

UNIFACS

CLARA CERQUEIRA

EDU ARAUJO MARCIEL

JOÃO VICTOR SOUZA MARTINS

LUCAS RIBEIRO DE SOUZA NEVES

MARCOS FELIPE

PEDRO HENRIQUE MOURA DE LIMA

E-COMMERCEXPRESS

Feira de Santana – BH

2024

CLARA CERQUEIRA
EDU ARAUJO MARCIEL
JOÃO VICTOR SOUZA MARTINS
LUCAS RIBEIRO DE SOUZA NEVES
MARCOS FELIPE
PEDRO HENRIQUE MOURA DE LIMA

E-COMMERCEEXPRESS

Este trabalho tem como objetivo desenvolver um sistema escolhido pela equipe utilizando de metodologias ágeis para gerenciamento projetos e técnicas de qualidade de software, levando em consideração as métricas e indicadores de qualidade de software

Orientador: Inácio Borges

Feira de Santana – BH

2024

SUMÁRIO

1.0 Requisitos

- 1.1 Requisitos Funcionais;
- 1.2 Requisitos Não Funcionais.

2.0 UML

- 2.1 Casos de Uso;
- 2.2 Especificação do Caso de Uso;
- 2.3 Diagrama de Classes;
- 2.4 Diagrama de Sequência;
- 2.5 Diagrama de Atividades.

3.0 Banco de Dados

- 3.1 Diagrama Entidade Relacionamento;
- 3.2 Modelo Relacional;
- 3.3 Modelo Físico do BD.

Requisitos

1.1 Requisitos Funcionais

Cadastro de Usuários: Permitir que os usuários se cadastrem na plataforma, com informações como nome, endereço, e-mail e senha.

Catálogo de Produtos: Possibilitar a exibição dos produtos disponíveis para compra, incluindo imagens, descrições e preços.

Carrinho de Compras: Permitir que os usuários adicionem produtos ao carrinho, removam itens e atualizem quantidades.

Processo de Checkout: Oferecer um fluxo de checkout intuitivo, incluindo opções de pagamento e envio.

Gerenciamento de Pedidos: Permitir que os usuários visualizem o status de seus pedidos e histórico de compras.

Sistema de Pagamento: Integrar métodos de pagamento seguros e diversos, como cartão de crédito, transferência bancária e carteiras digitais.

Sistema de Avaliações e Comentários: Permitir que os clientes deixem avaliações e comentários sobre os produtos adquiridos.

Gerenciamento de Inventário: Manter o controle do estoque e atualizar automaticamente a disponibilidade dos produtos.

Funcionalidades de Busca e Filtro: Facilitar a busca por produtos e a navegação através de filtros como categoria, preço e marca.

Integração com Redes Sociais: Permitir compartilhamento de produtos nas redes sociais e login via redes sociais.

1.2 Requisitos Não Funcionais

Desempenho: Garantir que o site seja rápido e responsivo, com tempos de carregamento curtos.

Segurança: Proteger os dados dos usuários, utilizando criptografia SSL, conformidade com normas de segurança e medidas antifraude.

Escalabilidade: Capacidade de lidar com picos de tráfego e aumentar a capacidade do sistema conforme necessário.

Usabilidade: Garantir uma interface amigável e intuitiva para melhor experiência do usuário.

Disponibilidade: Manter o sistema disponível 24/7, com um tempo de inatividade mínimo planejado para manutenção.

Compatibilidade: Certificar-se de que o site seja compatível com diferentes navegadores web e dispositivos, como desktops, tablets e smartphones.

Confiabilidade: Garantir que o sistema seja confiável e livre de falhas, minimizando erros e interrupções.

