



UEPB

Universidade
Estadual da Paraíba

Universidade Estadual da Paraíba
CAMPUS I – Campina Grande

Professor:	Allan Vilar de Carvalho
Curso:	Bacharelado em Computação
Componente curricular:	Programação Web

Projeto (Parte 1)

UNIVERSIDADE ESTADUAL DA PARAÍBA

DANILO PEDRO DA SILVA VALÉRIO

JOÃO VITOR AGUIAR DE SOUSA

MARIA EDUARDA DA SILVA

Documento de Visão do VL Store

Campina Grande - PB

2025

DANILO PEDRO DA SILVA VALÉRIO
JOÃO VITOR AGUIAR DE SOUSA
MARIA EDUARDA DA SILVA

Documento de Visão do VL Store

Este trabalho, intitulado "Documento de Visão do VL Store", é apresentado ao curso de Ciência da Computação, como parte dos requisitos necessários à obtenção da nota da disciplina de Programação Web, visando o desenvolvimento de um site funcional e de qualidade.

Professor: Allan Villar de Carvalho

1. Resumo:

O avanço das tecnologias web tem impulsionado a criação de marketplaces modernos, capazes de oferecer soluções eficientes para gestão de produtos e vendedores. Este trabalho apresenta o desenvolvimento de um sistema de sistema de gestão de pequenos comércios simplificado, focado na usabilidade e na escalabilidade. A aplicação será construída utilizando Node.js com Express.js e typescript no back-end, React e e Bootstrap no front-end, javascript em ambos, além do banco de dados MongoDB para armazenar as informações. O sistema permite o gerenciamento de lojas, produtos e compradores, garantindo uma experiência dinâmica e intuitiva. A adoção da arquitetura MVC possibilitou uma organização modular do código, favorecendo a manutenção e evolução da aplicação. Além disso, foram implementadas boas práticas de desenvolvimento web, como a criação de APIs RESTful para comunicação entre as camadas do sistema. Com isso, o projeto busca demonstrar a

Palavras-chave: Marketplace, Desenvolvimento Web, Node.js, Express.js, React, Next.js, Bootstrap, MongoDB.

Abstract

This work presents the development of a simplified marketplace using an architecture based on Node.js with Express.js for the back-end, React with Next.js and Bootstrap for the front-end, and MongoDB as the database. The system aims to provide an efficient platform for managing purchases and sales, allowing the registration, management, and visualization of products, stores, and buyers. The application follows the MVC architectural pattern, ensuring the separation between the user interface, business logic, and persistence layer. Additionally, best development practices are adopted, such as code modularization and the use of RESTful APIs for communication between the front-end and back-end. The choice of technologies seeks to provide scalability, flexibility, and a responsive user experience.

Keywords: Marketplace, Store Management, Commercial Management, Node.js, Express.js, React, Next.js, Bootstrap, MongoDB, MVC, Web Development.

2. Descrição do Sistema

O vl-store é um software de gestão comercial projetado para auxiliar no gerenciamento de produtos, funcionários, vendas e caixas. O sistema permite um controle abrangente do estoque, transações financeiras e desempenho dos vendedores, garantindo que o gestor tenha total visibilidade e controle sobre o seu negócio.

3. Requisitos funcionais

4. Nome: Gerenciamento de Produtos

Descrição:

Esta funcionalidade possibilita o gerenciamento completo do cadastro de produtos, permitindo operações de CRUD (Criar, Ler, Atualizar e Deletar) com acesso restrito conforme o perfil do usuário (apenas administradores têm permissão para alterações).

a. Cadastrar Produtos

Descrição:

Permite a inclusão de novos produtos no sistema com os seguintes atributos:

- **Título:** Nome ou identificação do produto.
- **Categoria:** Classificação do produto para organização e busca.
- **Material:** Informação sobre a composição ou material do produto.
- **Referência:** Código numérico de 4 dígitos gerado aleatoriamente e que não poderá ser alterado após o cadastro.
- **Gênero:** Campo opcional para segmentação (ex.: masculino, feminino ou unissex).

- **Variações:** Cada produto pode ter uma ou mais variações, onde cada variação deve conter:
 - **Descrição da variação:** Detalhes específicos (ex.: cor, tamanho, modelo).
 - **Quantidade:** Estoque disponível para a variação.
 - **Valor:** Preço aplicado à variação.

