Projeto Integrador 2º Semestre - DSM

# Disciplinas:

Banco de Dados Desenvolvimento Web II Engenharia de Software II

**Professores:** Bruno Nilton Orlando

**Grupo(n) / Nome da Empresa: Sistema:**

|  |  |
| --- | --- |
| Integrante | **Papel Principal** |
| Felipe Rafael Rocha | Back - end |
| Jamila Moraes Cardoso | PO, Analise de requisitos, analista de dados |
| Jefferson Felipe de Moura | Full-stack |
| Rafaela Cristina Leme de Morais | Front- end, Analise de requisitos |

**Fatec Araras 2025**

FICHA DE CONTROLE - PROJETO INTERDISCIPLINAR

TÍTULO DO PROJETO: título

DATA DA APRESENTAÇÃO: 08/07/2025 NOTA:

SEMESTRE: 2 semestre 2.2025

GRUPO: MorganDevs

DISCIPLINA CHAVE: Engenharia de Software II - PI II

PROFESSOR: Bruno Henrique de Paula Ferreira

INTEGRANTES DO GRUPO: Nome grupo

|  |  |
| --- | --- |
| Nome | Nota Individual |
| Felipe Rafael Rocha |  |
| Jamila Moraes Cardoso |  |
| Jefferson Felipe de Moura |  |
| Rafaela Cristina Leme de Morais |  |

Araras, 08 de julho de 2025

Professor Bruno Henrique de Paula Ferreira

Sumário

Sumário

[Disciplinas: 1](#_Toc202226605)

[1. Apresentação da Empresa 5](#_Toc202226606)

[ 5](#_Toc202226607)

[A MorganDevs surgiu a partir do Projeto Integrador do curso de Desenvolvimento de Sistemas Multiplataformas da Fatec Araras, formado pelos alunos do 2 semestre, com objetivo de principal de colocar em prática os conhecimentos adquiridos durante o semestre. 5](#_Toc202226608)

[1.1 Missão 5](#_Toc202226609)

[1.2 Visão 6](#_Toc202226610)

[1.3 Valores 6](#_Toc202226611)

[1.4 Link Repositório 6](#_Toc202226612)

[O projeto está disponibilizado no repositório do Github : 6](#_Toc202226613)

[1.5 Link Trello 6](#_Toc202226614)

[O cronograma foi feito pelo site do trello 6](#_Toc202226615)

[https://trello.com/invite/b/67be6b60a2251556f2c94776/ATTIc2d5523ff0fab36a1d3be7a3c511d993CB548127/20251-pi-2 6](#_Toc202226616)

[2. Escopo do sistema 7](#_Toc202226617)

[2.1 Objetivos do projeto 7](#_Toc202226618)

[2.2 Técnica de levantamento de requisitos 7](#_Toc202226619)

[2.3 Requisitos funcionais 7](#_Toc202226620)

[2.4 Requisitos não funcionais 9](#_Toc202226621)

[2.5 Comparativo entre sistemas 10](#_Toc202226622)

[2.6 Cronograma 10](#_Toc202226623)

[3. Documentação do Sistema 12](#_Toc202226624)

[3.1 Metodologia de Desenvolvimento 12](#_Toc202226625)

[**Caso de Uso: CRUDCliente** 12](#_Toc202226626)

[**Caso de Uso: PedirOrcamento** 13](#_Toc202226627)

[**Caso de Uso: FazerCompra** 13](#_Toc202226628)

[**Caso de Uso: RealizarPagamento *(include)*** 13](#_Toc202226629)

[**Caso de Uso: CRUDFornecedor** 14](#_Toc202226630)

[**Caso de Uso: CRUDUsuario** 16](#_Toc202226631)

[**Caso de Uso: CRUDOperacao** 16](#_Toc202226632)

[**Caso de Uso: AvaliaCotacao** 17](#_Toc202226633)

[**Caso de Uso: RealizarCompra** 17](#_Toc202226634)