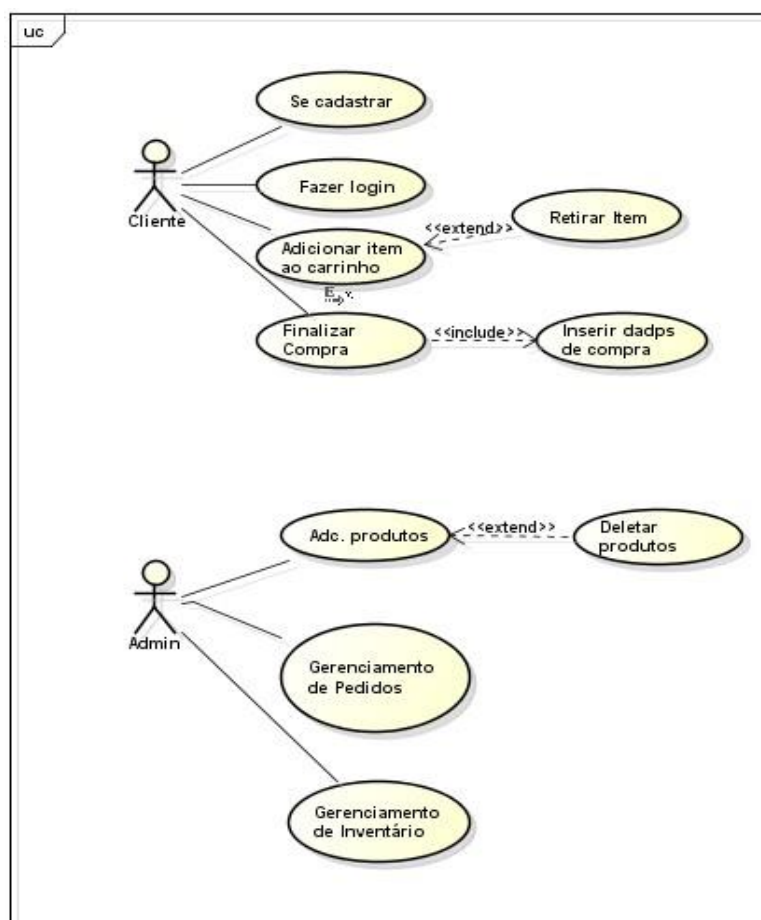
Privacidade: Respeitar a privacidade dos usuários e cumprir regulamentações de proteção de dados, como GDPR e LGPD.

Backup e Recuperação: Implementar rotinas de backup regularmente e garantir a capacidade de recuperação em caso de falhas.

Performance de Busca: Garantir que o mecanismo de busca seja eficiente e forneça resultados precisos e relevantes.

UMLs

2.1 Caso de Uso



2.2 Especificação do Caso de Uso

Atores

1. **Cliente**
2. **Admin**

Casos de Uso Cliente

1. **Se cadastrar**
 - **Descrição:** O cliente se cadastra no sistema criando uma conta com suas informações pessoais.
 - **Pré-condição:** O cliente deve fornecer informações válidas.
 - **Pós-condição:** Uma nova conta é criada no sistema.
- **Fluxo Principal:**
 1. Cliente acessa a página de cadastro.
 2. Cliente preenche o formulário de cadastro com suas informações pessoais.
 3. Cliente submete o formulário.
 4. Sistema verifica a validade das informações.
 5. Sistema cria uma nova conta e confirma o cadastro ao cliente.
- **Fluxo Alternativo:**
 - Se as informações fornecidas são inválidas ou já existem no sistema:
 1. Sistema exibe uma mensagem de erro indicando o problema específico.
 2. Cliente retorna ao formulário para corrigir as informações e submeter novamente.
2. **Fazer login**
 - **Descrição:** O cliente faz login no sistema utilizando suas credenciais.
 - **Pré-condição:** O cliente deve já estar cadastrado no sistema.
 - **Pós-condição:** O cliente está autenticado e pode realizar outras ações.
- **Fluxo Principal:**
 1. Cliente acessa a página de login.
 2. Cliente insere suas credenciais (email e senha).
 3. Cliente submete o formulário.
 4. Sistema verifica as credenciais.
 5. Sistema autentica o cliente e concede acesso ao sistema.
- **Fluxo Alternativo:**
 - Se as credenciais são inválidas:
 1. Sistema exibe uma mensagem de erro indicando o problema.
 2. Cliente retorna ao formulário para corrigir as informações e tentar novamente.
3. **Adicionar item ao carrinho**
 - **Descrição:** O cliente adiciona produtos ao seu carrinho de compras.
 - **Pré-condição:** O cliente deve estar autenticado.
 - **Pós-condição:** O item é adicionado ao carrinho de compras do cliente.
- **Fluxo Principal:**
 1. Cliente navega pelo catálogo de produtos.
 2. Cliente seleciona um produto.

