

Documento de Requisitos (Projeto Y)

Versão 1.0

Histórico de Alterações

Data	Versão	Descrição	Autor
25/04/2023	1.0	Criação do documento de requisitos	Guilherme Victor
06/06/2023	1.1	Recriação do documento de requisitos	Guilherme Victor
06/06/2023	1.2	Criação do documento relacional	Guilherme Victor

Conteúdo

1. Introdução	4
1.1 Visão geral do documento.....	4
2. Visão geral do sistema.....	4
2.1 Descrição dos Usuários.....	5
3. Requisitos Funcionais.....	5
Apêndices.....	6

• Introdução

Este documento especifica os requisitos do sistema Modelo Curso, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

• Visão geral do documento

Além desta seção introdutória, as seções seguintes estão organizadas como descrito abaixo.

• Apêndices:

- - **Modelo de funcionalidades:** apresenta uma ilustração do modelo de casos de uso do sistema.
- - **Descrição da interface com o usuário:** apresenta um esboço da interface gráfica do usuário.
- - **Descrição dos relatórios do sistema:** apresenta um esboço dos relatórios emitidos pelo sistema.

-
-
-
-
-
-
-
-
-

-
-
-
-
-

- **Visão geral do sistema**

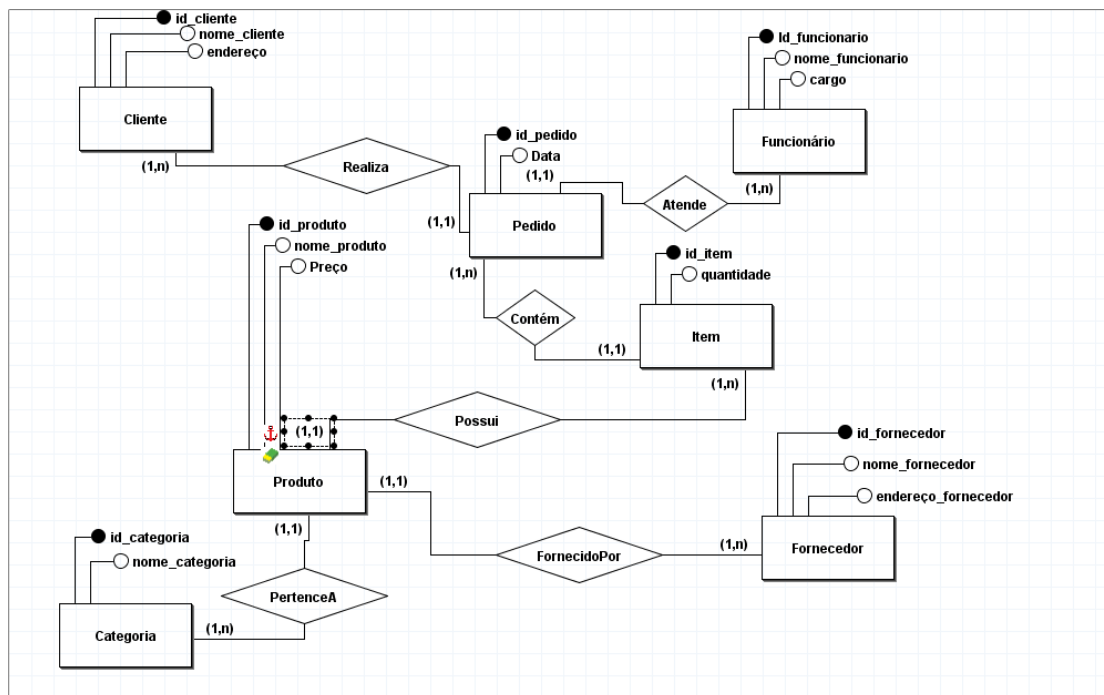
O sistema proposto é um sistema de gerenciamento de pedidos e produtos voltado para empresas que lidam com vendas e estoque. Ele permite que os clientes façam pedidos de produtos a serem atendidos pelos funcionários. O pedido contém informações como itens necessários e quantidades, e o sistema calcula o valor total do pedido. Além disso, o sistema mantém informações atualizadas sobre produtos oferecidos por diversos fornecedores e permite a busca de produtos de acordo com critérios específicos, proporcionando uma visão geral e um controle eficiente do pedido e do fluxo de estoque.

