Ícone

O conteúdo gerado por IA pode estar incorreto.

Aplicativo gestão de casa

# Ideias iniciais:

Nome: GHouse.

Função: gerenciar contas e finanças de casa.

Funcionalidades:

* Mercado;
* Contas;
* Relatórios;

# Requisitos Técnicos:

## Mercado:

* Cadastrar produtos da lista
* Excluir produtos da lista
* Consultar produtos da lista
* Atualizar informações dos produtos da lista
* Marcar produto como comprado
* Consultar histórico de compras
* Mostrar quem comprou
* Mostrar onde comprou

## Contas:

* Cadastrar uma conta
* Excluir uma conta
* Atualizar informações sobre uma conta
* Listar todas as contas
* Marcar como paga e anexar comprovante
* Mostra quem pagou
* Status da conta (No prazo, atrasada e qnts dias atrasada)

## Relatórios:

* Relatórios do mercado de cada mês
* Relatórios de contas de cada mês

## No menu:

* Quanto foi gasto no mês
* Quantos produtos ainda não foram comprados
* Quantas contas ainda não foram pagas
* Quanto cada pessoa gastou no mês

## Outros requisitos:

* Criação de uma família
* Para entrar na família só acessando o link como convite
* Login
* Cadastro feito pelo usuário
* Autenticação JWT´
* Pra cadastrar vai receber um código pelo email

# Banco de dados:

Diagrama

O conteúdo gerado por IA pode estar incorreto.

-- Usuários

CREATE TABLE usuarios (

Id UUID NOT NULL PRIMARY KEY,

Nome VARCHAR(100) NOT NULL,

Email VARCHAR(100) NOT NULL UNIQUE,

Data\_cadastro DATE DEFAULT now() NOT NULL,

Usuarios\_permissoes VARCHAR(100) NOT NULL,

Password TEXT NOT NULL,

CONSTRAINT chk\_usuario\_permissoes CHECK (Usuarios\_permissoes IN ('Admin', 'Admin\_family’, 'User'))

);

-- Famílias

CREATE TABLE familias (

Id UUID NOT NULL PRIMARY KEY,

Nome VARCHAR(100) NOT NULL,

Data\_criacao DATE DEFAULT now() NOT NULL

);

-- Relação usuários ↔ famílias (N:N)

CREATE TABLE usuarios\_familia (

Id\_usuario UUID NOT NULL REFERENCES usuarios(Id),

Id\_familia UUID NOT NULL REFERENCES familias(Id),

PRIMARY KEY (Id\_usuario, Id\_familia)

);

-- Convites para famílias

CREATE TABLE convites\_familias (

Id UUID NOT NULL PRIMARY KEY,

Id\_familia UUID NOT NULL REFERENCES familias(Id),

Codigo\_convite TEXT NOT NULL UNIQUE, -- melhor que usar Nome da família

Data\_criacao DATE DEFAULT now() NOT NULL

);

-- Tokens de autenticação

CREATE TABLE token\_autenticacao (

Id UUID NOT NULL PRIMARY KEY,

Id\_usuario UUID NOT NULL REFERENCES usuarios(Id),

Token TEXT NOT NULL UNIQUE,

Data\_criacao TIMESTAMP NOT NULL DEFAULT NOW(),

Data\_expiracao TIMESTAMP NOT NULL

);

-- Produtos

CREATE TABLE produtos (

Id UUID NOT NULL PRIMARY KEY,

Data\_cadastro DATE DEFAULT now() NOT NULL,

Nome VARCHAR(100) NOT NULL,

Descricao TEXT NOT NULL,

Categoria VARCHAR(50) NOT NULL,

Preco DECIMAL(10,2) NOT NULL,

Id\_familia UUID NOT NULL REFERENCES familias(Id),

CONSTRAINT chk\_categoria CHECK (Categoria IN ('Cozinha', 'Banheiro', 'Mistura', ‘Extras’,’Sacolão’))

);