[**Caso de Uso: RealizarVenda** 18](#_Toc202226635)

[**Caso de Uso: Pagamento *(include)*** 18](#_Toc202226636)

[**Caso de Uso: CRUDTransportadora** 18](#_Toc202226637)

[**Caso de Uso: CRUDCategoria** 18](#_Toc202226638)

[**Caso de Uso: CRUDProdutos** 19](#_Toc202226639)

[3.7 Etapas / Sprints realizados (1º e 2º) 27](#_Toc202226640)

[3.8 Interface do usuário 27](#_Toc202226641)

[4. Testes e Qualidade 28](#_Toc202226642)

[4.1 Estratégia de Testes: 28](#_Toc202226643)

[4.2 Resultados dos Testes: 28](#_Toc202226644)

[4.4 Requisitos mínimos de hardware e software para o sistema 28](#_Toc202226645)

[4.5 Contrato para desenvolvimento de software 29](#_Toc202226646)

[5. Considerações Finais 29](#_Toc202226647)

[6. Referências 29](#_Toc202226648)

[Anexo I - Diário de bordo 29](#_Toc202226649)

[Anexo II – Cronograma efetivo Anexo III – Evidências 29](#_Toc202226650)

# Apresentação da Empresa



A MorganDevs surgiu a partir do Projeto Integrador do curso de Desenvolvimento de Sistemas Multiplataformas da Fatec Araras, formado pelos alunos do 2 semestre, com objetivo de principal de colocar em prática os conhecimentos adquiridos durante o semestre.



Empresa escolhida para desenvolvimento é a MVS de Rio Claro, stakeholder é o Marcelo Vieira Silva. Fundada em Março de 2010, a MVS Info sempre está em busca de soluções tecnológicas e produtos de qualidade para atender a demanda de seus clientes. Visando atender essas necessidades construímos fortes parcerias comerciais onde o cliente se sinta à vontade para tratar do seu projeto com flexibilidade e alto poder de negociação.

## Missão

Desenvolver software de alta qualidade, combinando tecnologia e criatividade para entregar soluções eficientes, escaláveis ​​e alinhadas às necessidades dos clientes, promovendo inovação e crescimento sustentável.

## Visão

Ser referência no desenvolvimento de software inovador, criando soluções tecnológicas que impulsionam negócios, otimizam processos e geram valor para empresas e sociedade sustentável.

## Valores

**Inovação Contínua.**

**Excelência e Qualidade.**

**Transparência e Ética.**

## Link Repositório

O projeto está disponibilizado no repositório do Github :

https://github.com/jmcardoso18/2025.1-PI-2-Semestre

## Link Trello

O cronograma foi feito pelo site do trello

https://trello.com/invite/b/67be6b60a2251556f2c94776/ATTIc2d5523ff0fab36a1d3be7a3c511d993CB548127/20251-pi-2

# Escopo do sistema

Este documento descreve as especificações de projeto de uma aplicação web, em formato B2B, que intermedia compra e venda de produtos entre empresas pela MVS Info.

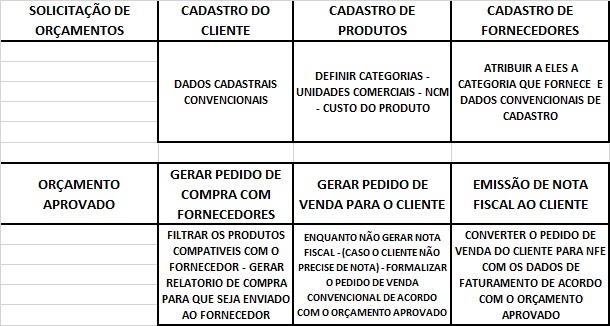
## Objetivos do projeto

O objetivo deste projeto é desenvolver uma aplicação web para o cliente MVS Info, trazendo uma solução para automatizar a capitação das cotações para o pedido de orçamendo de seus clientes e um resumo do seu fluxo de caixa.