3. Cliente adiciona o produto ao carrinho.
 4. Sistema atualiza o carrinho de compras com o novo item.
- **Fluxo Alternativo:**
 - Se o produto não está disponível:
 1. Sistema exibe uma mensagem de erro indicando que o produto está fora de estoque.
 2. Cliente escolhe outro produto ou tenta novamente mais tarde.
4. **Retirar item**
 - **Descrição:** O cliente pode remover produtos do seu carrinho de compras.
 - **Pré-condição:** O cliente deve estar autenticado e ter itens no carrinho.
 - **Pós-condição:** O item é removido do carrinho de compras do cliente.
 - **Relação de extensão:** Este caso de uso estende "Adicionar item ao carrinho".
- **Fluxo Principal:**
 1. Cliente visualiza o carrinho de compras.
 2. Cliente seleciona um item para remover.
 3. Cliente remove o item.
 4. Sistema atualiza o carrinho de compras, removendo o item selecionado.
- **Fluxo Alternativo:**
 - Se o item não está no carrinho:
 1. Sistema exibe uma mensagem de erro indicando que o item não foi encontrado.
 2. Cliente verifica o carrinho e tenta novamente.
5. **Finalizar compra**
 - **Descrição:** O cliente finaliza a compra dos itens no carrinho.
 - **Pré-condição:** O cliente deve estar autenticado e ter itens no carrinho.
 - **Pós-condição:** A compra é processada.
 - **Relação de inclusão:** Inclui o caso de uso "Inserir dados de compra".
- **Fluxo Principal:**
 1. Cliente visualiza o carrinho de compras.
 2. Cliente seleciona a opção para finalizar a compra.
 3. Sistema solicita os dados de compra (endereço de entrega, informações de pagamento).
 4. Cliente insere os dados de compra.
 5. Sistema processa a compra.
 6. Sistema confirma a compra e envia uma confirmação ao cliente.
- **Fluxo Alternativo:**
 - Se os dados de compra são inválidos:
 1. Sistema exibe uma mensagem de erro indicando o problema.
 2. Cliente retorna ao formulário de dados de compra para corrigir as informações e submeter novamente.
 - Se o pagamento falha:
 1. Sistema exibe uma mensagem de erro indicando a falha no pagamento.
 2. Cliente escolhe um método de pagamento alternativo ou tenta novamente.

6. Inserir dados de compra

- **Descrição:** O cliente insere os dados necessários para concluir a compra, como endereço de entrega e informações de pagamento.
- **Pré-condição:** O cliente deve estar autenticado.
- **Pós-condição:** Os dados são registrados no sistema.

- **Fluxo Principal:**

1. Cliente acessa a página de inserção de dados de compra.
2. Cliente preenche o formulário com os dados de compra necessários.
3. Cliente submete o formulário.
4. Sistema verifica a validade das informações.
5. Sistema registra os dados de compra.

- **Fluxo Alternativo:**

- Se os dados são inválidos:
 1. Sistema exibe uma mensagem de erro indicando o problema específico.
 2. Cliente retorna ao formulário para corrigir as informações e submeter novamente.