Permissões:

- **Administrador (usuário gerente):** Exclusivo para o cadastro de produtos.

b. Leitura dos Produtos

Descrição:

Exibição dos produtos cadastrados em uma tabela interativa que possibilita:

- **Busca:** Pesquisa por título, categoria ou referência.
- **Visualização Detalhada:** Exibição completa dos dados do produto, incluindo suas variações, ao selecionar um item.

Permissões:

- **Administrador:** Acesso completo.
- **Vendedor:** Pode realizar consulta, mas sem acesso às funcionalidades de edição ou exclusão.

c. Atualizar Produtos

Descrição:

Permite a edição dos dados de um produto existente, com a ressalva de que o campo **Referência** é imutável.

- **Campos Editáveis:** Título, Categoria, Material, Gênero e informações das variações (descrição, quantidade e valor).
- **Processo:** A atualização deve ser feita a partir da janela de visualização detalhada do produto, garantindo que as alterações sejam salvas de forma consistente.

Permissões:

- **Administrador:** Exclusivo para realizar atualizações.

d. Deletar Produtos

Descrição:

Habilita a exclusão de um produto do sistema, removendo todas as informações associadas, inclusive suas variações.

Permissões:

- **Administrador:** Exclusivo para a exclusão de produtos.

5. Nome: Controle de Funcionários

Descrição:

Esta funcionalidade gerencia o cadastro e as informações dos funcionários, permitindo operações de CRUD (Criar, Ler, Atualizar e Deletar) e a visualização de dados complementares (como vendas e vales), com acesso restrito ao administrador.

a. Cadastrar Funcionário

Descrição:

Permite o registro de novos funcionários com os seguintes dados:

- **Nome:** Nome completo do funcionário.
- **Cargo:** Posição ou função desempenhada na empresa.
- **Data de Nascimento:** Informação necessária para controle e conformidade com políticas internas.
- **Telefone:** Contato para comunicação.
- **CPF:** Identificação única do funcionário.
- **Senha:** Senha para o funcionário realizar login.

Permissões:

- **Administrador (usuário gerente):** Exclusivo para o cadastro de funcionários.

b. Leitura dos Funcionários

Descrição:

Exibição dos funcionários cadastrados em uma tabela que permite:

- **Busca e Filtro:** Pesquisa por nome, cargo ou CPF.
- **Visualização Detalhada:** Ao selecionar um funcionário, o sistema apresenta todas as informações cadastradas e, adicionalmente, dados sobre vendas e vales – dados esses que são acessíveis somente ao administrador.

Permissões:

- **Administrador:** Visualiza todas as informações detalhadas e históricas.

c. Atualizar Funcionários

Descrição:

Facilita a edição das informações de um funcionário já cadastrado.

- **Campos Editáveis:** Nome, Cargo, Data de Nascimento, Telefone e outros dados que possam precisar de atualização.
- **Restrições:** Dados únicos (como CPF)..

Permissões:

- **Administrador:** Exclusivo para a realização de atualizações.

d. Deletar Funcionários

Descrição:

Permite a remoção do cadastro de um funcionário do sistema.

- **Confirmação:** Deve haver um mecanismo de confirmação para evitar a exclusão acidental dos dados.

Permissões:

- **Administrador:** Exclusivo para a exclusão de funcionários.

6. Nome: Gerenciamento de Vendas

Descrição:

Funcionalidade que permite registrar e gerenciar as vendas realizadas, com permissões diferenciadas para vendedores (que podem registrar e visualizar apenas suas vendas) e administradores (com acesso completo a todas as operações).

a. Registrar Venda

Descrição:

Permite o registro de uma nova venda. Ao registrar, o sistema deverá capturar os seguintes dados:

- **Código da venda:** Identificador único para a transação.
- **Lista de produtos vendidos:** Incluindo identificação, quantidade e valor unitário de cada produto.
- **Valor total:** Soma dos valores dos produtos, aplicando possíveis descontos ou acréscimos.
- **Forma de pagamento:** Opções como dinheiro, cartão ou outros métodos.
- **Vendedor responsável:** Identificação do usuário que registrou a venda.
- **Data e hora da transação.**

Permissões:

- **Vendedor:** Pode registrar suas próprias vendas.
- **Administrador:** Pode registrar qualquer venda, se necessário.

b. Visualizar Vendas

Descrição:

Exibe uma lista de todas as vendas registradas em formato de tabela, com possibilidade de:

- **Filtrar por vendedor e forma de pagamento.**

Permissões:

- **Administrador:** Visualiza todas as vendas.

c. Detalhar Venda

Descrição:

Ao selecionar uma venda na listagem, o sistema apresentará uma visão detalhada com:

- **Informações completas dos produtos vendidos (descrição, quantidade, valor unitário).**
- **Resumo da transação (valor total, forma de pagamento e dados do vendedor).**

Permissões:

- **Administrador:** Detalhamento de todas as vendas.

d. Deletar Venda

Descrição:

Possibilita a exclusão de uma venda.

Permissões:

- **Administrador:** Exclusão autorizada.
- **Vendedor:** Não é autorizado a deletar vendas.

7. Nome: Controle de Caixa

Descrição:

Funcionalidade que gerencia as movimentações financeiras diárias,

registrando entradas e saídas, rastreando transações e permitindo a geração de relatórios financeiros.

a. Registrar Entrada de Caixa

Descrição:

Permite o registro de todas as entradas financeiras, provenientes de:

- **Movimentações**

Campos obrigatórios:

- **Valor recebido.**
- **Data e hora.**
- **Descrição da entrada.**
- **Identificação do id parcial da venda, se a entrada for uma venda**
- **Tipo**

Permissões:

- **Administrador:** Efetua e revisa os registros de entradas.

b. Registrar Saída de Caixa

Descrição:

Permite o registro de todas as entradas financeiras, provenientes de:

- **Movimentações**

Campos obrigatórios:

- **Valor retirado.**
- **Data e hora.**
- **Descrição da saída.**
- **Tipo**

Permissões:

- **Administrador:** Efetua e revisa os registros de entradas.

c. Visualizar Movimentações do Caixa

Descrição:

Exibe uma tabela consolidada de todas as movimentações (entradas e saídas) com paginação:

- **Dados exibidos:** Data/Hora, tipo, descrição e valor

Permissões:

- **Administrador:** Acesso completo à visualização e análise.
- **Vendedor:** Não é autorizado.

d. Fechamento Diário do Caixa**Descrição:**

Permite o fechamento do caixa a qualquer momento, a abertura não é possível.

Permissões:

- **Administrador:** Realiza o fechamento diário e tem acesso ao relatório.
- **Vendedor:** Não é autorizado.

8. Nome: Relatórios**Descrição:**

Funcionalidade que gera e disponibiliza diversos relatórios para análise de desempenho comercial, financeiro e operacional, permitindo uma visão estratégica do negócio.

a. Relatório de Vendas dos produtos mais vendidos**Descrição:**

Exibe tabela e gráfico dos produtos mais vendidos com base em um período:

- **15 produtos com mais variações vendidas da loja**

Permissões:

- **Administrador:** Exclusivo.
- **Vendedor:** Não é autorizado.

b. Relatório de Desempenho de Vendedores

Descrição:

Apresenta performance dos vendedores, incluindo:

- **Total de vendas por vendedor.**
- **Ranking ordenado dos melhores vendedores.**

Permissões:

- **Administrador:** Exclusivo.
- **Vendedor:** Não é autorizado.

c. Relatório Financeiro

Descrição:

Consolida os dados do Controle de Caixa, apresentando:

- **Resumo das entradas e saídas.**
- **Saldo final do período.**

Permissões:

- **Administrador:** Exclusivo.
- **Vendedor:** Não é autorizado.

d. Relatório de vendas por forma de pagamento

Descrição:

Permite ao administrador visualizar o total de vendas em determinados períodos com segregação pela forma de pagamento

- **Crédito**
- **Débito**
- **Pix**
- **Dinheiro físico**

Permissões:

- **Administrador:** Exclusivo.
- **Vendedor:** Não é autorizado.

e. Relatório de baixo estoque

Descrição:

Retorna todos os produtos com estoque inferior ao número inserido na entrada.

Permissões:

- **Administrador:** Exclusivo.
- **Vendedor:** Não é autorizado.

f. Exportação e Compartilhamento de Relatórios**Descrição:**

Oferece a possibilidade de exportar os relatórios gerados para formatos como:

- **PDF.**

Permissões:

- **Administrador:** Exclusivo.
- **Vendedor:** Não é autorizado.