## Técnica de levantamento de requisitos

O levantamento de requisitos foi feito através de uma videochamada com o proprietário da empresa, MVS Info, Marcelo, nesta reunião foi apresentado o aplicativo que ele sub utiliza, mas que acaba não usando todas as funções oferecidas. Ele também explicou como é seu processo de negócio e como gostaria de automatizar sua relação de orçamento com os clientes, cotação com fornecedores , a compra e revenda dos produtos, visto que sua empresa não trabalha com estoque.

O cliente nós enviou uma resumo do que gostaria que o sistema tivesse:



Obs: A emissão de nota fiscal ficou para um projeto futuro visto que para gerar é necessário o uso de uma API, que não é pré requisito deste semestre.

## Requisitos funcionais

|  |  |  |
| --- | --- | --- |
| **Cliente** |  |  |
|  |  |  |
| **RF01** | CRUD Cliente | Importante |
| O sistema deve permitir que o cliente crie, edite, visualize e exclua seu cadastro (CRUDCliente). | | |
|  |  |  |
| **RF02** | Login Cliente | Essencial |
| O sistema deve permitir que o cliente faça login e acesse suas funcionalidades. | | |
|  |  |  |
| **RF03** | Solicitação de orçamento | Essencial |
| O cliente poderá solicitar um orçamento selecionando produtos e quantidades. | | |
|  |  |  |
| **RF04** | Cliente aprova orçamento | Essencial |
| O sistema deve permitir que o cliente aprove ou rejeite orçamentos recebidos. | | |
|  |  |  |
| **RF05** | Cliente realiza a compra | Essencial |
| O sistema deve permitir que o cliente realize a compra de produtos após aprovação do orçamento. | | |
|  |  |  |
| **RF06** | Cliente realiza o pagamento | Essencial |
| O sistema deve permitir que o cliente realize o pagamento da compra. | | |
|  |  |  |
| **RF07** | Cliente acompanha o status do pedido | Importante |
| O sistema deve permitir que o cliente acompanhe o status do pedido. | | |
|  |  |  |
| **Fornecedor** | |  |
|  |  |  |
| **RF08** | CRUD fornecedor | Importante |
| O sistema deve permitir que o fornecedor crie, edite, visualize e exclua seu cadastro (CRUDFornecedor). | | |
|  |  |  |
| **RF09** | Login fornecedor | Essencial |
| O fornecedor deve fazer login e acessar suas funcionalidades. | | |
|  |  |  |
| **RF10** | Visualizar cotação conforme sua categoria | Essencial |
| O fornecedor deve visualizar solicitações de orçamento e preencher a cotação com valores unitários. | | |
|  |  |  |
| **RF11** | Visualizar pedido | Essencial |
| O fornecedor deve visualizar pedidos gerados. | | |
|  |  |  |
| **Administrativo (MVS)** | |  |
|  |  |  |
| **RF12** | Administrador com login pré definido | Essencial |
| O administrador deve ter login pré-cadastrado. | | |
|  |  |  |
| **RF13** | Gerenciar todos as entidades do sistema | Essencial |
| O sistema deve permitir que o administrador gerencie todas as tabelas do sistema | | |
|  |  |  |
| **RF14** | Administrador gerenciando as operações do sistema | Essencial |
| O sistema deve permitir que o administrador visualize e gerencie orçamentos, compras, vendas e pedidos. | | |
|  |  |  |
| **RF15** | Gerar relatórios | Importante |
| O sistema deve gerar relatórios de vendas e orçamentos com os devidos campos: valor de compra, imposto, lucro e total. | | |

