



Estrutura de Dados – API de Controle de Finanças Pessoais

Tabelas

 Tabela: clientes

```
CREATE TABLE cliente (  
    id VARCHAR(100) PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    senha VARCHAR(100) NOT NULL,  
    ativo BOOLEAN NOT NULL DEFAULT true,  
    criado_em TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    foto VARCHAR(100) NOT NULL  
);
```

 Tabela: transacoes

```
CREATE TABLE transacao (  
    id VARCHAR(100) PRIMARY KEY,  
    cliente_id VARCHAR(100) REFERENCES cliente(id),  
    tipo VARCHAR(50) NOT NULL,  
    descricao VARCHAR(100) NOT NULL,  
    valor DECIMAL(10, 2) NOT NULL,  
    categoria VARCHAR(100) NOT NULL,  
    criado_em TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Autenticação com JWT

Fluxo:

1. Registro (POST /cliente/registrar)

- Envia: `nome`, `email`, `senha`, `foto`
- A senha é criptografada com bcrypt

2. Login (POST /cliente/login)

- Envia: `email`, `senha`
- Se estiver correto, gera um JWT token

3. Proteção de rotas com middleware

- Rota de transações exige token JWT
- Middleware extrai o `cliente_id` e filtra dados do cliente

Exemplo de payload JWT

```
{  
  "cliente_id": "a6d9-54a2-489c...",  
  "email": "gabriel@email.com"  
}
```

Estrutura de Pastas Adicional

```
/middlewares  
  authMiddleware.js  
/routes  
  clientes.js
```