**Caso de Uso — Consulta de Saldo**

**1) Identificação**

* **ID:** UC-REL-001
* **Nome:** Consulta de Saldo da Conta
* **Objetivo:** Permitir ao cliente visualizar o **saldo atual** de sua conta de forma segura e consistente.

**2) Atores**

* **Primário:** Cliente autenticado (app/web) — apenas a **própria conta**.
* **Secundário (opcional):** Operador/Administrador (canal interno) para suporte/atendimento.

**3) Pré-condições**

* Sistema, banco de dados e **API de Autenticação** disponíveis.
* Cliente **autenticado** com **token válido**.
* Conta existente no sistema.

**4) Disparador**

* Cliente solicita a visualização do saldo (via app/web).

**5) Regras de Negócio (MVP)**

* **Autorização/Ownership:** o titular só pode consultar **a própria conta** (match clienteId do token com ClienteId da conta).
* **Status da conta:** consulta é **permitida** para contas Ativa, Inativa, Bloqueada e Encerrada (modo leitura).
* **Moeda:** "BRL" (MVP).
* **Consistência:** saldo retornado deve refletir o estado **confirmado** (read committed) após a última transação.
* **Sem idempotência:** operação apenas de leitura (não usa IdOperacao).
* **(Futuro) Reservas/Bloqueios:** quando modelados, exibir também **saldoDisponivel = saldo − reservas**. **[TODO-REL-01]**

**6) Fluxo Principal (Feliz)**

1. Cliente chama o endpoint de saldo com **Bearer Token** válido.
2. Sistema **valida o token** via API de Autenticação.
3. Sistema determina o **contaId** (via rota/param e/ou associação do cliente).
4. Sistema carrega a conta e valida **ownership**.
5. Sistema lê o **saldo atual** (e moeda) do banco de contas.
6. Sistema retorna **200 OK** com o saldo atual (e timestamp de atualização).

**7) Fluxos Alternativos / Exceções**

* **A1 — Token inválido/expirado:**  
  2a. **401 Unauthorized** NAO\_AUTENTICADO.
* **A2 — Conta inexistente:**  
  4a. **404 Not Found** CONTA\_NAO\_ENCONTRADA.
* **A3 — Sem permissão (falta de ownership/role):**  
  4b. **403 Forbidden** PERMISSAO\_NEGADA.
* **A4 — Falha de leitura/persistência:**  
  5a. **500 Internal Server Error** ERRO\_LEITURA.

**8) Pós-condições**

* Nenhuma alteração em dados.
* Registro de auditoria/log de consulta (mínimo necessário).

**9) Dados Lidos / Persistência**

* **Conta:**
  + Id, ClienteId, Saldo, Moeda, Status
  + (opcional) AtualizadoEm / RowVersion para compor metadata do retorno

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

* **Endpoint:** GET /api/contas/{id}/saldo
* **Headers obrigatórios:** Authorization: Bearer <token>
* **Response 200 (exemplo):**

{

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

"moeda": "BRL",

"saldo": 1234.56,

"statusConta": "Ativa",

"atualizadoEm": "2025-08-15T02:30:00Z"

}

* **Erros comuns:** 401, 403, 404, 500.
* **Cabeçalhos recomendados (segurança/privacidade):**
  + Cache-Control: no-store (evitar cache de dado sensível)
  + Pragma: no-cache

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

1. **Consulta válida**
   * Given token válido e conta do próprio cliente
   * When GET /api/contas/{id}/saldo
   * Then **200** com saldo e moeda.
2. **Não autenticado**
   * Given token inválido/ausente
   * Then **401 NAO\_AUTENTICADO**.
3. **Sem permissão**
   * Given conta não pertence ao cliente do token
   * Then **403 PERMISSAO\_NEGADA**.
4. **Conta inexistente**
   * Given id não encontrado
   * Then **404 CONTA\_NAO\_ENCONTRADA**.
5. **Conta encerrada (leitura)**
   * Given status Encerrada
   * Then **200** com saldo final e statusConta = Encerrada.

**12) Observações de Segurança e Privacidade**

* **Nunca** armazenar/registrar saldo em caches compartilhados do lado servidor/edge sem controle.
* Logs **mínimos**: ContaId mascarado, ClienteId mascarado, timestamp e resultado.
* HTTPS obrigatório.
* Rate limit por usuário/conta (prevenir scraping). **[TODO-REL-02]**

**13) Métricas & Logs (MVP)**

* Latência média, taxa de sucesso/erro por motivo (401/403/404/500).
* Contagem de consultas por conta/usuário (com limites).

**14) TODOs / Futuro**

* **[TODO-REL-01]** Expor saldoDisponivel quando houver **reservas/bloqueios**.
* **[TODO-REL-02]** Implementar rate limiting por cliente/conta.
* **[TODO-REL-03]** Suportar **ETag/If-None-Match** para respostas condicionais (evitar tráfego desnecessário sem expor cache sensível).
* **[TODO-REL-04]** Endpoint agregado GET /api/contas/me/saldos (multi-contas, quando houver PJ/múltiplas contas).