# **Modelagem de Dados - Cadastro de Cliente**

**QuantumFinance - MVP Arquitetura de Dados**

—-

## **1. MODELO CONCEITUAL**

### **Entidades Principais**

* **CLIENTE**: Pessoa física que possui conta na QuantumFinance
* **CONTATO**: Informações de contato do cliente (telefone, email, contato de emergência)
* **ENDERECO**: Endereço residencial e/ou comercial do cliente
* **DOCUMENTO**: Arquivos digitais (foto, assinatura eletrônica)

### **Relacionamentos**

* Um CLIENTE pode ter múltiplos CONTATOS (1:N)
* Um CLIENTE pode ter múltiplos ENDERECOS (1:N)
* Um CLIENTE pode ter múltiplos DOCUMENTOS (1:N)

### **Diagrama Conceitual**

[CLIENTE] ——< [CONTATO]  
 |  
 |——< [ENDERECO]  
 |  
 |——< [DOCUMENTO]

—-

## **2. MODELO LÓGICO**

### **Entidade: CLIENTE**

* **cliente\_id** (PK): Identificador único
* **cpf**: CPF do cliente (único)
* **nome\_completo**: Nome completo
* **data\_nascimento**: Data de nascimento
* **genero**: Gênero (M/F/O)
* **estado\_civil**: Estado civil
* **profissao**: Profissão
* **renda\_mensal**: Renda mensal declarada
* **data\_cadastro**: Data do cadastro
* **status\_cliente**: Ativo/Inativo/Bloqueado
* **created\_at**: Timestamp de criação
* **updated\_at**: Timestamp de atualização

### **Entidade: CONTATO**

* **contato\_id** (PK): Identificador único
* **cliente\_id** (FK): Referência ao cliente
* **tipo\_contato**: Tipo (telefone, email, emergencia)
* **valor\_contato**: Valor do contato
* **principal**: Indica se é o contato principal (S/N)
* **verificado**: Indica se foi verificado (S/N)
* **created\_at**: Timestamp de criação

### **Entidade: ENDERECO**

* **endereco\_id** (PK): Identificador único
* **cliente\_id** (FK): Referência ao cliente
* **tipo\_endereco**: Tipo (residencial, comercial)
* **logradouro**: Rua, avenida, etc.
* **numero**: Número
* **complemento**: Complemento
* **bairro**: Bairro
* **cidade**: Cidade
* **estado**: Estado (UF)
* **cep**: CEP
* **pais**: País
* **principal**: Indica se é o endereço principal (S/N)
* **created\_at**: Timestamp de criação

### **Entidade: DOCUMENTO**

* **documento\_id** (PK): Identificador único
* **cliente\_id** (FK): Referência ao cliente
* **tipo\_documento**: Tipo (foto, assinatura, rg, cnh)
* **nome\_arquivo**: Nome do arquivo
* **caminho\_arquivo**: Caminho no Data Lake
* **tamanho\_arquivo**: Tamanho em bytes
* **formato\_arquivo**: Formato (jpg, png, pdf)
* **hash\_arquivo**: Hash para integridade
* **data\_upload**: Data do upload
* **status\_documento**: Status (pendente, aprovado, rejeitado)
* **created\_at**: Timestamp de criação

—-

## **3. MODELO FÍSICO**

### **Tabela: TB\_CLIENTE**

CREATE TABLE TB\_CLIENTE (  
 cliente\_id BIGINT IDENTITY(1,1) PRIMARY KEY,  
 cpf VARCHAR(11) NOT NULL UNIQUE,  
 nome\_completo VARCHAR(255) NOT NULL,  
 data\_nascimento DATE NOT NULL,  
 genero CHAR(1) CHECK (genero IN (‘M‘, ‘F‘, ‘O‘)),  
 estado\_civil VARCHAR(20),  
 profissao VARCHAR(100),  
 renda\_mensal DECIMAL(15,2),  
 data\_cadastro DATE NOT NULL DEFAULT GETDATE(),  
 status\_cliente VARCHAR(20) NOT NULL DEFAULT ‘ATIVO‘   
 CHECK (status\_cliente IN (‘ATIVO‘, ‘INATIVO‘, ‘BLOQUEADO‘)),  
 created\_at DATETIME2 NOT NULL DEFAULT GETDATE(),  
 updated\_at DATETIME2 NOT NULL DEFAULT GETDATE()  
);  
  
— Índices  
CREATE INDEX IX\_TB\_CLIENTE\_CPF ON TB\_CLIENTE(cpf);  
CREATE INDEX IX\_TB\_CLIENTE\_STATUS ON TB\_CLIENTE(status\_cliente);  
CREATE INDEX IX\_TB\_CLIENTE\_DATA\_CADASTRO ON TB\_CLIENTE(data\_cadastro);

