**Caso de Uso — Depósito**

**1) Identificação**

* **ID:** UC-MOV-001
* **Nome:** Depósito em Conta
* **Objetivo:** Creditar um valor em uma conta **Ativa**, registrando o movimento e atualizando o saldo de forma **atômica** e **idempotente**.

**2) Atores**

* **Primário:** Cliente autenticado (app/web) realizando depósito **na própria conta**.
* **Secundário (opcional):** Operador/Administrador (canal interno).

**3) Pré-condições**

* Sistema e banco de dados disponíveis.
* Usuário autenticado e autorizado (**Depositar** para a própria conta; ou role interna).
* Conta existente.
* **Idempotência** via IdOperacao (header) definida.

**4) Disparador**

* Solicitação de depósito informando contaId e valor (> 0).

**5) Regras de Negócio e Validações (MVP)**

* **Status da conta:** deve estar **Ativa**.
  + Inativa, Bloqueada ou Encerrada → **não permite** depósito.
* **Valor:** valor > 0 (duas casas decimais, BRL).
* **Moeda:** conta em "BRL"; depósito considerado em BRL no MVP.
* **Idempotência:**
  + Se IdOperacao já processado para a **mesma conta** e **mesmo payload**, retornar a **mesma resposta** (200/201).
  + Se IdOperacao reaparecer com **valor diferente**, retornar **409 CONFLITO\_IDEMPOTENCIA**.
  + Recomenda-se **índice único** em (ContaId, IdOperacao) em **Movimentacao**.
* **Transação atômica:** registrar **Movimentacao** e atualizar **Saldo** **no mesmo commit**.
  + Usar *row-level locking* e/ou **UPDATE atômico** para evitar condições de corrida.
* **Auditoria:** registrar IdOperacao, CorrelationId (se enviado), canal, timestamp, usuário.

**6) Fluxo Principal (Feliz)**

1. Ator envia requisição com contaId, valor e IdOperacao.
2. Sistema valida existência da conta e **Status = Ativa**.
3. Sistema valida valor > 0 e formato.
4. Sistema verifica idempotência (ContaId + IdOperacao).
5. Sistema inicia **transação**.
6. Sistema registra **Movimentacao** do tipo **Depósito** (crédito), com SaldoAnterior e SaldoPosterior.
7. Sistema atualiza o **Saldo** da conta (saldo = saldo + valor).
8. Sistema confirma a transação, registra auditoria.
9. Sistema retorna **201 Created** com os dados do movimento e saldo atualizado.

**7) Fluxos Alternativos / Exceções**

* **A1 — Conta inexistente:**  
  2a. **404 Not Found** CONTA\_NAO\_ENCONTRADA.
* **A2 — Conta não ativa (Inativa/Bloqueada/Encerrada):**  
  2b. **409 Conflict** CONTA\_NAO\_ATIVA.
* **A3 — Valor inválido (≤ 0 ou formato inválido):**  
  3a. **422 Unprocessable Entity** VALOR\_INVALIDO.
* **A4 — Idempotência (reenvio igual):**  
  4a. **200 OK/201 Created** com a **mesma resposta** (sem novo crédito).
* **A5 — Idempotência (reenvio divergente):**  
  4b. **409 Conflict** CONFLITO\_IDEMPOTENCIA.
* **A6 — Falha de persistência:**  
  8a. **500 Internal Server Error** ERRO\_PERSISTENCIA.
* **A7 — Permissão insuficiente:**  
  0a. **403 Forbidden** PERMISSAO\_NEGADA.

**8) Pós-condições**

* Movimento de **Depósito** gravado.
* **Saldo** da conta atualizado.
* Auditoria registrada (IdOperacao, CorrelationId, usuário, canal).

**9) Dados Persistidos (rascunho)**

* **Movimentacao:**
  + Id (GUID)
  + ContaId (GUID, FK)
  + Tipo (enum: **Deposito**)
  + Valor (decimal(18,2))
  + SaldoAnterior (decimal(18,2))
  + SaldoPosterior (decimal(18,2))
  + Descricao (string, opcional)
  + IdOperacao (GUID/string) **[único por ContaId]**
  + CorrelationId (GUID/string, opcional)
  + CriadoEm (datetime), CriadoPor (string/userId)
* **Conta:**
  + Saldo atualizado
  + AtualizadoEm, AtualizadoPor (opcional)
* **Índices/Restrições:**
  + **Unique** (ContaId, IdOperacao) em **Movimentacao**.
  + Índice por ContaId + CriadoEm (consulta de extrato).

**10) Interface/Contrato de API (MVP)**

* **Endpoint:** POST /api/contas/{id}/depositos
* **Headers:**
  + IdOperacao: "<uuid>" *(obrigatório para idempotência)*
  + X-Correlation-Id: "<uuid>" *(recomendado)*
* **Request (exemplo):**

{

"valor": 150.75,

"descricao": "Depósito via app"

}

* **Response 201 (exemplo):**

{

"movimentoId": "c2e0a6b5-8a6a-4c8d-9b1a-0f98a2e7f111",

"contaId": "7f6a2e34-1c67-4b1a-9d23-1f23a1b9c0e3",

"tipo": "Deposito",

"valor": 150.75,

"saldoAnterior": 349.25,

"saldoPosterior": 500.00,

"criadoEm": "2025-08-15T01:05:00Z"

}

* **Erros comuns:** 403, 404, 409, 422, 500.

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

1. **Depósito válido**
   * Given conta **Ativa** e valor > 0
   * When POST /api/contas/{id}/depositos com IdOperacao
   * Then **201** com movimento criado e saldo atualizado.
2. **Conta não ativa**
   * Given conta Bloqueada/Inativa/Encerrada
   * Then **409 CONTA\_NAO\_ATIVA**.
3. **Valor inválido**
   * Given valor <= 0
   * Then **422 VALOR\_INVALIDO**.
4. **Idempotência (reenvio igual)**
   * Given mesmo IdOperacao e mesmo payload
   * Then **200/201** com **mesma resposta** e sem novo crédito.
5. **Idempotência (reenvio diferente)**
   * Given mesmo IdOperacao e **valor diferente**
   * Then **409 CONFLITO\_IDEMPOTENCIA**.

**12) Observações de Segurança e Privacidade**

* Não logar valores ou identificadores sensíveis em nível **INFO** sem mascarar.
* Somente **proprietário da conta** (ownership) ou **roles internas** podem operar.
* HTTPS obrigatório.

**13) Métricas & Logs (MVP)**

* Taxa de sucesso; número de **409 por idempotência**; tempo médio de crédito.
* Logs estruturados: ContaId, MovimentoId, IdOperacao, Valor, SaldoAnterior→SaldoPosterior, canal.

**14) Pontos em Aberto / TODOs**

* **[TODO-DEP-01]** Política antifraude/limites de **depósito** (valor máximo por operação/dia, hold, origem).
* **[TODO-DEP-02]** Evoluir para **contabilidade de dupla entrada** (ex.: “Caixa” ↔ “Conta do Cliente”).
* **[TODO-DEP-03]** Suportar outras moedas/câmbio em versões futuras.