documentación básica (README.md) con pasos para ejecutar el proyecto.

**✅ Criterios de evaluación:**

* Correcto diseño de modelos, relaciones y validaciones.
* Uso apropiado de DRF: serializers, viewsets, routers, permisos.
* Funcionalidad de la interfaz (dashboard funcional).
* Limpieza del código y buenas prácticas.
* Manejo de autenticación y seguridad básica.
* Facilidad de instalación y pruebas.

**📦 Entrega:**

* Subir el proyecto a un repositorio público (GitHub, GitLab, etc.) o compartir un archivo .zip.
* Incluir instrucciones claras en README.md para:
  + Crear entorno virtual.
  + Instalar dependencias (requirements.txt).
  + Correr migraciones y levantar el servidor.
  + Acceder al dashboard.
* Tiempo estimado: **8 a 12 horas** (libre según disponibilidad del candidato).