**Caso de Uso — Tratamento de Operações Idempotentes (IdOperacao)**

**1) Identificação**

* **ID:** UC-SEG-003
* **Nome:** Idempotência de Operações (IdOperacao)
* **Objetivo:** Garantir que **reenvios** (retries) de operações de **movimentação financeira** não gerem **efeitos duplicados**, devolvendo a **mesma resposta** quando o IdOperacao já tiver sido processado.

**2) Atores**

* **Primário:** Cliente (app/web) — em chamadas de Depósito, Saque e Transferência Interna.
* **Secundário:** Serviços internos (API de Autenticação, DB).

**3) Pré-condições**

* Endpoints de **movimentação** exigem IdOperacao (UUID v4) no **header**.
* “Loja de Idempotência” (tabela/armazenamento dedicado) disponível.
* Banco transacional disponível.

**4) Disparador**

* Chega uma solicitação **mutável** (ex.: POST depósito/saque/transferência) com IdOperacao.

**5) Regras de Negócio (MVP)**

* **Obrigatório nos endpoints de movimentação:** IdOperacao no header.
* **Formato:** IdOperacao deve ser **UUID** válido.
* **Escopo da chave:** por **usuário** (ou cliente) e **endpoint**, para evitar colisões cruzadas.
  + Chave composta recomendada: (UsuarioId, RotaNormalizada, IdOperacao).
* **Fingerprint do request:** registrar Método, Rota, BodyHash (ex.: SHA-256 do corpo normalizado), Conta(s) envolvida(s).
  + Se o mesmo IdOperacao chegar com **Fingerprint diferente** ⇒ **409 CONFLITO\_IDEMPOTENCIA**.
* **Respostas em replay:**
  + Se **já processado com sucesso** ⇒ retornar **mesma resposta** (**200/201**) sem efeitos colaterais.
  + Se **em processamento** (registro “Processing”) ⇒ política do MVP: aguardar curto período e, se ainda “Processing”, retornar **200** com status: processing **ou** **409** (mantemos simples: **200 com status processing** é opcional; por padrão devolveremos a **última resposta persistida**).
* **Persistência do resultado:** salvar **status HTTP**, **payload de resposta** (ou referência), timestamps e **TTL**.
* **Dupla barreira (ledger):** em **Movimentacao**, manter **índice único** (ContaId, IdOperacao) — evita crédito/débito duplicado caso a “loja” falhe.
* **Retenção (TTL):** manter registros de idempotência por **72h** (MVP), purgar após expirado.
* **Auditoria:** registrar correlação (CorrelationId, se houver), quem, quando, rota e decisão (novo, replay, conflito).

**6) Fluxo Principal (Feliz)**

1. **Receber** requisição com IdOperacao.
2. **Consultar** a Loja de Idempotência por (UsuarioId, Rota, IdOperacao).
3. **Se não existir** registro:  
   3.1 **Criar** registro com status **Processing** (INSERT com **unique** em IdOperacao).  
   3.2 **Processar transação** (abrir **transação atômica** → registrar **Movimentacao** com (ContaId, IdOperacao) e **atualizar saldo**).  
   3.3 **Persistir** no registro de idempotência: HttpStatus, ResponseBody (ou ponteiro), Fingerprint, ConcluidoEm, Status=Succeeded.  
   3.4 **Retornar** a resposta de sucesso (**201/200**).
4. **Se existir** registro:  
   4.1 **Comparar** Fingerprint.  
   4.2 **Se igual** ⇒ **retornar resposta armazenada** (**200/201**) sem reprocessar.  
   4.3 **Se diferente** ⇒ **409 CONFLITO\_IDEMPOTENCIA**.

**7) Fluxos Alternativos / Exceções**

* **A1 — Header ausente:** **400 Bad Request** IDOPERACAO\_OBRIGATORIO.
* **A2 — UUID inválido:** **422 Unprocessable Entity** IDOPERACAO\_INVALIDO.
* **A3 — Conflito de fingerprint (mesmo IdOperacao com payload diferente):** **409 Conflict** CONFLITO\_IDEMPOTENCIA.
* **A4 — “Loja” indisponível:**
  + **503 Service Unavailable** IDEMPOTENCIA\_INDISPONIVEL **OU** degradar para “dupla barreira”: segue processamento confiando no **unique (ContaId, IdOperacao)** da tabela **Movimentacao** e depois tenta **best-effort** salvar a resposta.
* **A5 — Queda entre débito/crédito e gravação da loja:**
  + Reenvio encontrará **Movimentacao** com (ContaId, IdOperacao) e **reconstruirá** a resposta ⇒ retorna **200/201** consistente.
* **A6 — Persistência falhou:** **500 ERRO\_PERSISTENCIA** (sem efeitos duplicados por causa da dupla barreira).

