

## Prueba Técnica – Ingeniero de Desarrollo Back End

Fecha: 20/06/2025

Empresa: BTG Pactual

### Parte 1 – Fondos (80%)

#### Necesidad de negocio:

BTG Pactual desea crear una plataforma que permita a los clientes gestionar sus fondos de inversión sin necesidad de contactar a un asesor. Las funcionalidades requeridas son:

#### Funcionalidades del sistema:

1. Suscribirse a un nuevo fondo (apertura).
2. Cancelar la suscripción a un fondo actual.
3. Ver historial de transacciones (aperturas y cancelaciones).
4. Enviar notificación por email o SMS según preferencia del usuario al suscribirse a un fondo.

#### Reglas de negocio:

Monto inicial del cliente: COP \$500.000.

Cada transacción debe tener un identificador único.

Cada fondo tiene un monto mínimo de vinculación.

Al cancelar una suscripción, el valor de vinculación se retorna al cliente.

Si no hay saldo suficiente, mostrar:

“No tiene saldo disponible para vincularse al fondo <Nombre del fondo>”

#### Información de los fondos:

ID	Nombre	Monto mínimo	Categoría
1	FPV_BTGPACTUAL_RECAUDADORA	COP \$75.000	FPV
2	FPV_BTGPACTUAL_ECOPETROL	COP \$125.000	FPV
3	DEUDAPRIVADA	COP \$50.000	FIC
4	FDO-ACCIONES	COP \$250.000	FIC
5	FPV_BTGPACTUAL_DINAMICA	COP \$100.000	FPV

#### Actividades solicitadas:

1. Tecnologías sugeridas: Python con FastAPI (Deseable) o .NET 9
2. Diseñar un modelo de datos NoSQL que soporte las operaciones.
3. Construir una API REST que implemente las funcionalidades descritas.
4. Incluir:
  - Manejo de excepciones.
  - Código limpio (Clean Code).
  - Pruebas unitarias.
  - Buenas prácticas de seguridad y mantenibilidad.
5. Definir los procesos de autenticación, autorización, perfilamiento por roles y encriptación.
6. Despliegue: El backend debe poder desplegarse mediante AWS CloudFormation usando ServerlessFramework o mediante Terraform, con documentación incluida.
7. Deseable que toda la infraestructura y los recursos utilizados estén desplegados en AWS.
8. Justificar la arquitectura utilizada para la solución junto con diagrama
9. Entregar colección de postman para la ejecución

## Parte 2 – SQL (20%)

### Base de datos: BTG

Escriba las consultas SQL correspondientes, para ello, tenga en cuenta la base de datos llamada “BTG” la cual tiene las siguientes tablas (tenga en cuenta que se puede presentar el caso de que no todas las sucursales ofrecen los mismos productos).

Tablas disponibles:

Cliente			
id	nombre	apellidos	ciudad
number	varchar	varchar	varchar
PK	NN	NN	NN

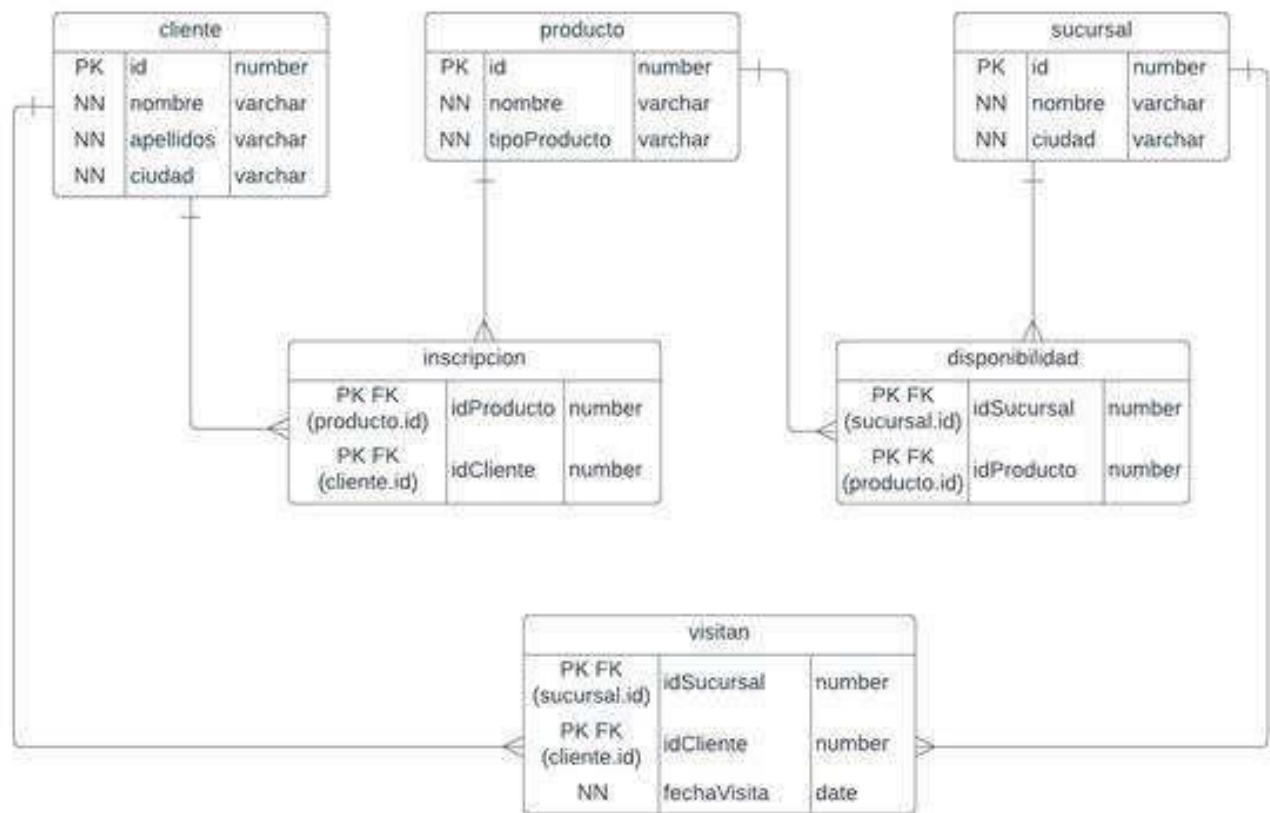
Sucursal		
id	nombre	ciudad
number	varchar	varchar
PK	NN	NN

Producto		
id	nombre	tipoProducto
number	varchar	varchar
PK	NN	NN

Inscripción	
idProducto	idCliente
number	number
PK FK(Producto.id)	PK FK(Cliente.id)

Disponibilidad	
idSucursal	idProducto
number	number
PK FK(Sucursal.id)	PK FK(Producto.id)

Visitan		
idSucursal	idCliente	fechaVisita
number	number	Date
PK FK(Sucursal.id)	PK FK(Cliente.id)	NN



#### Scripts solicitados:

- Teniendo en cuenta que esto va a ser una base de datos PostgreSQL, crear los archivos para crear esta base de datos, crear schema, tablas, constantes y relaciones entre tablas (FK)
- Obtener los nombres de los clientes que tienen inscrito algún producto disponible solo en las sucursales que visitan.