- **Descrição dos Usuários**

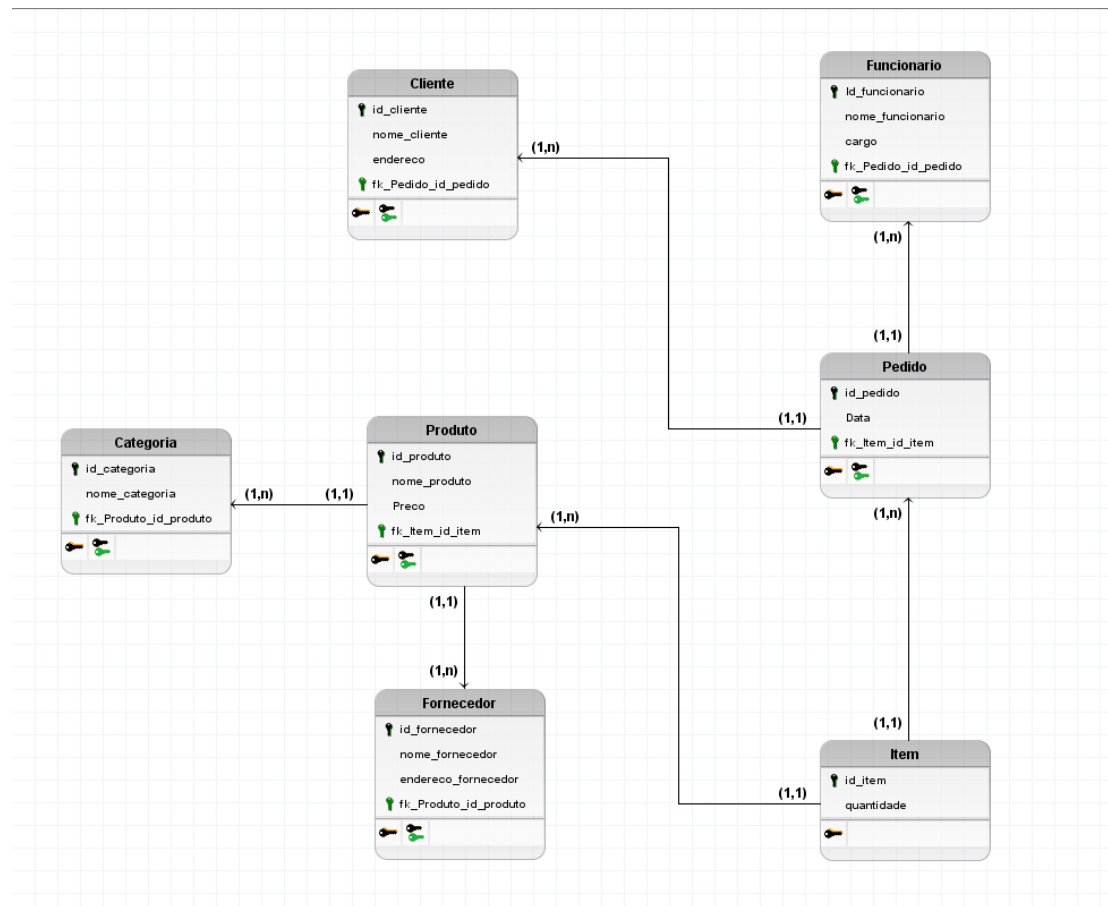
- **Cliente:** O cliente é um usuário do sistema que realiza pedidos de produtos. Ele pode criar novos pedidos, adicionar itens e especificar as quantidades desejadas. O cliente também pode consultar o status dos pedidos e atualizar informações pessoais, como nome e endereço.
- **Funcionário:** O funcionário é responsável por atender e gerenciar os pedidos dos clientes. Ele tem acesso ao sistema para visualizar os pedidos em andamento, atualizar informações dos pedidos, como a data, e adicionar/remover itens. O funcionário também pode atualizar informações de produtos e categorias, fornecidas pelos fornecedores, além de realizar buscas de produtos para auxiliar os clientes.
- **Fornecedor:** O fornecedor é um usuário do sistema responsável por fornecer informações atualizadas sobre os produtos que oferece. Ele pode adicionar novos produtos, atualizar preços e disponibilidade dos itens fornecidos. O fornecedor também pode visualizar os pedidos realizados pelos clientes que envolvem seus produtos.
- **Administrador:** O administrador é responsável pela gestão geral do sistema. Ele tem acesso privilegiado e pode criar, modificar e excluir registros de usuários, incluindo clientes, funcionários e fornecedores. Além disso, o administrador pode monitorar o desempenho do sistema, gerenciar permissões de acesso e garantir a segurança e integridade dos dados.

