**Caso de Uso — Recuperação/Redefinição de Senha**

**1) Identificação**

* **ID:** UC-SEG-002
* **Nome:** Recuperação/Redefinição de Senha
* **Objetivo:** Permitir que o cliente redefina a senha com segurança, sem expor se o login existe e revogando sessões antigas.

**2) Atores**

* **Primário:** Cliente (app/web).
* **Sistemas envolvidos:** API de Autenticação (interna), serviço de e-mail.

**3) Pré-condições**

* API de Autenticação e serviço de e-mail disponíveis.
* Cliente previamente **cadastrado** (UC-CLI-001).
* Status do cliente:
  + **Ativo** → pode redefinir.
  + **Bloqueado por tentativas** → pode redefinir (para desbloquear).
  + **Inativo/Encerrado** → redefinição negada (tratado sem revelar existência).

**4) Disparadores**

* **Etapa 1 (pedido):** cliente envia **login** (email ou CPF).
* **Etapa 2 (confirmação):** cliente envia **token** recebido + **nova senha**.

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

* **Campos obrigatórios:** login na etapa 1; token e novaSenha na etapa 2.
* **Login:** aceitar **email** (formato válido) ou **CPF** (DV válido, normalizado).
* **Anti-enumeração:** resposta **genérica de sucesso** no pedido, **independente** de o login existir.
* **Rate limit:** limitar tentativas por **login** e por **IP** (ex.: 5/min). Estourou → **429**.
* **Token de reset:**
  + **One-time**, **expira em 15 minutos**, associado ao usuário, com jti.
  + Armazenar **hash** do token; marcar **usado/revogado** após sucesso.
  + Invalida também tokens anteriores pendentes (apenas o mais novo fica válido).
* **Política de senha (MVP):** mínimo **8** chars, recomendar ≥ 12; impedir senha igual às 3 últimas (opcional).
* **Sessões:** ao redefinir, **revogar todas as sessões/refresh tokens** do usuário.
* **Auditoria:** logar pedido (mínimo) e êxito/falhas de redefinição (com motivo), IP e user agent.
* **Entrega de e-mail:** falha de envio **não altera** a resposta genérica (mas gera alerta/log).
* **Auto-login após reset:** **não** habilitado no MVP (exigir login). *(pode ativar no futuro).*

**6) Fluxo Principal (Feliz)**

**Etapa 1 — Pedido de reset**

1. Cliente envia POST /api/auth/password/reset/request com login.
2. Sistema valida campos e formato.
3. Sistema aplica rate limit (login/IP).
4. Se usuário existir e elegível, cria **ticket** e **token** (15 min, one-time) e envia **e-mail** com link.
5. Sistema retorna **200 OK** com mensagem genérica.

**Etapa 2 — Confirmação do reset**  
6. Cliente acessa o link (ou envia token) e informa novaSenha.  
7. Sistema valida o **token** (existe, pertence ao usuário, não expirado, não usado).  
8. Sistema valida a **política de senha**.  
9. Sistema aplica o **hash** (Argon2id/bcrypt), atualiza a senha e **revoga** o token.  
10. Sistema **revoga todas as sessões/refresh** do usuário.  
11. Sistema registra auditoria e retorna **200 OK** (orientando a fazer login).

**7) Fluxos Alternativos / Exceções**

* **A1 — Campos obrigatórios ausentes (etapa 1):**  
  **400 Bad Request** CAMPOS\_OBRIGATORIOS.
* **A2 — Login inválido (formato):**  
  **422 Unprocessable Entity** LOGIN\_INVALIDO.
* **A3 — Rate limit excedido:**  
  **429 Too Many Requests** MUITAS\_TENTATIVAS.
* **A4 — Token inválido/expirado (etapa 2):**  
  **400/410** TOKEN\_INVALIDO\_OU\_EXPIRADO.
* **A5 — Senha fraca:**  
  **422** SENHA\_FRACA.
* **A6 — Usuário inelegível (inativo/encerrado):**  
  Pedido retorna **200** genérico; confirmação com token (se chegar a existir) → **403** USUARIO\_INELEGIVEL.
* **A7 — Erro interno (persistência/envio):**  
  **500** ERRO\_RECUPERACAO\_SENHA.

