

Checkout Transparente

1. Visão geral

O sistema protótipo apresenta a tela de um produto único. Ao selecionar a opção de compra, o usuário é direcionado a uma página de pagamento. Nesta página, ele pode inserir seus dados e finalizar a transação, escolhendo entre os métodos de pagamento disponíveis: boleto e PIX. Após a conclusão da compra, o usuário receberá um feedback e informações detalhadas sobre a transação.

1.1. Objetivo do projeto

Criação de um protótipo para um sistema de checkout transparente, focado na venda simulada de um produto. A solução permitirá pagamentos diretos no site (Boleto ou PIX), sem redirecionamentos, integrando-se à API Sandbox da Asaas para processamento de pagamentos e a um banco de dados para gerenciamento de clientes e pedidos.

1.2. Requisitos de dados (regras de negócio)

- Um CLIENTE deve se cadastrar com nome, email, cpf_cnpj, cep e num_residencia;
- O EMAIL e CPF_CNPJ deve ser único para cada CLIENTE.
- O sistema deve armazenar o id_cliente_asaas retornado pela API de pagamento.
- Um CLIENTE pode ter muitos PEDIDOS.
- Um PEDIDO deve pertencer a um, e somente a um, CLIENTE.
- Um PRODUTO pode estar em muitos PEDIDOS diferentes.
- Um PEDIDO está associado a um, e somente um, PRODUTO.
- Um PEDIDO deve registrar a quantidade de produtos comprados.
- Um PEDIDO deve armazenar seu status_pagamento, metodo_pagamento, valor_total e o id_pagamento_asaas e data_compra.
- O sistema deve armazenar dados do PRODUTO, PEDIDO e CLIENTE.
- O sistema deve identificar se um cliente é novo ou se já comprou antes no site.

2. Modelo Conceitual

2.1. Entidades Identificadas

- CLIENTE
- PEDIDO
- PRODUTO

2.2. Descrição das entidades e atributos

- CLIENTE
 - id_cliente
 - nome
 - email
 - cpf_cnpj
 - id_cliente_asaas
 - cep
 - num_residencia
- PEDIDO
 - id_pedido
 - id_produto
 - status_pagamento
 - data_compra
 - id_cliente (FK)
 - id_pagamento_asaas
 - valor_total
 - quantidade
- PRODUTO
 - id_produto

- nome_produto
- valor_unidade
- descricao

2.3. Relacionamentos e cardinalidades

Relação: CLIENTE e PEDIDO

Perspectiva A -> B: Um CLIENTE pode ter muitos PEDIDO.

Cardinalidade: (1, n)

Perspectiva B -> A: Um PEDIDO deve pertencer a um CLIENTE.

Cardinalidade: (1, 1)

Resumo da Relação: CLIENTE (1,N) --- (1,1) PEDIDO

Relação: PEDIDO e PRODUTO

Perspectiva A -> B: Um PEDIDO pode ter um PRODUTO.

Cardinalidade: (1, 1)

Perspectiva B -> A: Um PRODUTO pode estar em zero ou em muitos PEDIDOS.

Cardinalidade: (0, n)

Resumo da Relação: PEDIDO (1,1) --- (0,N) PRODUTO

3. Modelo Lógico

Tabela **CLIENTE**

COLUNA	TIPO	CHAVE	RESTRIÇÕES	NOTAS
id_cliente	inteiro	primária	autoincremento	ID único do cliente

nome	caractere		não nulo	
email	caractere		única	
cpf_cnpj	caractere		única	
cep	inteiro		não nulo	
num_residencia	caractere		não nulo	Num. da casa do cliente
id_cliente_asaas	caractere		nulo	ID retornado pela API do Asaas

Tabela **PEDIDO**

COLUNA	TIPO	CHAVE	RESTRIÇÕES	NOTAS
id_pedido	inteiro	primária	autoincremento	ID único do pedido
id_cliente	inteiro	estrangeira	não nulo	Quem comprou
id_produto	inteiro	estrangeira	não nulo	O que comprou
quantidade	inteiro		não nulo com valor padrão	Unidades do produto
valor_total	decimal		não nulo	
status_pagamento	caractere		não nulo com valor padrão	
data_compra	data		não nulo	
id_pagamento_asaas	caractere		nulo e único	ID da transação no Asaas

Tabela **PRODUTO**

COLUNA	TIPO	CHAVE	RESTRIÇÕES	NOTAS
id_produto	inteiro	primária	autoincremento	ID único do produto
nome_produto	caractere		não nulo	
valor_produto	decimal		não nulo	

descricao	texto		nulo	
-----------	-------	--	------	--

3.1. Diagrama Entidade-Relacionamento (DER)

O DER está estruturado no brModelo, que você pode ver [aqui](#).

3.2. Dicionário de Dados (Esquema)

O dicionário de dados pode ser visto [aqui](#).

4. Modelo Físico

4.1. Informações do Ambiente

- SGBD Alvo: PostgreSQL
- Plataforma de hospedagem: Supabase

4.2. Scripts DDL (Data Definition Language)

O script de criação das tabelas está hospedado [aqui](#).