

## PRUEBA TÉCNICA – FULL COLOMBIANO

### 1. OBJETIVO DE LA PRUEBA

Validar que el equipo de desarrollo posee las capacidades técnicas, prácticas y de arquitectura necesarias para construir el MVP del marketplace Full Colombiano utilizando Python + Django REST Framework en el backend y React en el frontend.

### 2. ALCANCE DE LA PRUEBA

El candidato deberá entregar un módulo funcional dividido en backend, frontend, integración con API pública y despliegue real en servidor, más un repositorio organizado.

### 3. PRUEBA BACKEND – DJANGO REST FRAMEWORK (Obligatoria)

#### Tareas:

- Crear proyecto Django con app base.
- **Modelos:** User (o extendido), Vendor, Product.
- Endpoints: Registro, Login (Token o JWT), CRUD de productos, productos por vendedor.
- **Permisos:** solo el vendedor autenticado puede crear y editar sus productos.
- Documentación de API: Swagger o Postman.

### 4. PRUEBA FRONTEND – REACT (Obligatoria)

#### Tareas:

- Pantalla de Login (consume API).
- Listado de productos desde el backend.
- Formulario para crear producto y enviarlo al backend.
- Manejo de errores y estados de carga.

### 5. PRUEBA DE INTEGRACIÓN CON API PÚBLICA (Obligatoria)

#### Objetivo:

Validar la capacidad del candidato para integrar servicios externos, fundamental para futuras integraciones con pasarelas de pago, envíos, etc.

#### Tareas:

- Consumir una API pública (Rick & Morty, Pokémon, OpenWeather, u otra aprobada).
- Crear una vista React que muestre los datos procesados.
- Manejar estados de carga y error.

#### Entregables:

- Código en el mismo repositorio.
- Pantalla funcional mostrando la integración.

## 6. PRUEBA DE DESPLIEGUE EN SERVIDOR REAL (Obligatoria)

### Objetivo:

Validar que el candidato puede llevar un proyecto a producción, configurar entornos y asegurar comunicación entre frontend y backend.

### Tareas Backend:

- Desplegar el backend Django REST en Railway, Render, Fly.io o similar.
- Exponer la API vía URL pública.
- Configurar variables de entorno.

### Tareas Frontend:

- Desplegar React en Vercel, Netlify, Render o similar.
- Conectar el frontend desplegado con la API desplegada.

### Entregables:

- URL pública del backend.
- URL pública del frontend.
- README explicando brevemente el proceso de despliegue.

## 7. ENTREGABLES OBLIGATORIOS GENERALES

- Repositorio GitHub en la cuenta del cliente.
- README con instrucciones de instalación, ejecución y despliegue.
- Archivo Postman o Swagger.

## 8. CRITERIOS DE EVALUACIÓN

**Backend (35%):** organización, uso de DRF, modelo de datos, permisos y autenticación.

**Frontend (25%):** consumo adecuado de API, organización de componentes, manejo de errores.

**Integración API pública (20%):** consumo de API externa, procesamiento, limpieza del código.

**Despliegue real (20%):** API y frontend funcionando, configuración de entornos, claridad de explicación.