9. Requisitos não funcionais**Acessibilidade****Descrição:**

O sistema deve ser estilizado com um padrão escuro e de alto contraste, garantindo melhor visibilidade para usuários com deficiência visual. Esse padrão também deve ser aplicado na exportação de relatórios.

CrITÉrios de Aceitação:

- Todos os elementos da interface seguem o padrão de alto contraste.

Formatação de Senhas**Descrição:**

O sistema deve impor regras para a criação de senhas, garantindo que tenham entre 8 e 16 caracteres e incluam pelo menos uma letra maiúscula, uma letra minúscula, um número e um caractere especial.

CrITÉrios de Aceitação:

- O sistema exibe um erro ao tentar cadastrar uma senha fora das regras estabelecidas.
- O usuário só pode prosseguir com o cadastro se a senha atender a todas as regras.
- A regra de formatação da senha é informada ao usuário antes da criação da conta e caso o usuário não insira os caracteres corretamente.

Segurança (Criptografia de Senhas)

Descrição:

As senhas dos usuários devem ser criptografadas utilizando o algoritmo Bcrypt antes de serem armazenadas no banco de dados.

CrITÉrios de Aceitação:

- As senhas nunca são armazenadas em formato de texto puro.
- O sistema utiliza Bcrypt com um fator de custo adequado para garantir segurança.
- O processo de login compara a senha inserida com o hash armazenado.

10. Perfil dos Usuários

11.1. Administrador(Gerente contratador do serviço)

- **Descrição:** Usuário responsável pelo gerenciamento completo do sistema, tendo acesso total às funcionalidades.
- **Tipo:** Gerente ou proprietário do comércio.
- **Responsabilidade:**
 - Cadastrar, visualizar, editar e excluir produtos.
 - Gerenciar funcionários, incluindo cadastro, leitura, atualização e exclusão.
 - Cadastrar, visualizar, editar e excluir vendas e monitorar o desempenho da equipe.
 - Controlar o gerenciamento de caixa e de relatórios.

11.2. Vendedor

- **Descrição:** Funcionário responsável por realizar vendas e auxiliar no controle de estoque.
- **Tipo:** Funcionário com permissões limitadas.
- **Responsabilidade:**
 - Registrar vendas no sistema.
 - Consultar produtos disponíveis para informar clientes.
 - Acompanhar seu próprio desempenho de vendas.

11. Arquitetura:

O projeto será desenvolvido seguindo a arquitetura MVC (Model-View-Controller), garantindo uma separação clara entre as responsabilidades de apresentação, lógica de negócio e manipulação de dados. Essa abordagem facilita a manutenção, a escalabilidade e a organização do código.

Além disso, o projeto será construído utilizando um paradigma orientado a objetos (POO), garantindo reutilização de código, modularidade e encapsulamento. As classes serão projetadas para representar entidades e suas interações de forma estruturada.

A implementação será feita em uma linguagem tipada, garantindo maior segurança e previsibilidade no desenvolvimento, reduzindo erros relacionados a tipos de dados. Isso também facilita a detecção de inconsistências durante a fase de compilação ou interpretação.

12. Classes:

a. Modelos (Model)

■ Arquivo: Produto.ts

- **Classe:** Produto

- **Responsabilidade:** Representar um produto com atributos de nome, categoria, material, referência, variações, cada variação tendo sua descrição curta, quantidade e valor.

■ Arquivo: **Funcionario.ts**

- **Classe:** Funcionario
- **Responsabilidade:** Representar um funcionário com dados como nome, cargo, telefone, CPF, e outros. Relacionamento com vendas realizadas (caso o vendedor esteja associado à venda).

■ Arquivo: **Venda.ts**

- **Classe:** Venda
- **Responsabilidade:** Representar uma venda, incluindo produtos vendidos, valor total, forma de pagamento, vendedor responsável e a data da transação.

■ Arquivo: **Usuario.ts**

- **Classe:** Usuario
- **Responsabilidade:** Representar o usuário, incluindo nome completo, senha criptografada, e-mail de recuperação de senha, telefone e data de nascimento.