-- Contas

CREATE TABLE contas (

Id UUID NOT NULL PRIMARY KEY,

Nome VARCHAR(100) NOT NULL,

Descricao TEXT NOT NULL,

Foto TEXT,

Status VARCHAR(20) NOT NULL CHECK (Status IN ('Atrasada', 'No prazo', 'Paga')),

Data\_vencimento DATE NOT NULL,

Data\_cadastro DATE DEFAULT now() NOT NULL,

Numero\_codigo\_barra TEXT NOT NULL,

Id\_familia UUID NOT NULL REFERENCES familias(Id)

);

-- Pagamentos

CREATE TABLE pagamentos (

Id UUID NOT NULL PRIMARY KEY,

Id\_usuario UUID NOT NULL REFERENCES usuarios(Id),

Id\_conta UUID NOT NULL REFERENCES contas(Id),

Data\_pagamento DATE NOT NULL,

Foto\_comprovante TEXT,

Id\_familia UUID NOT NULL REFERENCES familias(Id),

Status\_conta VARCHAR(20) NOT NULL CHECK (Status\_conta IN ('Atrasada', 'No prazo', 'Paga'))

);

-- Compras

CREATE TABLE compras (

Id UUID NOT NULL PRIMARY KEY,

Id\_usuario UUID NOT NULL REFERENCES usuarios(Id), -- quem comprou

Id\_familia UUID NOT NULL REFERENCES familias(Id),

Data\_compra DATE NOT NULL,

Valor\_total DECIMAL(10,2) NOT NULL

);

-- Produtos dentro de uma compra

CREATE TABLE produtos\_compra (

Id UUID NOT NULL PRIMARY KEY,

Id\_compra UUID NOT NULL REFERENCES compras(Id),

Id\_produto UUID NOT NULL REFERENCES produtos(Id),

Quantidade INT NOT NULL CHECK (Quantidade > 0),

Preco\_unitario DECIMAL(10,2) NOT NULL,

Subtotal DECIMAL(10,2) GENERATED ALWAYS AS (Quantidade \* Preco\_unitario) STORED

);

select \* from usuarios

# Cores:

* Fundo: EDF2F4
* Sobreposições: 61A5C2
* Icons: ffffff

# Imagens docker:

#Rodar o PgAdmin4  
docker run --name pgadmin4 -p 15432:80 -e PGADMIN\_DEFAULT\_EMAIL=admin@admin.com -e PGADMIN\_DEFAULT\_PASSWORD=admin –-network GHouse-network dpage/pgadmin4

#Rodar e subir banco Posgres  
docker run --name GHousedb -e POSTGRES\_PASSWORD=AdminGHouse -e POSTGRES\_USER=GHouse\_admin -e POSTGRES\_DB=GHouse –-network GHouse-network -p 5432:5432 postgres:17.6

#Criar network

docker network create GHouse-network

# EndPoints API:

**🔹 1. Autenticação e registro**

| **Método** | **Endpoint** | **Acesso** | **Função** |
| --- | --- | --- | --- |
| POST | /usuarios/register | Público | Registro de novo usuário (Nome, Email, Password) → Usuarios\_permissoes = User |
| POST | /login | Público | Login do usuário, retorna token JWT |
| POST | /logout | Autenticado | Invalida token JWT (opcional) |
| GET | /token/refresh | Autenticado | Renova token JWT |

**🔹 2. Usuários (/usuarios)**

| **Método** | **Endpoint** | **Acesso** | **Função** |
| --- | --- | --- | --- |
| GET | /usuarios | Admin (sistema) | Listar todos usuários |
| GET | /usuarios/me | Autenticado | Ver dados do usuário logado |
| GET | /usuarios/{id} | Admin (sistema) | Buscar usuário por ID |
| POST | /usuarios | Admin (sistema) | Criar usuário |
| PUT | /usuarios/{id} | Admin (sistema) | Atualizar dados de qualquer usuário |
| DELETE | /usuarios/{id} | Admin (sistema) | Deletar usuário |

**🔹 3. Famílias (/familias)**

| **Método** | **Endpoint** | **Acesso** | **Função** |
| --- | --- | --- | --- |
| GET | /familias | Todos membros da família | Listar famílias do usuário |
| GET | /familias/{id} | Todos membros da família | Detalhes da família |
| POST | /familias | Admin\_family / Admin | Criar nova família |
| PUT | /familias/{id} | Admin\_family (criador) / Admin | Editar dados da família |
| DELETE | /familias/{id} | Admin (sistema) | Remover família |