### **Tabela: TB\_CONTATO**

CREATE TABLE TB\_CONTATO (  
 contato\_id BIGINT IDENTITY(1,1) PRIMARY KEY,  
 cliente\_id BIGINT NOT NULL,  
 tipo\_contato VARCHAR(20) NOT NULL   
 CHECK (tipo\_contato IN (‘TELEFONE‘, ‘EMAIL‘, ‘EMERGENCIA‘)),  
 valor\_contato VARCHAR(255) NOT NULL,  
 principal CHAR(1) NOT NULL DEFAULT ‘N‘ CHECK (principal IN (‘S‘, ‘N‘)),  
 verificado CHAR(1) NOT NULL DEFAULT ‘N‘ CHECK (verificado IN (‘S‘, ‘N‘)),  
 created\_at DATETIME2 NOT NULL DEFAULT GETDATE(),  
   
 CONSTRAINT FK\_TB\_CONTATO\_CLIENTE   
 FOREIGN KEY (cliente\_id) REFERENCES TB\_CLIENTE(cliente\_id)  
);  
  
— Índices  
CREATE INDEX IX\_TB\_CONTATO\_CLIENTE ON TB\_CONTATO(cliente\_id);  
CREATE INDEX IX\_TB\_CONTATO\_TIPO ON TB\_CONTATO(tipo\_contato);

### **Tabela: TB\_ENDERECO**

CREATE TABLE TB\_ENDERECO (  
 endereco\_id BIGINT IDENTITY(1,1) PRIMARY KEY,  
 cliente\_id BIGINT NOT NULL,  
 tipo\_endereco VARCHAR(20) NOT NULL   
 CHECK (tipo\_endereco IN (‘RESIDENCIAL‘, ‘COMERCIAL‘)),  
 logradouro VARCHAR(255) NOT NULL,  
 numero VARCHAR(20),  
 complemento VARCHAR(100),  
 bairro VARCHAR(100) NOT NULL,  
 cidade VARCHAR(100) NOT NULL,  
 estado CHAR(2) NOT NULL,  
 cep VARCHAR(8) NOT NULL,  
 pais VARCHAR(50) NOT NULL DEFAULT ‘BRASIL‘,  
 principal CHAR(1) NOT NULL DEFAULT ‘N‘ CHECK (principal IN (‘S‘, ‘N‘)),  
 created\_at DATETIME2 NOT NULL DEFAULT GETDATE(),  
   
 CONSTRAINT FK\_TB\_ENDERECO\_CLIENTE   
 FOREIGN KEY (cliente\_id) REFERENCES TB\_CLIENTE(cliente\_id)  
);  
  
— Índices  
CREATE INDEX IX\_TB\_ENDERECO\_CLIENTE ON TB\_ENDERECO(cliente\_id);  
CREATE INDEX IX\_TB\_ENDERECO\_CEP ON TB\_ENDERECO(cep);

### **Tabela: TB\_DOCUMENTO**

CREATE TABLE TB\_DOCUMENTO (  
 documento\_id BIGINT IDENTITY(1,1) PRIMARY KEY,  
 cliente\_id BIGINT NOT NULL,  
 tipo\_documento VARCHAR(20) NOT NULL   
 CHECK (tipo\_documento IN (‘FOTO‘, ‘ASSINATURA‘, ‘RG‘, ‘CNH‘)),  
 nome\_arquivo VARCHAR(255) NOT NULL,  
 caminho\_arquivo VARCHAR(500) NOT NULL,  
 tamanho\_arquivo BIGINT NOT NULL,  
 formato\_arquivo VARCHAR(10) NOT NULL,  
 hash\_arquivo VARCHAR(64) NOT NULL,  
 data\_upload DATETIME2 NOT NULL DEFAULT GETDATE(),  
 status\_documento VARCHAR(20) NOT NULL DEFAULT ‘PENDENTE‘  
 CHECK (status\_documento IN (‘PENDENTE‘, ‘APROVADO‘, ‘REJEITADO‘)),  
 created\_at DATETIME2 NOT NULL DEFAULT GETDATE(),  
   
 CONSTRAINT FK\_TB\_DOCUMENTO\_CLIENTE   
 FOREIGN KEY (cliente\_id) REFERENCES TB\_CLIENTE(cliente\_id)  
);  
  
— Índices  
CREATE INDEX IX\_TB\_DOCUMENTO\_CLIENTE ON TB\_DOCUMENTO(cliente\_id);  
CREATE INDEX IX\_TB\_DOCUMENTO\_TIPO ON TB\_DOCUMENTO(tipo\_documento);  
CREATE INDEX IX\_TB\_DOCUMENTO\_STATUS ON TB\_DOCUMENTO(status\_documento);