**8) Pós-condições**

* Se confirmado: senha atualizada com hash seguro; **todas as sessões/refresh revogados**.
* Tickets e tokens de reset com status atualizado (usado/expirado).
* Auditoria registrada.

**9) Dados Persistidos (rascunho)**

* **PasswordResetTicket:**
  + Id (GUID), UsuarioId, TokenHash, Jti, ExpiraEm, UsadoEm?, Revogado (bool), CriadoEm, IpCriacao, UserAgent.
* **Usuario/Cliente:**
  + SenhaHash, SenhaAlg, SenhaAtualizadaEm, *(opcional)* HistoricoSenhas[].
* **Sessao:** revogadas (marcar Revogado=true, RevogadoEm).
* **AuditoriaResetSenha:** Id, UsuarioId?, Evento (Pedido/Sucesso/Falha), MotivoFalha?, Ip, UserAgent, CorrelationId, CriadoEm.

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

**Pedido de Reset**

* **Endpoint:** POST /api/auth/password/reset/request
* **Headers (recomendado):** X-Correlation-Id
* **Request:**

{

"login": "ana.souza@example.com"

}

Também aceitar: "login": "12345678909" (CPF normalizado)

* **Response 200 (genérica):**

{

"message": "Se o login existir, enviaremos instruções para redefinir a senha."

}

**Confirmação do Reset**

* **Endpoint:** POST /api/auth/password/reset/confirm
* **Request:**

{

"token": "<token-de-reset>",

"novaSenha": "UmaSenhaBemF0rte!"

}

* **Response 200:**

{

"message": "Senha redefinida com sucesso. Faça login novamente."

}

* **Erros comuns:** 400, 410, 422, 429, 500 (mensagens claras e seguras).

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

1. **Pedido de reset válido**
   * Given login em formato válido
   * When POST /api/auth/password/reset/request
   * Then **200** com mensagem genérica (sem revelar existência).
2. **Confirmação com token válido**
   * Given token válido e nova senha forte
   * When POST /api/auth/password/reset/confirm
   * Then **200** e todas as sessões anteriores revogadas.
3. **Token inválido/expirado**
   * Given token inválido
   * Then **400/410 TOKEN\_INVALIDO\_OU\_EXPIRADO**.
4. **Senha fraca**
   * Given nova senha fora da política
   * Then **422 SENHA\_FRACA**.
5. **Rate limit**
   * Given muitas tentativas de pedido
   * Then **429 MUITAS\_TENTATIVAS**.

**12) Observações de Segurança e Privacidade**

* **Hash de senha:** Argon2id (recomendado) ou bcrypt (cost ≥ 12); sal único; comparação tempo constante.
* **Token de reset:** assinado/aleatório forte; armazenar **hash**; escopo exclusivo de redefinição; **one-time**.
* **Links:** expiração curta (15 min) e **single use**; invalidar outros tokens pendentes no novo pedido.
* **E-mail:** conteúdo sem dados sensíveis; incluir CorrelationId nos logs de envio.
* **TLS obrigatório**; cabeçalhos anti-cache (Cache-Control: no-store).
* **Erros genéricos** para evitar enumeração.
* **Revogar sessões** sempre após redefinir.

**13) Métricas & Logs (MVP)**

* Taxa de pedidos, taxa de sucesso de redefinição, % de tokens expirados, tentativas por IP/login, tempo médio fim-a-fim.
* Alertas para spikes de pedidos por IP (possível abuso).

**14) TODOs / Futuro**

* **[TODO-SEG-RESET-01]** **MFA/2FA** na confirmação (ex.: TOTP/SMS além do link).
* **[TODO-SEG-RESET-02]** **Captcha** adaptativo após N tentativas.
* **[TODO-SEG-RESET-03]** UI para **revogar todas as sessões** manualmente.
* **[TODO-SEG-RESET-04]** **Blacklist** de senhas vazadas (Have I Been Pwned).
* **[TODO-SEG-RESET-05]** Opção de **auto-login seguro** pós-reset (mesmo dispositivo) — configurável.
* **[TODO-SEG-RESET-06]** Canal alternativo (SMS) quando e-mail indisponível.