

Estrutura de Dados – API de Controle de Finanças Pessoais

Tabelas

Tabela: clientes

```
CREATE TABLE clientes (  
  id UUID PRIMARY KEY,  
  nome TEXT NOT NULL,  
  email TEXT UNIQUE NOT NULL,  
  senha TEXT NOT NULL,  
  criado_em TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Tabela: transacoes

```
CREATE TABLE transacoes (  
  id UUID PRIMARY KEY,  
  cliente_id UUID REFERENCES clientes(id) ON DELETE CASCADE,  
  tipo VARCHAR(10) NOT NULL, -- 'receita' ou 'despesa'  
  descricao TEXT NOT NULL,  
  valor NUMERIC(10,2) NOT NULL,  
  categoria TEXT,  
  data DATE NOT NULL,  
  criado_em TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Autenticação com JWT

Fluxo:

1. Registro (POST /clientes/registrar)

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

2. Login (POST /clientes/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
```