Casos de Uso do Admin

1. Adicionar produtos

- **Descrição:** O administrador adiciona novos produtos ao sistema.
- **Pré-condição:** O administrador deve estar autenticado.
- **Pós-condição:** Um novo produto é adicionado ao catálogo do sistema.

- **Fluxo Principal:**

1. Admin acessa a página de gerenciamento de produtos.
2. Admin preenche o formulário com os detalhes do novo produto.
3. Admin submete o formulário.
4. Sistema verifica as informações do produto.
5. Sistema adiciona o novo produto ao catálogo.

- **Fluxo Alternativo:**

- Se as informações são inválidas ou o produto já existe:
 1. Sistema exibe uma mensagem de erro indicando o problema.
 2. Admin retorna ao formulário para corrigir as informações e submeter novamente.

○

2. Deletar produtos

- **Descrição:** O administrador remove produtos do sistema.
- **Pré-condição:** O administrador deve estar autenticado e o produto deve existir no sistema.
- **Pós-condição:** O produto é removido do catálogo do sistema.
- **Relação de extensão:** Este caso de uso estende "Adicionar produtos".

- **Fluxo Principal:**
 1. Admin acessa a página de gerenciamento de produtos.
 2. Admin seleciona um produto para deletar.
 3. Admin confirma a deleção.
 4. Sistema remove o produto do catálogo.
- **Fluxo Alternativo:**
 - Se o produto não existe:
 1. Sistema exibe uma mensagem de erro indicando que o produto não foi encontrado.
 2. Admin verifica a lista de produtos e tenta novamente.

3. Gerenciamento de Pedidos

- **Descrição:** O administrador gerencia os pedidos realizados pelos clientes.
- **Pré-condição:** O administrador deve estar autenticado.
- **Pós-condição:** O status dos pedidos pode ser atualizado e os pedidos podem ser visualizados.

- **Fluxo Principal:**

1. Admin acessa a página de gerenciamento de pedidos.
2. Admin visualiza a lista de pedidos.
3. Admin atualiza o status de um pedido ou visualiza detalhes de um pedido específico.

- **Fluxo Alternativo:**

- Se o pedido não existe ou já foi atualizado:
 1. Sistema exibe uma mensagem de erro indicando o problema.
 2. Admin verifica a lista de pedidos e tenta novamente.

○

4. Gerenciamento de Inventário

- **Descrição:** O administrador gerencia o inventário de produtos no sistema.
- **Pré-condição:** O administrador deve estar autenticado.
- **Pós-condição:** O inventário é atualizado conforme necessário.