- **Requisitos Funcionais**

- 1: O sistema deve permitir que um cliente crie um novo pedido, fornecendo informações como os itens desejados e a quantidade de cada item.
- 2: O sistema deve permitir que um funcionário consulte e atualize as informações de um pedido existente, incluindo a data do pedido e os itens associados.
- 3: O sistema deve ser capaz de calcular o valor total de um pedido com base nos preços dos itens selecionados pelo cliente.
- 4: O sistema deve permitir que um fornecedor atualize as informações sobre os produtos que ele fornece, incluindo o nome, o preço e a disponibilidade.
- 5: O sistema deve fornecer uma funcionalidade de busca que permita aos usuários pesquisar produtos com base em critérios como o nome, a categoria ou o fornecedor.



- **Modelo Relacional**



- DDL
- O Sistema de Gerenciamento de Banco de Dados utilizado foi o SQLite

```

CREATE TABLE Cliente (
    idcliente    INTEGER PRIMARY KEY AUTOINCREMENT,
    Nome_Cliente TEXT    NOT NULL,
    Endereço     TEXT
);
    
```

```

CREATE TABLE Produto (
    
```

```

    idproduto    INTEGER PRIMARY KEY AUTOINCREMENT,

    Nome_Produto TEXT    NOT NULL,

    Preço        REAL

);

CREATE TABLE Pedido (

    idpedido     INTEGER PRIMARY KEY AUTOINCREMENT,

    Data         NUMERIC,

    ClienteID    INTEGER REFERENCES Cliente (idcliente),

    FuncionarioID INTEGER REFERENCES Funcionario (idfuncionario),

    ItemID       INTEGER REFERENCES Item (iditem)

);

CREATE TABLE Item (

    iditem       INTEGER PRIMARY KEY AUTOINCREMENT,

    Quantidade INTEGER,

    ProdutoID    INTEGER REFERENCES Produto (idproduto)

);

CREATE TABLE Funcionario (

    idfuncionario INTEGER PRIMARY KEY AUTOINCREMENT,

    Nome_Funcionario TEXT    NOT NULL,

    Cargo        TEXT

);

CREATE TABLE Fornecedor (

```

```
Nome_Fornecedor TEXT NOT NULL,  
  
Endereço TEXT  
);
```

```
CREATE TABLE Categoria (  
  
idcategoria INTEGER PRIMARY KEY AUTOINCREMENT,  
  
Nome_Categoria TEXT NOT NULL  
);
```

