Caso de Uso — Alteração de Cadastro de Cliente

**1) Identificação**

* ID: UC-CLI-002
* Nome: Alteração de Cadastro de Cliente
* Objetivo: Atualizar dados cadastrais de um cliente existente preservando integridade, unicidade e histórico de alterações.

**2) Atores**

* Primário: Cliente autenticado (via app/web).
* Sistemas externos (futuro): Serviço de verificação de identidade (casos específicos).

**3) Pré-condições**

* Sistema e banco de dados disponíveis.
* Usuário autenticado e autorizado com permissão AlterarCadastroCliente (ou o próprio cliente alterando seus dados).
* Cliente existente no sistema.
* Controle de concorrência otimista via If-Match/ETag.

**4) Disparador**

* Solicitação de alteração enviando ID do cliente e patch com os campos a atualizar.

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

* Campos alteráveis: Nome, Email, Telefone, Endereco, Status (conforme política).
* Campos imutáveis: CPF (não pode ser alterado via este caso de uso).
* Formato/consistência:
  + Nome: 3–100 caracteres.
  + Email: formato válido.
  + Telefone (se informado): DDI+DDD+Número (normalizado).
  + Endereco: objeto (Logradouro, Numero, Complemento?, Bairro, Cidade, Estado=UF, CEP).
* Unicidade: Email não pode estar vinculado a outro cliente.
* No-op: Se o patch não resultar em mudança efetiva, responder 200 OK com o recurso atual (ETag inalterado) — política definida.
* Auditoria: registrar *quem* alterou, *quais campos*, *antes/depois*, *timestamp*, *origem/canal*.
* Concorrência: rejeitar atualização se If-Match divergir do ETag atual → 412 Precondition Failed.
* Validação externa (futuro): Se algum dia permitir mudança de atributos críticos (ex.: identidade), integrar serviço externo. MVP: não aplicável.

**6) Fluxo Principal (Feliz)**

1. Ator envia PATCH /api/clientes/{id} com operações de patch.
2. Sistema busca o cliente pelo clienteId.
3. Sistema valida If-Match/ETag.
4. Sistema valida operações e formatos dos novos valores.
5. Sistema verifica unicidade de Email (se alterado).
6. Sistema aplica as alterações e persiste no banco.
7. Sistema registra auditoria.
8. Sistema retorna 200 OK com o recurso atualizado e novo ETag.

**7) Fluxos Alternativos / Exceções**

* A1 — Cliente não encontrado:  
  2a. 404 Not Found CLIENTE\_NAO\_ENCONTRADO.
* A2 — Dados/patch inválidos:  
  4a. 422 Unprocessable Entity CLIENTE\_DADOS\_INVALIDOS (detalhes por campo).  
  4b. 400 Bad Request PATCH\_OPERACAO\_INVALIDA (op inválida, path inválido, JSON Patch malformado).
* A3 — E-mail já utilizado por outro cliente:  
  5a. 409 Conflict EMAIL\_JA\_UTILIZADO.
* A4 — Tentativa de alteração de campo imutável (CPF):  
  4c. 422 Unprocessable Entity CAMPO\_IMUTAVEL (path /cpf não permitido).
* A5 — Falha de persistência:  
  6a. 500 Internal Server Error ERRO\_PERSISTENCIA.
* A6 — Concorrência violada:  
  3a. 412 Precondition Failed VERSAO\_DESATUALIZADA.
* A7 — Permissão insuficiente:  
  0a. 403 Forbidden PERMISSAO\_NEGADA.
* A8 — No-op (nada mudou):  
  6b. 200 OK sem alterações (mesmo ETag).

**8) Pós-condições**

* Dados do cliente atualizados conforme solicitação.
* Auditoria de alteração registrada.
* Unicidade de Email preservada.
* ETag do recurso atualizado (se houve mudanças).

**9) Dados Persistidos (diferenças relevantes)**

* Cliente: Nome, Email, Telefone, Endereco, Status (conforme política).
* AtualizadoEm (datetime), AtualizadoPor (string/userId).
* RowVersion (binário/base64) no banco (se usado) — pode ser base para cálculo do ETag.

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

* Endpoint: PATCH /api/clientes/{id}
* Headers:
  + If-Match: "<ETag atual do recurso>" *(obrigatório para atualizar)*
  + Content-Type: application/json-patch+json
* Corpo (JSON Patch — exemplo: alterar email, telefone e número do endereço):

[

{ "op": "replace", "path": "/email", "value": "ana.souza+novo@example.com" },

{ "op": "replace", "path": "/telefone", "value": "+5588988887777" },

{ "op": "replace", "path": "/endereco/numero", "value": "120" }

]

* Response 200 (exemplo):

{

"id": "b3c3a6b1-7a2c-4c49-8b6f-0d3c4a9d2e10",

"nome": "Ana M. Souza",

"cpf": "12345678909",

"email": "ana.souza+novo@example.com",

"telefone": "+5588988887777",

"endereco": {

"logradouro": "Rua Y",

"numero": "120",

"complemento": "Casa",

"bairro": "Centro",

"cidade": "Juazeiro do Norte",

"estado": "CE",

"cep": "63000000"

},

"status": "Ativo",

"atualizadoEm": "2025-08-14T21:30:00Z",

"eTag": "\"4-9d2e10\""

}

* Erros comuns: 400, 403, 404, 409, 412, 422, 500.

Paths permitidos no JSON Patch (MVP):  
/nome, /email, /telefone, /status,  
/endereco/logradouro, /endereco/numero, /endereco/complemento, /endereco/bairro, /endereco/cidade, /endereco/estado, /endereco/cep.  
Proibido: /cpf.

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

1. Alteração válida (PATCH)
   * Given cliente existe, ETag válido e patch correto
   * When envio PATCH /api/clientes/{id}
   * Then recebo 200 com dados atualizados e novo ETag.
2. Cliente inexistente
   * Given id inexistente
   * When envio a requisição
   * Then recebo 404 CLIENTE\_NAO\_ENCONTRADO.
3. E-mail duplicado
   * Given e-mail pertence a outro cliente
   * When envio o patch
   * Then recebo 409 EMAIL\_JA\_UTILIZADO.
4. Campo imutável (CPF)
   * Given patch tenta alterar /cpf
   * When envio a alteração
   * Then recebo 422 CAMPO\_IMUTAVEL.
5. Concorrência
   * Given If-Match desatualizado
   * When envio o patch
   * Then recebo 412 VERSAO\_DESATUALIZADA.
6. Patch inválido
   * Given JSON Patch malformado ou path não permitido
   * When envio a alteração
   * Then recebo 400 PATCH\_OPERACAO\_INVALIDA.
7. No-op
   * Given patch que não muda o estado
   * When envio a alteração
   * Then recebo 200 OK com o recurso inalterado (mesmo ETag).

**12) Observações de Segurança e Privacidade**

* Não expor dados sensíveis em logs; mascarar Email e Telefone quando necessário.
* HTTPS obrigatório.
* Autorização baseada em ownership (cliente só altera o próprio cadastro).
* Validar e normalizar campos antes de aplicar o patch (ex.: email/cep/ddd).

**13) Métricas & Logs (MVP)**

* Taxa de sucesso, taxa de 412 (concorrência), 409 (unicidade), tempo médio.
* Logs estruturados: campos alterados, ClienteId, UserId, ETag antigo/novo, origem/canal.