**8) Pós-condições**

* Registro de idempotência criado/atualizado com resultado final.
* Em operações financeiras, **nenhum efeito duplicado** para o mesmo IdOperacao.
* Auditoria registrada.

**9) Dados Persistidos (rascunho)**

* **IdempotencyStore**
  + UsuarioId (FK)
  + Rota (string normalizada, ex.: /api/contas/{id}/saques)
  + IdOperacao (UUID) **[unique por (UsuarioId, Rota, IdOperacao)]**
  + Metodo (ex.: POST)
  + BodyHash (SHA-256 do JSON normalizado)
  + ContaOrigemId?, ContaDestinoId? (quando aplicável)
  + Status (Processing | Succeeded | FailedNonRetryable | FailedRetryable)
  + HttpStatus (int)
  + ResponseRef **ou** ResponseBody (texto/binário compactado)
  + CriadoEm, ConcluidoEm?, ExpiresAt (TTL)
  + CorrelationId?
* **Movimentacao (já definido nos UCs)**
  + **Unique** (ContaId, IdOperacao) — “segunda trava”.

**10) Interface/Contrato de API (padrão de headers)**

* **Header obrigatório nos POST de movimentação:**
  + IdOperacao: "<uuid-v4>"
* **Headers recomendados:**
  + X-Correlation-Id: "<uuid>"
* **Exemplo (Depósito):**

POST /api/contas/{id}/depositos

IdOperacao: 2a1b3c4d-5e6f-7081-92a3-b4c5d6e7f809

Content-Type: application/json

{ "valor": 150.75, "descricao": "Depósito via app" }

Reenvio com o **mesmo** body ⇒ **mesma resposta** (**201/200**).  
Reenvio com body **diferente** ⇒ **409 CONFLITO\_IDEMPOTENCIA**.

**11) Critérios de Aceitação (Given–When–Then)**

1. **Primeiro envio**
   * Given IdOperacao válido e payload válido
   * When POST de movimentação
   * Then **201/200** e registro em IdempotencyStore com Status=Succeeded.
2. **Reenvio igual (retry)**
   * Given mesmo IdOperacao e **mesmo** payload
   * When POST novamente
   * Then **200/201** com **mesma resposta**, sem novo lançamento.
3. **Reenvio diferente (conflito)**
   * Given mesmo IdOperacao e payload **diferente**
   * Then **409 CONFLITO\_IDEMPOTENCIA**.
4. **Falha entre lançar o movimento e salvar na loja**
   * Given transação financeira concluída
   * When reenvio com mesmo IdOperacao
   * Then sistema detecta pela **Movimentacao** e **reconstrói** a resposta (200/201).
5. **Header ausente**
   * Given POST de movimentação sem IdOperacao
   * Then **400 IDOPERACAO\_OBRIGATORIO**.

**12) Observações de Segurança e Privacidade**

* **Não** armazenar ResponseBody com dados sensíveis em texto puro; preferir **hash + pointer** ou **cifrar**.
* BodyHash impede leitura do corpo; não grava PII desnecessária.
* TTL curto (ex.: **72h**) reduz superfície de dados.
* Logs mínimos: mascarar IDs e truncar payloads.

**13) Métricas & Logs (MVP)**

* % de requisições com IdOperacao, taxa de **replay** atendido na loja, taxa de **409** por conflito, latência média.
* Alarmes para quedas da loja e **picos de replays** (rede instável).

**14) TODOs / Futuro**

* **[TODO-IDEMP-01]** Definir **TTL** oficial (24h vs 72h) por endpoint.
* **[TODO-IDEMP-02]** Expor endpoint interno de **consulta** a IdempotencyStore por IdOperacao (suporte/observabilidade).
* **[TODO-IDEMP-03]** Estratégia de **criptografia** do ResponseBody e **redação** de PII.
* **[TODO-IDEMP-04]** Suporte a status **Processing** com **202 + Retry-After** para workloads longos.
* **[TODO-IDEMP-05]** Rate limiting quando detectar **excesso de colisões** por IdOperacao por cliente.
* **[TODO-IDEMP-06]** Job de **limpeza** (purgar por ExpiresAt) e índices de manutenção.