



# Banco de dados StoreAPI

## Tabelas:

- Usuário (User): informações do usuário, autenticação, perfil.
- Produto (Product): dados do produto, preço, descrição, imagens, estoque.
- Categoria (Category): categorias para organizar produtos.
- Pedido (Order): registro de compras feitas por usuários.
- Item do Pedido (OrderItem): produtos dentro de cada pedido.
- Carrinho (Cart): itens que o usuário está considerando comprar.
- Pagamento (Payment): informações sobre pagamentos e status.

## Relacionamentos:

## Modelo de Relacionamentos de Banco de Dados (E-commerce)

### Usuários e Pedidos

- **Relação:** `users_tb (1) <--> (1,N) orders_tb`
- **Tipo:** Um para Muitos (1:N)
- **Descrição:**
  - Um usuário (`users_tb`) pode ter **um ou vários** pedidos (`orders_tb`).
  - Cada pedido (`orders_tb`) pertence a **exatamente um** usuário (`users_tb`).

### Usuários e Carrinhos

- **Relação:** `users_tb (1) <--> (1,1) carts_tb`
- **Tipo:** Um para Um (1:1)
- **Descrição:**
  - Um usuário (`users_tb`) possui **exatamente um** carrinho (`carts_tb`).
  - Cada carrinho (`carts_tb`) pertence a **exatamente um** usuário (`users_tb`).

### Categorias e Produtos

- **Relação:** `categories_tb (1) <--> (1,N) products_tb`
- **Tipo:** Um para Muitos (1:N)
- **Descrição:**
  - Uma categoria (`categories_tb`) pode conter **um ou vários** produtos (`products_tb`).
  - Cada produto (`products_tb`) pertence a **exatamente uma** categoria (`categories_tb`).



# Carrinhos e Itens do Carrinho

- **Relação:** `carts_tb (1) <--> (0,N) cart_items_tb`
- **Tipo:** Um para Muitos (1:N)
- **Descrição:**
  - Um carrinho (`carts_tb`) pode ter **zero ou vários** itens (`cart_items_tb`). Um carrinho pode estar vazio.
  - Cada item de carrinho (`cart_items_tb`) pertence a **exatamente um** carrinho (`carts_tb`).

## Produtos e Itens do Carrinho

- **Relação:** `products_tb (1) <--> (0,N) cart_items_tb`
- **Tipo:** Um para Muitos (1:N)
- **Descrição:**
  - Um produto (`products_tb`) pode estar presente em **zero ou vários** itens de carrinho (em carrinhos de diferentes usuários).
  - Cada item de carrinho (`cart_items_tb`) refere-se a **exatamente um** produto (`products_tb`).

**Nota:** A tabela `cart_items_tb` funciona como uma **tabela de junção** (ou tabela associativa), que efetivamente cria um relacionamento **Muitos para Muitos (N:M)** entre `carts_tb` e `products_tb`.

## Pedidos e Itens do Pedido

- **Relação:** `orders_tb (1) <--> (1,N) order_items_tb`
- **Tipo:** Um para Muitos (1:N)
- **Descrição:**
  - Um pedido (`orders_tb`) deve conter **pelo menos um ou vários** itens (`order_items_tb`). Um pedido não pode estar vazio.
  - Cada item de pedido (`order_items_tb`) pertence a **exatamente um** pedido (`orders_tb`).

## Produtos e Itens do Pedido

- **Relação:** `products_tb (1) <--> (0,N) order_items_tb`
- **Tipo:** Um para Muitos (1:N)
- **Descrição:**
  - Um produto (`products_tb`) pode estar presente em **zero ou vários** itens de pedido (em pedidos diferentes).
  - Cada item de pedido (`order_items_tb`) refere-se a **exatamente um** produto (`products_tb`).

**Nota:** Similarmente, a tabela `order_items_tb` é uma **tabela de junção** que cria um relacionamento **Muitos para Muitos (N:M)** entre `orders_tb` e `products_tb`.

## Pedidos e Pagamentos

- **Relação:** `orders_tb (1) <--> (1,N) payments_tb`
- **Tipo:** Um para Muitos (1:N)
- **Descrição:**
  - Um pedido (`orders_tb`) pode ter **um ou vários** pagamentos (`payments_tb`), permitindo pagamentos parcelados ou com múltiplos métodos.
  - Cada pagamento (`payments_tb`) está associado a **exatamente um** pedido (`orders_tb`).
  - Lista