- DML
- insert

```
INSERT INTO Cliente (Nome_Cliente, Endereço) VALUES ('Guilherme Victor','REC' );
```

```
INSERT INTO Cliente (Nome_Cliente, Endereço) VALUES ( 'Tiago Ribeiro', 'REC');
```

```
INSERT INTO Cliente (Nome_Cliente, Endereço) VALUES ( 'Gabriel Silva', 'REC');
```

```
INSERT INTO Cliente (Nome_Cliente, Endereço) VALUES ( 'Mateus Araujo', 'REC');
```

```
INSERT INTO Funcionario (Nome_Funcionario, Cargo ) VALUES ( 'Pedro', 'Vendedor' );
```

```
INSERT INTO Funcionario (Nome_Funcionario, Cargo ) VALUES ( 'Jorge', 'Gerente' );
```

```
INSERT INTO Categoria ( Nome_Categoria ) VALUES ( 'Comida' );
```

```
INSERT INTO Categoria ( Nome_Categoria ) VALUES ( 'Higiene pessoal' );
```

```
INSERT INTO Categoria ( Nome_Categoria ) VALUES ( 'Outros' );
```

```
INSERT INTO Produto ( Nome_Produto, Preço ) VALUES ( 'Amendoim', '10' );
```

```
INSERT INTO Produto ( Nome_Produto, Preço ) VALUES ( 'Shampoo', '15' );
```



```
INSERT INTO Fornecedor ( Nome_Fornecedor, Endereço ) VALUES ( 'Santa loki', 'REC' );
```

```
INSERT INTO Fornecedor ( Nome_Fornecedor, Endereço ) VALUES ( 'Brilho Eterno', 'SP' );
```

```
INSERT INTO Item ( Quantidade, ProdutoID ) VALUES ( '10', '1' );
```

```
INSERT INTO Item ( Quantidade, ProdutoID ) VALUES ( '2', '1' );
```

```
INSERT INTO Item ( Quantidade, ProdutoID ) VALUES ( '15', '2' );
```

```
INSERT INTO Item ( Quantidade, ProdutoID ) VALUES ( '100', '2' );
```

```
INSERT INTO Pedido ( Data, ClienteID, FuncionarioID, ItemID ) VALUES ( '05052023', '12', '1','1' );
```

```
INSERT INTO Pedido ( Data, ClienteID, FuncionarioID, ItemID ) VALUES ( '07052023', '13', '1','2' );
```

```
INSERT INTO Pedido ( Data, ClienteID, FuncionarioID, ItemID ) VALUES ( '02062023', '14', '1','3' );
```

```
INSERT INTO Pedido ( Data, ClienteID, FuncionarioID, ItemID ) VALUES ( '04062023', '15', '1','4' );
```

- Update

```
UPDATE Pedido SET FuncionarioID = '2' WHERE idpedido = 5 ;
```

- Delete

```
DELETE FROM Categoria WHERE idcategoria = 3 ;
```

- Relatórios

*Mostre todos os dados dos pedidos de produto 2

```
Select * from Pedido where ItemID = 2 ;
```

*Mostre todos os dados dos pedido atendidos pelo gerente

```
Select *
```

```
from Pedido
```

```
where FuncionarioID like 2
```

*Quantidade de pedidos por cliente

```
SELECT
```

```
    Pedido.ClienteID AS "ID Cliente",
```

```
    Cliente.Nome_Cliente AS "Nome Cliente",
```

```
    (
```

```
        SELECT COUNT(ClienteID) FROM Pedido
```

```
        WHERE Pedido.ClienteID = Cliente.idcliente
```

```
    ) AS "Quant Pedidos"
```

```
FROM Pedido
```

```
LEFT JOIN CLIENTE
```

```
    ON Pedido.ClienteID = Cliente.idcliente
```

```
GROUP BY
```

idcliente,

Nome_Cliente

- Link

<https://github.com/reedgv/Faculdade/blob/5ecafff4cdc0e0d68fed95fa0376ac93aea8037b/Projeto%20Y.db>