

# Challenge

Trabajamos con empresas del sector logístico internacional que enfrentan un reto común: la información clave sobre sus operaciones llega desordenada y en distintos momentos.

Nuestros clientes reciben datos de bookings, contenedores, órdenes de compra y facturas, pero rara vez toda la información está disponible desde el inicio. A veces solo se recibe una orden de compra, otras veces un contenedor, y solo con el tiempo se puede conectar todo.

El desafío es diseñar una solución que permita a nuestra plataforma interpretar, organizar y relacionar esa información de forma progresiva, ayudando a nuestros clientes a tener una vista clara y confiable de lo que está pasando en su operación logística.

Debes diseñar una API Rest que permita:

1. Recibir información logística relacionada (contenedores, bookings, órdenes de compra e invoices).
2. Mantener consistencia de relaciones entre entidades, evitando duplicados o cruces de datos entre clientes.
3. Consultar estas entidades de forma individual y mostrar la data relacionada

## Ingreso de datos

### POST /api/email

```
{
  "booking": "BK123",
  "containers": [
    { "container": "MEDU1234567" }
  ],
  "orders": [
    {
      "purchase": "P0123",
      "invoices": [{ "invoice": "IN123" }]
    }
  ]
}
```

Es posible solo recibir la información de las órdenes o solo de los contenedores.

## Endpoints de consulta

### GET /api/orders

Devuelve todas las órdenes de compra registradas por un cliente.

### GET /api/containers

Devuelve todos los contenedores registrados por un cliente.

### GET /api/orders/{purchaseId}/containers

Devuelve todos los contenedores asociados a una orden de compra específica.

### GET /api/containers/{containerId}/orders

Devuelve todas las órdenes de compra asociadas a un contenedor específico.

## Requisitos técnicos

- Buenas prácticas y estándares de código
- Pruebas unitarias
- Seguridad en la gestión de datos (protección de información, validación de entradas, aislamiento de clientes)
- Implementación de un sistema de persistencia de datos
- Optimizaciones y eficiencia en las consultas
- Resiliencia en el sistema