

1. Objetivo de la prueba

El objetivo de esta prueba es evaluar las competencias del candidato en la construcción de **APIs REST orientadas a flujos de autorización**, utilizando **Python y FastAPI**, con énfasis en:

- Modelado de procesos con múltiples estados.
 - Implementación de reglas de negocio dependientes de rol.
 - Manejo consistente de transiciones de estado.
 - Validaciones, errores y control de concurrencia lógica.
 - Calidad, claridad y mantenibilidad del código.
-

2. Contexto del negocio (Escenario Operador – Aprobador)

La institución ofrece a sus clientes empresariales la posibilidad de ejecutar **transacciones financieras sensibles** (por ejemplo, transferencias o pagos a proveedores) bajo un esquema de **segregación de funciones**.

En este modelo:

- Un **Operador** registra la transacción.
- Un **Aprobador** revisan y autorizan la operación antes de su ejecución final.

El sistema debe asegurar que:

- Una transacción no pueda ejecutarse sin la aprobación correspondiente.
- Las acciones estén correctamente validadas según el rol.
- El estado de la transacción refleje claramente su situación dentro del flujo.

Para esta prueba, **no se ejecutarán movimientos reales de dinero**; la ejecución será simulada.

3. Alcance técnico

- **Lenguaje:** Python
- **Framework:** FastAPI
- **Persistencia:**
 - En memoria (diccionarios) o
 - SQLite, Postgre (opcional, valorado positivamente)
- **Autenticación real:** No requerida (el rol se enviará por header)
- **Testing automático:** Opcional
- **Entrega:** Repositorio Git o archivo comprimido (.zip)

El flujo debe ser **completamente testeable vía Postman**.

4. Modelo de dominio

Entidad: Transaction

Campo	Tipo	Descripción
transaction_id	UUID	Identificador único
reference	string	Referencia de la transacción
amount	decimal	Monto
currency	string	Moneda
status	enum	DRAFT, PENDING_APPROVAL, APPROVED, REJECTED, EXECUTED
created_by	string	Identificador del operador
approved_by	string	Identificador del aprobador
created_at	datetim e	Fecha de creación
updated_at	datetim e	Fecha de última actualización

5. Reglas de negocio

- 1. Creación de la transacción**
 - Solo un usuario con rol **OPERADOR** puede crear transacciones.
 - La transacción se crea con estado **DRAFT**.
 - El monto debe ser mayor a cero.
 - La moneda debe tener exactamente 3 caracteres.
- 2. Envío a aprobación**
 - Solo el **OPERADOR** puede enviar la transacción a aprobación.
 - El estado cambia de **DRAFT** a **PENDING_APPROVAL**.
- 3. Aprobación**
 - Solo un usuario con rol **APROBADOR** puede aprobar.
 - Solo se pueden aprobar transacciones en estado **PENDING_APPROVAL**.
 - Al aprobar, el estado pasa a **APPROVED** y se registra el aprobador.

4. Rechazo

- Solo un **APROBADOR** puede rechazar.
- El estado pasa a **REJECTED**.

5. Ejecución simulada

- Solo transacciones en estado **APPROVED** pueden ejecutarse.
 - Al ejecutar, el estado cambia a **EXECUTED**.
 - La ejecución es simulada, sin integración externa.
-

6. Endpoints requeridos

6.1 Crear transacción

POST /transactions

Headers:

```
None  
X-User-Role: OPERADOR  
X-User-Id: op-123
```

Body:

```
None  
{  
  "reference": "TRX-001",  
  "amount": 5000,  
  "currency": "MXN"  
}
```

Respuestas:

- **201 Created:** Transacción creada.
 - **403 Forbidden:** Rol no permitido.
 - **400 Bad Request:** Error de validación.
-

6.2 Enviar a aprobación

POST /transactions/{transaction_id}/submit

Headers:

None

X-User-Role: OPERADOR

Respuestas:

- **200 OK:** Enviada a aprobación.
 - **409 Conflict:** Estado inválido.
 - **404 Not Found:** Transacción inexistente.
-

6.3 Aprobar transacción

POST /transactions/{transaction_id}/approve

Headers:

None

X-User-Role: APROBADOR

X-User-Id: ap-456

Respuestas:

- **200 OK:** Transacción aprobada.
 - **403 Forbidden:** Rol no permitido.
 - **409 Conflict:** Estado inválido.
-

6.4 Rechazar transacción

POST /transactions/{transaction_id}/reject

Headers:

None

X-User-Role: APROBADOR

Respuestas:

- **200 OK:** Transacción rechazada.
 - **403 Forbidden:** Rol no permitido.
 - **409 Conflict:** Estado inválido.
-

6.5 Ejecutar transacción

POST /transactions/{transaction_id}/execute

Respuestas:

- **200 OK:** Transacción ejecutada.
 - **409 Conflict:** Estado inválido.
 - **404 Not Found:** Transacción inexistente.
-

6.6 Consultar transacción

GET /transactions/{transaction_id}

Respuesta:

```
None
{
    "transaction_id": "uuid",
    "reference": "TRX-001",
    "amount": 5000,
    "currency": "MXN",
    "status": "APPROVED",
    "created_by": "op-123",
    "approved_by": "ap-456"
}
```

7. Criterios de evaluación

Criterios obligatorios

- Uso correcto de FastAPI y Pydantic.
- Implementación clara del flujo Operador – Aprobador.
- Validaciones por rol y estado.
- Manejo adecuado de errores y códigos HTTP.
- API funcional y testeable vía Postman.

Criterios de seniority (valor agregado)

- Modelado limpio de la máquina de estados.
- Separación de capas (API, servicio, dominio).
- Uso de enums y tipado estricto.
- Manejo de transiciones inválidas de forma explícita.

- README claro y conciso.
 - Pruebas automatizadas.
-

8. Entregables

1. Código fuente.
2. README con instrucciones de ejecución y ejemplos.

ENVIAR ENTREGABLE A: adan@covalto.com y dbonansea@covalto.com con copia a jforero_deel@covalto.com.