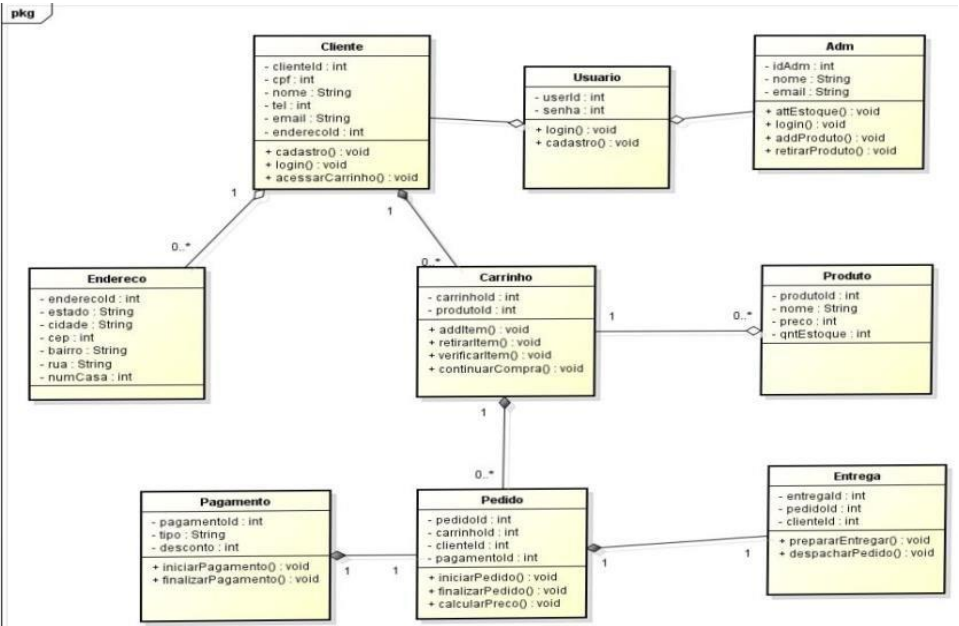
- **Fluxo Principal:**

1. Admin acessa a página de gerenciamento de inventário.
2. Admin visualiza o inventário de produtos.
3. Admin atualiza a quantidade de um produto específico.

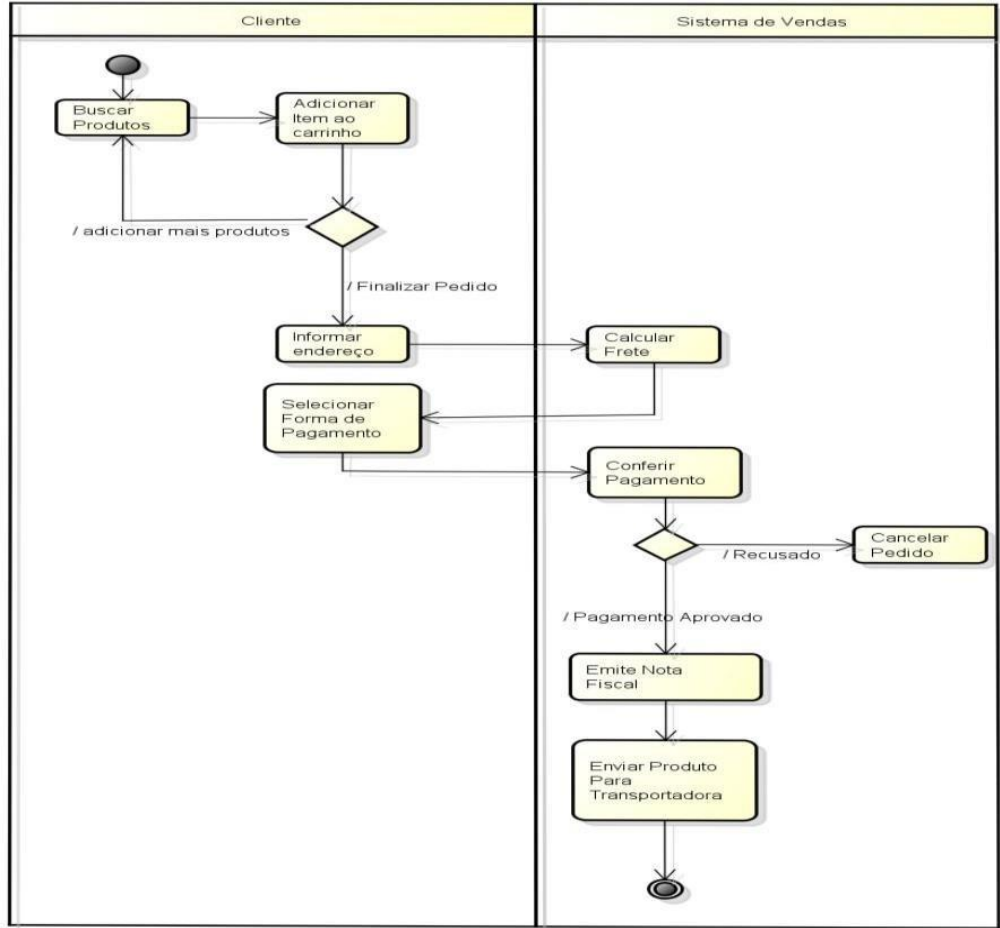
- **Fluxo Alternativo:**

- Se o produto não existe ou a quantidade é inválida:
 1. Sistema exibe uma mensagem de erro indicando o problema.
 2. Admin verifica o inventário e tenta novamente.