—-

## **4. REGRAS DE NEGÓCIO IMPLEMENTADAS**

### **Validações de Integridade**

1. **CPF único**: Cada cliente deve ter um CPF único no sistema
2. **Contato principal**: Cada cliente deve ter pelo menos um contato principal
3. **Endereço principal**: Cada cliente deve ter pelo menos um endereço principal
4. **Documentos obrigatórios**: Todo cliente deve ter foto e assinatura

### **Constraints e Triggers**

— Trigger para garantir apenas um contato principal por tipo  
CREATE TRIGGER TR\_CONTATO\_PRINCIPAL  
ON TB\_CONTATO  
AFTER INSERT, UPDATE  
AS  
BEGIN  
 — Lógica para garantir apenas um contato principal por tipo por cliente  
 IF EXISTS (  
 SELECT cliente\_id, tipo\_contato  
 FROM TB\_CONTATO  
 WHERE principal = ‘S‘  
 GROUP BY cliente\_id, tipo\_contato  
 HAVING COUNT(\*) > 1  
 )  
 BEGIN  
 RAISERROR(‘Apenas um contato principal por tipo é permitido‘, 16, 1);  
 ROLLBACK TRANSACTION;  
 END  
END;

—-

## **5. ESTRATÉGIA DE PARTICIONAMENTO**

### **Particionamento por Data**

* **TB\_CLIENTE**: Particionada por `data\_cadastro` (mensal)
* **TB\_DOCUMENTO**: Particionada por `data\_upload` (mensal)

### **Arquivamento**

* Dados de clientes inativos > 7 anos: movidos para tabelas de arquivo
* Documentos > 5 anos: compactados e movidos para storage de baixo custo

—-

## **6. SEGURANÇA E COMPLIANCE**

### **Criptografia**

* **CPF**: Criptografado em repouso usando AES-256
* **Dados pessoais**: Mascaramento para usuários não autorizados

### **Auditoria**

CREATE TABLE TB\_AUDITORIA\_CLIENTE (  
 auditoria\_id BIGINT IDENTITY(1,1) PRIMARY KEY,  
 cliente\_id BIGINT NOT NULL,  
 operacao VARCHAR(10) NOT NULL, — INSERT, UPDATE, DELETE  
 usuario VARCHAR(100) NOT NULL,  
 data\_operacao DATETIME2 NOT NULL DEFAULT GETDATE(),  
 dados\_anteriores NVARCHAR(MAX),  
 dados\_novos NVARCHAR(MAX)  
);

### **LGPD Compliance**

* Campo `consentimento\_lgpd` na tabela TB\_CLIENTE
* Procedures para anonização e exclusão de dados
* Log de acesso a dados pessoais

—-

## **7. MÉTRICAS E MONITORAMENTO**

### **KPIs de Qualidade de Dados**

* Taxa de completude dos campos obrigatórios
* Taxa de validação de documentos
* Tempo médio de processamento do cadastro

### **Views para Relatórios**

CREATE VIEW VW\_CLIENTE\_COMPLETO AS  
SELECT   
 c.cliente\_id,  
 c.nome\_completo,  
 c.cpf,  
 c.status\_cliente,  
 COUNT(DISTINCT ct.contato\_id) as total\_contatos,  
 COUNT(DISTINCT e.endereco\_id) as total\_enderecos,  
 COUNT(DISTINCT d.documento\_id) as total\_documentos  
FROM TB\_CLIENTE c  
LEFT JOIN TB\_CONTATO ct ON c.cliente\_id = ct.cliente\_id  
LEFT JOIN TB\_ENDERECO e ON c.cliente\_id = e.cliente\_id  
LEFT JOIN TB\_DOCUMENTO d ON c.cliente\_id = d.cliente\_id  
GROUP BY c.cliente\_id, c.nome\_completo, c.cpf, c.status\_cliente;

—-

## **8. INTEGRAÇÃO COM O PIPELINE**

### **Mapeamento de Campos**

|  |  |  |
| --- | --- | --- |
| \*\*Campo API\*\* | \*\*Tabela Destino\*\* | \*\*Campo Destino\*\* |
| ———————- | —————————- | —————————- |
| `nome` | TB\_CLIENTE | `nome\_completo` |
| `cpf` | TB\_CLIENTE | `cpf` |
| `email` | TB\_CONTATO | `valor\_contato` |
| `telefone` | TB\_CONTATO | `valor\_contato` |
| `endereco.rua` | TB\_ENDERECO | `logradouro` |
| `foto\_url` | TB\_DOCUMENTO | `caminho\_arquivo` |

### **Sequência de Carga**

1. Inserir registro em TB\_CLIENTE
2. Inserir contatos em TB\_CONTATO
3. Inserir endereços em TB\_ENDERECO
4. Inserir referências de documentos em TB\_DOCUMENTO

—-

**Documento preparado por:** SmartMind Analytics

**Versão:** 1.0

**Data:** Junho 2025