**🔹 4. Relação Usuários ↔ Famílias (/usuarios\_familia)**

| **Método** | **Endpoint** | **Acesso** | **Função** |
| --- | --- | --- | --- |
| GET | /usuarios\_familia | Admin\_family / Admin | Listar usuários da família |
| POST | /usuarios\_familia | Admin\_family / Admin | Adicionar usuário à família |
| DELETE | /usuarios\_familia | Admin\_family / Admin | Remover usuário da família |
| PUT | /usuarios\_familia/permissao | Admin\_family / Admin | Definir Admin\_family ou User |

**🔹 5. Convites de famílias (/convites\_familias)**

| **Método** | **Endpoint** | **Acesso** | **Função** |
| --- | --- | --- | --- |
| GET | /convites\_familias | Admin\_family / Admin | Listar convites de uma família |
| POST | /convites\_familias | Admin\_family / Admin | Criar novo convite |
| DELETE | /convites\_familias/{id} | Admin\_family / Admin | Remover convite |
| POST | /convites\_familias/usar | Autenticado | Usar código de convite para entrar na família |

**🔹 6. Produtos (/produtos)**

| **Método** | **Endpoint** | **Acesso** | **Função** |
| --- | --- | --- | --- |
| GET | /produtos | Todos membros da família | Listar produtos da família |
| GET | /produtos/{id} | Todos membros da família | Detalhes do produto |
| POST | /produtos | Todos membros da família | Criar produto |
| PUT | /produtos/{id} | Todos membros da família | Atualizar produto |
| DELETE | /produtos/{id} | Todos membros da família | Remover produto |

**🔹 7. Contas (/contas)**

| **Método** | **Endpoint** | **Acesso** | **Função** |
| --- | --- | --- | --- |
| GET | /contas | Todos membros da família | Listar contas da família |
| GET | /contas/{id} | Todos membros da família | Detalhes da conta |
| GET | /contas/status/{status} | Todos membros da família | Listar contas por status |
| POST | /contas | Todos membros da família | Criar conta |
| PUT | /contas/{id} | Todos membros da família | Atualizar conta |
| DELETE | /contas/{id} | Todos membros da família | Remover conta |

**🔹 8. Pagamentos (/pagamentos)**

| **Método** | **Endpoint** | **Acesso** | **Função** |
| --- | --- | --- | --- |
| GET | /pagamentos | Todos membros da família | Listar pagamentos da família |
| GET | /pagamentos/{id} | Todos membros da família | Detalhes do pagamento |
| GET | /pagamentos/usuario/{id\_usuario} | Todos membros da família | Pagamentos por usuário |
| GET | /pagamentos/conta/{id\_conta} | Todos membros da família | Pagamentos de uma conta |
| POST | /pagamentos | Todos membros da família | Registrar pagamento |
| PUT | /pagamentos/{id} | Todos membros da família | Atualizar pagamento |
| DELETE | /pagamentos/{id} | Todos membros da família | Remover pagamento |

**🔹 9. Compras (/compras)**

| **Método** | **Endpoint** | **Acesso** | **Função** |
| --- | --- | --- | --- |
| GET | /compras | Todos membros da família | Listar compras da família |
| GET | /compras/{id} | Todos membros da família | Detalhes da compra |
| POST | /compras | Todos membros da família | Registrar compra |
| PUT | /compras/{id} | Todos membros da família | Atualizar compra |
| DELETE | /compras/{id} | Todos membros da família | Remover compra |

**🔹 10. Produtos dentro da compra (/produtos\_compra)**

| **Método** | **Endpoint** | **Acesso** | **Função** |
| --- | --- | --- | --- |
| GET | /produtos\_compra/{id\_compra} | Todos membros da família | Listar produtos de uma compra |
| POST | /produtos\_compra | Todos membros da família | Adicionar produto à compra |
| PUT | /produtos\_compra/{id} | Todos membros da família | Atualizar quantidade/preço |
| DELETE | /produtos\_compra/{id} | Todos membros da família | Remover produto da compra |

# Design & Telas: