

🔒 Secure Payment Service Challenge

Este challenge técnico consiste en construir un microservicio para manejar transferencias entre cuentas bancarias. El objetivo es evaluar diseño, manejo de concurrencia, testing, y buenas prácticas en servicios backend.

🔒 Requerimientos Funcionales

- Implementar un servicio de transferencias de dinero entre cuentas:
 - Cada transferencia debe comenzar en estado `PENDING`.
 - El servicio debe responder inmediatamente con ese estado al iniciar la operación.
 - Debe existir una forma de recibir una notificación externa (webhook) con el estado final de la transferencia (`COMPLETED` o `FAILED`).
 - El backend debe actualizar el estado de la transferencia al recibir esa notificación.
 - Si la notificación no llega, el servicio debe implementar alguna estrategia alternativa para actualizar el estado (ej. timeout, verificación periódica, etc.).
 - Es importante que se aplique algun mecanismo de transaccionalidad para evitar inconsistencias en las modificaciones a los saldos de las cuentas.
 - Se debe poder consultar el estado de una transferencia y el saldo de cada cuenta.
 - Toda operación sensible debe requerir autenticación (token simple o JWT).
 - Se espera una cobertura básica con pruebas unitarias.
-

🔒 Bonus

- Logging estructurado.
 - Retries en caso de errores.
 - Dockerfile para levantar servicio y DB.
 - Métricas básicas (`/metrics`).
 - CI pipeline opcional.
-

🔒 Stack sugerido

- Lenguaje: Go
 - DB: PostgreSQL (puede simularse con SQLite o memoria).
 - Framework web: libre.
 - Testing: `go test`.
-

🔒 Entrega

Subí el código a un repo público. Incluí un `README` breve si tomás decisiones arquitectónicas relevantes o si implementás extras.

🔒 Qué evaluamos

- Claridad del código y estructura.
 - Concurrencia y consistencia de datos (saldos de cuentas y estados de transferencias).
 - Seguridad básica.
 - Testing y cobertura.
 - Observabilidad/logs.
 - Simplicidad + elegancia.
-

¡Éxitos y que te diviertas programando! 🚀