■ Arquivo: **Caixa.ts**

- **Classe:** Caixa
- **Responsabilidade:** Representar o caixa, incluindo vendas associadas a ele, movimentações financeiras e datas.

■ Arquivo: **Movimentacao.ts**

- **Classe:** Movimentação
- **Responsabilidade:** Representar movimentações financeiras dos caixas.

b. Controladores (Controller)

■ **Arquivo: ProdutoController.ts**

- **Classe:** ProdutoController
- **Responsabilidade:** Gerenciar as operações de CRUD para os produtos (criar, listar, atualizar, excluir produtos).

■ **Arquivo: FuncionarioController.ts**

- **Classe:** FuncionarioController
- **Responsabilidade:** Gerenciar as operações de CRUD para os funcionários (criar, listar, atualizar, excluir funcionários).

■ **Arquivo: VendaController.ts**

- **Classe:** VendaController
- **Responsabilidade:** Gerenciar as operações de CRUD para as vendas (registrar, listar, visualizar detalhes, atualizar e excluir vendas).

■ **Arquivo: RelatorioController.ts**

- **Classe:** RelatorioController
- **Responsabilidade:** Gerenciar as operações de relacionadas aos relatórios.

■ **Arquivo: CaixaController.ts**

- **Classe:** CaixaController
- **Responsabilidade:** Gerenciar as operações básicas para o caixa como adicionar venda em um caixa, visualizar vendas de um caixa específico, ver movimentações de um caixa específico e adicionar movimentações.

c. Rotas (Routes)

■ **Arquivo: produtoRoutes.js**

- **Classe:** ProdutoRoutes
 - **Responsabilidade:** Define as rotas que estarão disponíveis para interação com o modelo Produto (e.g., /api/produtos para criar, listar, editar e excluir produtos).
- **Arquivo: funcionarioRoutes.js**
 - **Classe:** FuncionarioRoutes
 - **Responsabilidade:** Define as rotas para manipulação de dados de funcionários (e.g., /api/funcionarios para criar, listar, editar e excluir funcionários).
- **Arquivo: vendaRoutes.js**
 - **Classe:** VendaRoutes
 - **Responsabilidade:** Define as rotas para operações relacionadas às vendas (e.g., /api/vendas para registrar, listar e atualizar vendas).
- **Arquivo: relatorioRoutes.js**
 - **Classe:** relatorioRoutes
 - **Responsabilidade:** Define as rotas para operações relacionadas aos relatórios.
- **Arquivo: caixaRoutes.js**
 - **Classe:** caixaRoutes
 - **Responsabilidade:** Define as rotas para operações relacionadas aos caixas (e.g., /api/caixa para registrar, listar e atualizar caixa).

14. Tecnologias Utilizadas

O vl-store será desenvolvido utilizando as seguintes tecnologias:

- **Backend:** Node.js com Express.js para criação da API REST + Typescript. Node.js oferece uma formulação em typescript no back-end, enquanto Express.js facilita a criação de APIs

RESTful. Typescript adiciona tipagem estática ao JavaScript, melhorando a segurança e a manutenção do código.

- **Banco de Dados:** MongoDB para armazenamento das informações: MongoDB é um banco de dados NoSQL que armazena dados em formato JSON, oferecendo flexibilidade e escalabilidade. Ideal para armazenar dados dinâmicos como informações de lojas, produtos e compradores.
- **Frontend:** React.js e Node.js para a interface do usuário além do Bootstrap para auxiliar na responsividade em diferentes telas: React.js permite criar interfaces de usuário interativas e eficientes enquanto Bootstrap facilita a construção de layouts responsivos para diferentes dispositivos.
- **Autenticação:** JWT (JSON Web Token) para controle de acesso: JWT é utilizado para autenticar e autorizar usuários de maneira segura e eficiente, permitindo a comunicação entre o front-end e back-end sem a necessidade de sessões tradicionais.
- **Segurança:** Bcrypt para hash de senhas e validações no backend: Bcrypt é utilizado para proteger senhas armazenadas no banco de dados, garantindo maior segurança com hash e salt. Isso evita que senhas em texto claro sejam comprometidas em caso de ataque.
- **Versionamento:** GitHub: O GitHub será utilizado para versionar o código-fonte do projeto, facilitando a colaboração, o controle de alterações e a integração contínua entre os membros da equipe.