

## **Escopo do Projeto:**

---

Equipe de Desenvolvimento

Desenvolvedor 1: Ygor Andrade, RA 14378.

Desenvolvedor 2: Mickael M. Mineo, RA 14169.

Desenvolvedor 3: Hiago H, Tavares, RA 06885.

### **1. Definição do Projeto**

- Nome do Projeto: Marcenaria Extrema.
- Objetivo: Desenvolver uma plataforma de gerenciamento de produtos e seus fornecedores, clientes e pedidos, além de gerar relatórios do inventário e dos produtos mais solicitados, de uma marcenaria que está querendo expandir os seus negócios na cidade de Extrema. Essa plataforma permite a criação, modificação e remoção de pedidos.
- Escopo Funcional: cadastrar, gerenciar e deletar produtos e clientes.
  - Cadastro e Modificação de Clientes, Produtos, Pedidos e Fornecedores;
  - Gerenciamento de pedidos, relatórios e inventário;
  - Criação de Relatórios e inventário;

### **2. Tecnologias Utilizadas**

- Linguagem de Programação: PHP com foco na linguagem de Programação Orientada a Objetos.
- Banco de Dados: Supabase.
- Frontend:
  - HTML5: Interface;
  - CSS3: Design do website;
  - Javascript: Manipulação de Objetos como formulários, listas, entre outros.
- Servidor: Supabase.
- Controle de Versão: Git e GitHub

### **3. Requisitos do Sistema**

- Funcionais:
  - Gerenciamento de Produtos:
    - CRUD completo;
    - Produtos podem ser alterados ou deletados;
    - Um fornecedor também pode ser um cliente.
  - Gerenciamento de Clientes:
    - CRUD completo;
    - Clientes podem ser alterados ou deletados;

- Um cliente pode ter vários endereços e telefones e e-mail, porém a sua região terá que ser especificada (Exemplo: Fornecedor SP, Fornecedor MG, etc.).
- o Gerenciamento de Pedidos:
  - CRUD completo;
  - Um pedido pode ter vários clientes, desde que sejam da mesma marca (Exemplo: Fornecedor SP, Fornecedor MG fizeram uma requisição de 50 unidades de madeira);
  - Pedidos podem ser tanto para importar ou exportar produtos;
  - Se um pedido for para exportar um produto a data atual irá ser atribuída ao mesmo e a quantidade dos produtos solicitados serão subtraídos do estoque. Se eles forem importados, serão adicionados no estoque;
  - Pedidos podem ser modificados caso sejam cancelados ou finalizados.
- o Gerenciamento de Fornecedores:
  - CRUD completo;
  - Um fornecedor também pode ser um cliente;
- o Gerenciamento de Relatórios:
  - CRUD completo;
  - Será possível criar dois relatórios: Relatórios de Compra e Relatórios de venda;
  - Os relatórios de venda irão mostrar o total de itens vendidos no mês, quando foram vendidos e qual foi o lucro mensal de cada um;
  - Os relatórios de compra irão mostrar o total de produtos comprados (contendo o nome, preço e descrição) por fornecedor.
  - Será possível filtrar ambos os relatórios para ter uma visão melhor de um produto específico
  - Caso um cliente tenha feito vários pedidos ou um a loja ter comprado de um fornecedor várias vezes, ele será atribuído ao relatório como uma entidade de longo prazo.
- o Gerenciamento de Inventário:
  - CRUD completo;
  - Será adicionado no inventário todos os produtos que ainda estão no estoque;

- Observações poderão ser feitas sobre os produtos, seja da condição que eles foram recebidos pelo responsável do almoxarifado, a falta de certos produtos essenciais no estoque, entre outros.
- Gerenciamento de Usuários:
  - CRUD Completo;
  - Cada setor terá pelo menos um administrador;
  - Cada usuário terá um nível de controle (1 para controle administrativo, 2 para edição de certos arquivos sensíveis, 3 para editar arquivos públicos).
- Não Funcionais:
  - Interface amigável;
  - Prevenção de ataques cibernéticos.

#### **4. Estrutura de Banco de Dados**

- Usuário:
  - id;
  - nome;
  - nivel\_controle.
- Cliente:
  - id (Chave Primária);
  - nome;
  - cnpj (opcional);
  - email (pode ser registrado mais de um);
  - telefone (pode ser registrado mais de um);
  - endereco (pode ser registrado mais de um).
- Produto:
  - id (Chave Primária);
  - nome;
  - quantidade;
  - preco;
  - descricao.
- Fornecedor:
  - cnpj (Chave Primária);
  - nome;
  - produto;
  - quantidade.
- Venda:
  - id (Chave Primária);

- cliente\_id;(FK)
  - Produto\_id; (FK)
  - quantidade;
  - valor;
  - data\_venda;
  - status (Se o pedido foi entregue ou não).
- Compra:
    - id (Chave Primária);
    - fornecedor\_id;(FK)
    - Produto\_id(FK)
    - quantidade;
    - valor;
    - data\_compra;
    - status (Se o pedido foi entregue ou não).

## 5. Estrutura do Projeto em PHP

- Model-View-Controller (MVC):
  - Modelos (Classes que representam as entidades do banco de dados):
    - ♣ Cada tabela do banco de dados deve ter uma classe correspondente. Exemplo: Usuario, Produto, Pedido.
  - Controladores:
    - ♣ Definir controladores para manipular as ações do sistema. Exemplo: Cliente\_Controller para lidar com o cadastro de clientes, Produto\_Controller para gerenciar produtos.
  - Views (Interfaces de apresentação):
    - ♣ Usar templates HTML/PHP para exibir dados. As views devem ser responsáveis apenas pela interface.
- Classes e Métodos:
  - Cada classe deve ser modular e encapsular as regras de negócio relacionadas a ela. Exemplo: Métodos de criação e gerenciamento de relatórios, métodos de gerenciamento de estoque na classe Produto.
- Conexão com o Banco de Dados:
  - Usar PDO para lidar com as transações do banco de dados.
  - Implementar uma classe Database para abstrair a conexão com o banco e facilitar o uso em outras partes do código.

## 6. Planejamento das Funcionalidades

- Criação e Modificação:
  - Adicionar, remover e editar clientes, produtos, relatórios, fornecedores e relatórios. Criar relações entre produtos, fornecedores e clientes caso possuam algum atributo em comum.
- CRUD:
  - Interfaces para adicionar, editar, deletar e listar produtos, clientes, relatórios e fornecedores. Cada um desses terá uma categoria e um método de pesquisa por nome.
- Relatórios:

- o Criação de relatórios de vendas e de estoque com opções de filtragem por data ou produto.

## **7. Cronograma**

- Divisão em Sprints: Definir entregas incrementais, priorizando as funcionalidades principais nas primeiras fases.
- Estimativa de prazo para cada etapa:
  - o Design da interface: 3 semanas
  - o Implementação do backend: 3 semanas
  - o Integração e testes: 1 semana