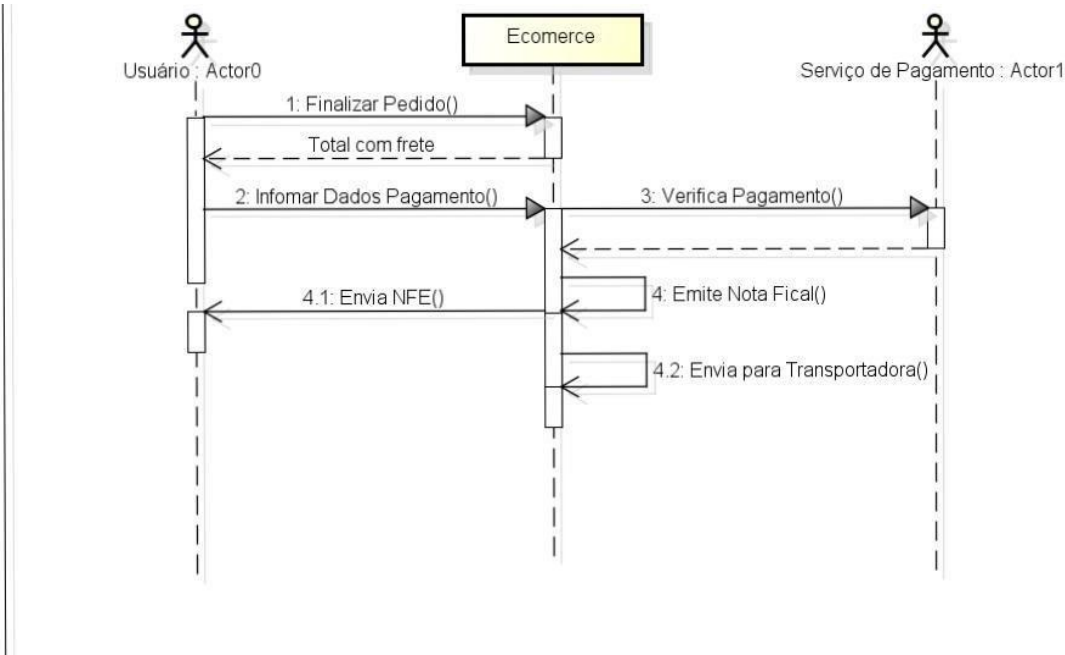
2.3 Diagrama de Classes



2.4 Diagrama de Sequência

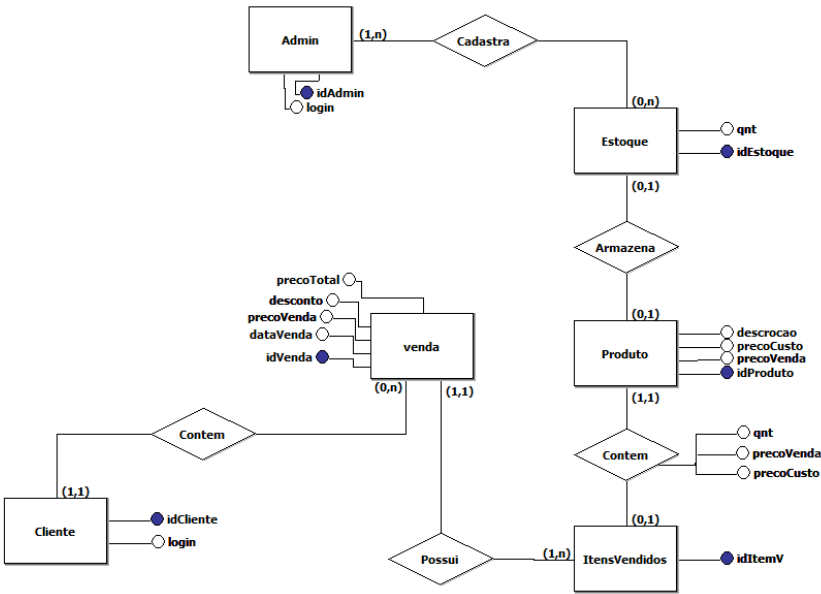


2.5 Diagrama de Atividades

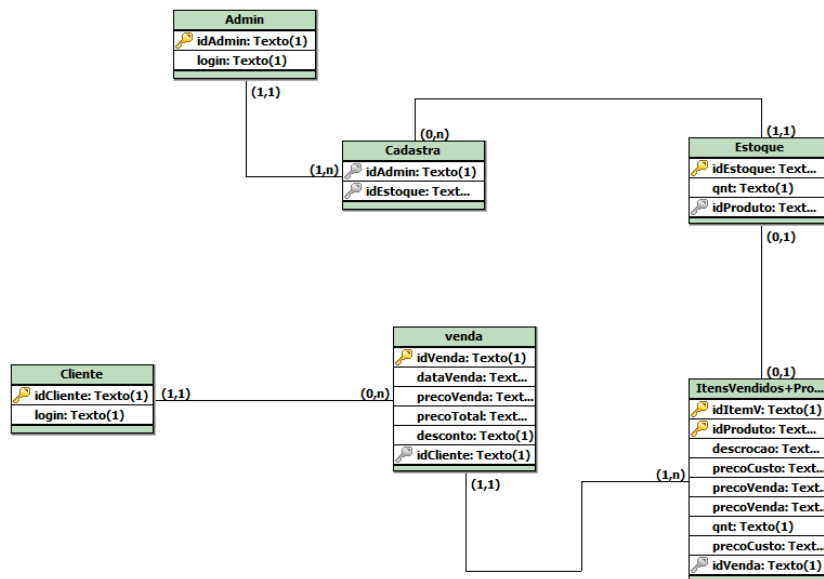


Banco de Dados

3.1 Diagrama Entidade Relacionamento



3.2 Modelo Relacional



3.3 Modelo Físico do BD

```

CREATE TABLE Cliente (
  idCliente varchar(1) PRIMARY KEY,
  login varchar(1)
)

CREATE TABLE Admin (
  idAdmin varchar(1) PRIMARY KEY,
  login varchar(1)
)

CREATE TABLE Estoque (
  idEstoque varchar(1) PRIMARY KEY,
  qnt varchar(1),
  idProduto varchar(1)
)

CREATE TABLE venda (
  idVenda varchar(1) PRIMARY KEY,
  dataVenda varchar(1),
  precoVenda varchar(1),
  precoTotal varchar(1),
  desconto varchar(1),
  idCliente varchar(1),
  FOREIGN KEY(idCliente) REFERENCES Cliente (idCliente)
)

CREATE TABLE ItensVendidos+Produto (
  idItemV varchar(1),
  idProduto varchar(1),
  descricao varchar(1),
  precoCusto varchar(1),
  precoVenda varchar(1),
  qnt varchar(1),
  precoCusto varchar(1),
  idVenda varchar(1),
  PRIMARY KEY(idItemV,idProduto),
  FOREIGN KEY(idVenda) REFERENCES venda (idVenda)
)

CREATE TABLE Cadastra (
  idAdmin varchar(1),
  idEstoque varchar(1),
  FOREIGN KEY(idAdmin) REFERENCES Admin (idAdmin),
  FOREIGN KEY(idEstoque) REFERENCES Estoque (idEstoque)
)

ALTER TABLE Estoque ADD FOREIGN KEY(*erro: ??*) REFERENCES ItensVendidos+Produto (idItemV,idProduto)
  
```