Caso de Uso — Cadastro de Cliente

**1) Identificação**

* ID: UC-CLI-001
* Nome: Cadastro de Cliente
* Objetivo: Registrar um novo cliente no sistema, gerando um identificador único para uso nos demais módulos (conta, movimentações, etc.).

**2) Atores**

* Primário: Usuário Interno (operador/administrador) ou Cliente (via app/web, quando existir).
* Sistemas externos (futuros): Serviço de verificação de CPF/identidade.

**3) Pré-condições**

* Sistema e banco de dados disponíveis.
* Usuário autenticado (quando a rota exigir) e com permissão de CadastrarCliente.
* Regras de unicidade definidas (CPF e e-mail).
* Idempotência habilitada via IdOperacao (header) para evitar duplicidades em reenvio.

**4) Disparador**

* Ação do ator para cadastrar um novo cliente enviando os dados obrigatórios.

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

* Obrigatórios: Nome, CPF, Email.
* Opcionais (MVP): Telefone, Endereco.
* Formato/consistência:
  + Nome: 3–100 caracteres.
  + CPF: formato e dígitos verificadores válidos (validação local no MVP).
  + Email: formato válido.
  + Telefone (se informado): DDI+DDD+Número (limpar máscara).
* Unicidade: CPF e e-mail não podem existir previamente.
* Idempotência: Se IdOperacao já processado para payload equivalente, retornar o mesmo resultado (200/201) sem novo insert.
* Auditoria: registrar quem cadastrou, timestamp e origem (canal).
* Validação externa de CPF (futuro): Integração com serviço oficial ou de terceiros para validar a autenticidade do CPF informado.
  + MVP: Apenas validação local (formato + dígito verificador).
  + Versão futura: Chamada à API externa após validação local e antes de persistir o cliente.

**6) Fluxo Principal (Feliz)**

1. Ator envia requisição de cadastro com os dados do cliente.
2. Sistema verifica presença de campos obrigatórios.
3. Sistema valida formatos e consistência (nome, CPF, email, telefone).
4. Sistema verifica existência prévia por CPF e e-mail.
5. Sistema persiste o cliente no banco.
6. Sistema registra auditoria e idempotência (quando aplicável).
7. Sistema retorna ID do cliente (GUID) e dados básicos confirmados.

**7) Fluxos Alternativos / Exceções**

* A1 — Dados obrigatórios ausentes:  
  2a. Retornar 400 com code: CLIENTE\_DADOS\_INCOMPLETOS.
* A2 — Formato inválido:  
  3a. Retornar 422 com code: CLIENTE\_DADOS\_INVALIDOS.
* A3 — Cliente já existe:  
  4a. Retornar 409 com code: CLIENTE\_JA\_CADASTRADO (não retornar ClienteId).
* A4 — Falha de persistência:  
  5a. Retornar 500 com code: ERRO\_PERSISTENCIA.
* A5 — Idempotência:  
  6a. Retornar 200/201 com mesma resposta anterior.
* A6 — Permissão insuficiente:  
  0a. Retornar 403 com code: PERMISSAO\_NEGADA.

**8) Pós-condições**

* Cliente gravado com ID único e status Ativo.
* Auditoria registrada.
* Índice único em CPF e e-mail.
* Sistema pronto para operações subsequentes.

**9) Dados Persistidos (rascunho)**

* Cliente:
  + Id (GUID)
  + Nome (string)
  + CPF (string, único, normalizado)
  + Email (string, único, normalizado)
  + Telefone (string, opcional, normalizado)
  + Endereco (objeto)
    - Logradouro (string)
    - Numero (string)
    - Complemento (string, opcional)
    - Bairro (string)
    - Cidade (string)
    - Estado (string — sigla UF)
    - CEP (string, normalizado)
  + Status (enum: Ativo/Inativo)
  + CriadoEm (datetime)
  + CriadoPor (string/userId)

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

* Endpoint: POST /api/clientes
* Headers: IdOperacao (opcional, recomendado)
* Request (exemplo):

{

"nome": "Ana Souza",

"cpf": "12345678909",

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

"telefone": "+5588999999999",

"endereco": {

"logradouro": "Rua X",

"numero": "100",

"complemento": "Apto 201",

"bairro": "Centro",

"cidade": "Juazeiro do Norte",

"estado": "CE",

"cep": "63000000"

}

}

* Response 201 (exemplo):

{

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

"nome": "Ana Souza",

"cpf": "12345678909",

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

"status": "Ativo",

"criadoEm": "2025-08-14T21:05:00Z"

}

* Erros comuns: 400, 409, 422, 500.

**11) Critérios de Aceitação**

1. Cadastro válido:
   * Given dados corretos, When envio requisição, Then recebo 201 com ID.
2. Dados incompletos:
   * Given falta campo obrigatório, Then recebo 400.
3. CPF inválido:
   * Given CPF inválido, Then recebo 422.
4. Cliente duplicado:
   * Given CPF cadastrado, Then recebo 409 sem ClienteId.
5. Idempotência:
   * Given mesmo IdOperacao, Then recebo resposta anterior.

**12) Observações de Segurança e Privacidade**

* Não logar CPF/Email em texto puro.
* HTTPS obrigatório.
* Campos retornados mínimos.

**13) Métricas & Logs**

* Taxa de sucesso, taxa de conflito, tempo médio de processamento.
* Logs estruturados com decisão de unicidade, ClienteId, IdOperacao.

**14) Pontos em Aberto**

* Modelagem final de endereço confirmada (classe com subcampos).
* Integração de validação externa de CPF na versão futura.