

Situacion 4. Cajero automático — Análisis, Diseño lógico y Diagrama de flujo

1. Análisis del problema

Requisitos funcionales mínimos para corregirlo:

1. Limitar intentos de PIN (ej. 3) y bloquear la tarjeta o sesión tras exceder el límite.
2. Implementar tiempo de espera (opcional) o bloqueo temporal.
3. Registrar intentos (éxitos y fallos) con timestamp.
4. Verificar saldo antes de autorizar retiro.
5. Asegurar que la opción "Consultar saldo" siempre responda.
6. Validar entradas de usuario (monto positivo, opciones válidas).
7. (Si aplica) Concurrencia: proteger operaciones críticas (p. ej. con locks).

2. Diseño lógico

Variables principales:

- PIN_CORRECTO (4 dígitos)
- intentos = 0
- LIMITE_INTENTOS = 3
- saldo (entero en centavos o unidades mínimas)
- opcion, monto

proceso

1. **Pedir PIN al usuario.**
2. Si $PIN == PIN_CORRECTO$: registrar éxito y pasar al menú.
3. Si $PIN \neq PIN_CORRECTO$: registrar fallo, incrementar intentos.
4. Si $intentos \geq LIMITE_INTENTOS$: bloquear tarjeta (o terminar sesión) y Fin.
5. Si $intentos < LIMITE_INTENTOS$: repetir paso 2.1 (permitir nuevo intento).

2. Menú principal

1. Mostrar opciones: 1) Consultar saldo, 2) Retirar dinero, 3) Salir.
2. Si opción == 1: mostrar saldo. Volver a menú.
3. Si opción == 2: pedir monto.
 - Validar monto > 0. Si no válido, mostrar error y volver a menú.
 - Si monto <= saldo, entonces saldo -= monto — confirmar retiro y registrar transacción. Volver a menú.
 - Si monto > saldo, mostrar "Saldo insuficiente". Volver a menú.
4. Si opción == 3: terminar sesión y Fin.
5. Si opción inválida: mostrar "Opción inválida" y volver a menú.
6. Fin. Mostrar mensaje de despedida.

3. Diagrama de flujo

