

PRUEBA TÉCNICA – DESARROLLADOR/A FULLSTACK

Entrega de la prueba técnica

La presente prueba debe ser desarrollada y entregada en un máximo de 2 días.

Como hoy es viernes, la fecha límite de entrega es el domingo a las 22:00 p.m.

Formato de entrega:

- Comprime el proyecto en un archivo .zip con el siguiente nombre:
Apellido_Nombre_Fullstack_GeneradorInteligente.zip
- Súbelo a un repositorio privado (GitHub o Google Drive)
- Graba un video mostrando la funcionalidad del mismo y sube a Google Drive
- Comparte el enlace con acceso habilitado a los correos: desarrollador.be@trigono.pe y **desarrollador.fe@trigono.pe**

1. Contexto de la aplicación bajo prueba

Vas a construir una plataforma web llamada “Generador Inteligente de Informes Empresariales”.

Esta herramienta está pensada para ayudar a empresas a transformar archivos de datos (CSV/Excel) en informes visuales y resúmenes automáticos que ayuden en la toma de decisiones empresariales.

El usuario autenticado podrá subir un archivo .csv con datos de ventas.

El sistema debe analizar los datos, generar gráficos, métricas clave y permitir la exportación del informe completo en PDF.

2. Objetivos

- Desarrollar un flujo completo Fullstack: autenticación, upload, procesamiento, dashboard e informe.
- Demostrar conocimientos sólidos en Django REST Framework, PostgreSQL, React, React Query y TypeScript.
- Aplicar buenas prácticas de diseño, arquitectura, seguridad y experiencia de usuario.
- Crear una experiencia avanzada de frontend moderna, profesional y reactiva.

3. Tareas a realizar

| Módulo | Descripción |
|-------------------|---------------------------------------------------------------------|
| Login / Registro | Permitir autenticación con JWT |
| Upload de archivo | Interfaz clara para subir un .csv de ventas (validación incluida) |
| Procesamiento | Backend lee el CSV, analiza los datos con Pandas y retorna insights |
| Dashboard | Visualización de: métricas clave, gráficas, resumen automatizado |
| Exportación PDF | Opción de descargar el informe en formato PDF estético y limpio |

4. Flujo completo que debe implementar el postulante

- o Login/Registro: el usuario inicia sesión.
- o Ventana principal: se muestra una bienvenida y opción para subir un archivo CSV.

- o Subida: el usuario sube un archivo con datos de ventas. Ejemplo:

| Portapapeles | | Fuente | | | | |
|--------------|--------------------------------------------------------------|--------|---|----|---|---|
| H8 | | X | ✓ | fx | | |
| | A | B | C | D | E | F |
| 1 | Fecha, Producto, Categoría, Cantidad, PrecioUnitario, Región | | | | | |
| 2 | 2024-05-01, Monitor, Electrónica, 2, 150, Lima | | | | | |
| 3 | 2024-05-01, Teclado, Electrónica, 1, 50, Arequipa | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |
| 6 | | | | | | |
| 7 | | | | | | |
| 8 | | | | | | |

Procesamiento, el backend calcula:

- ✓ Ventas totales
- ✓ Top productos
- ✓ Ventas por región
- ✓ Ventas por fecha
- ✓ Insight automático (ej. “Las ventas de abril cayeron un 20%”)

5. Dashboard

- Visualización con gráficas (line, bar, pie)
- Tarjetas de KPIs
- Texto generado automáticamente con resumen

6. Informe PDF:

- Se puede descargar el informe en PDF con los datos analizados y gráficos embebidos.

7. Requisitos Técnicos Obligatorios (Stack y dependencias)

- Backend: Django REST Framework, PostgreSQL, Pandas
- Frontend: React + Vite, TypeScript, Tailwind CSS, React Query

- Autenticación: JWT
- Visualización: Recharts, Chart.js o Plotly (Puedes usar las librerías con las que más cómodo se sientan)
- PDF: WeasyPrint, ReportLab o equivalente