## Requisitos não funcionais

|  |  |  |
| --- | --- | --- |
| **RNF01** |  | Versão 1 |
| O sistema deve ser acessível via navegador com interface responsiva. | | |
|  |  |  |
| **RNF02** |  | Versão 1 |
| O sistema deve enviar e-mails automaticamente aos fornecedores ao solicitar orçamento. | | |
|  |  |  |
| **RNF03** |  | Versão 1 |
| O sistema deve utilizar autenticação com login e senha criptografada. | | |
|  |  |  |
| **RNF04** |  | Versão 1 |
| O sistema deve apresentar mensagens amigáveis em caso de erro. | | |
|  |  |  |
| **RNF05** |  | Versão 1 |
| O tempo de resposta para ações de CRUD e login deve ser inferior a 10 segundos. | | |
|  |  |  |

## Regras de négocio

|  |  |
| --- | --- |
| **RN01** | Apenas usuários com login válido e ativo podem acessar o sistema. |
|  |  | |  |  |  |  |  |
| **RN02** | Clientes e fornecedores devem possuir CNPJ válidos e únicos. |
|  |  | |  |  |  |  |  |
| **RN03** | Um orçamento só poderá ser convertido em pedido se for aprovado pelo cliente e por um operador da MVS. |
|  |  | |  |  |  |  |  |
| **RN04** | O pagamento deve ser confirmado antes de gerar o pedido para o fornecedor. |
|  |  | |  |  |  |  |  |
| **RN05** | O valor total da venda deve ser calculado com base nos valores unitários da cotação, impostos e lucro. |
|  |  | |  |  |  |  |  |
| **RN06** | Um fornecedor só pode visualizar e responder orçamentos relacionados às categorias em que está cadastrado. |
|  |  | |  |  |  |  |  |
| **RN07** | O sistema deve impedir alteração de dados do cliente ou fornecedor se houver pedido em andamento. |
|  |  | |  |  |  |  |  |
| **RN08** | A atualização de senha deve refletir nas tabelas cliente e login. |

## Comparativo entre sistemas

O sistema **MVS** é uma plataforma que intermedia a relação entre clientes e fornecedores. Permite ao cliente realizar cadastros, solicitar orçamentos, efetuar compras e pagamentos, enquanto os fornecedores respondem com cotações e processam vendas. O administrador (MVS) gerencia usuários, transportadoras, produtos e acompanha todas as transações do sistema. O sistema possui controle de login por tipo de usuário, emissão de relatórios e integração entre módulos via casos de uso relacionados.

| **Funcionalidade** | **Sistema MVS (Desenvolvido)** | **Mercado Livre** | **B2Brazil** |
| --- | --- | --- | --- |
| Cadastro de Cliente e Fornecedor | Sim | Sim | Sim |
| Solicitação de Orçamento | Sim (cliente → fornecedor) | Não | Sim |
| Resposta com Cotação | Sim | Não | Sim |
| Compra Direta | Sim (após orçamento aprovado) | Sim | Não direto |
| Pagamento Integrado | Não | Sim | Não (externo) |
| Gestão de Produtos e Categorias | Sim (administrador) | Sim | Sim |
| Acompanhamento de Pedidos | Sim | Sim | Limitado |
| Relatórios de Vendas e Orçamentos | Sim | Sim (vendas) | Não |
| Controle de Transportadoras | Sim | Limitado | Não |
| Interface Multiusuário | Sim (cliente, fornecedor, admin) | Sim | Sim |

**Tabela 1 – Comparativo das funcionalidades da aplicação**

## Cronograma

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tarefas** | **18/02 a**  **25/02** | **25/02 a**  **04/03** | **25/02 a**  **18/03** | **04/03 a**  **25/03** | **25/03 a**  **15/04** | **22/04 a**  **20/05** | **27/05**  **01/07** | **08/07** |
| Definição Grupos | X |  |  |  |  |  |  |  |
| Criação Marca Empresa |  | X |  |  |  |  |  |  |
| Definição Empresa |  |  | X |  |  |  |  |  |
| Escopo Sistema |  |  | X | X |  |  |  |  |
| Requisitos |  |  | X | X | X | X |  |  |
| Diagramas |  |  |  | X | X | X |  |  |
| Protótipo |  |  |  | X | X | X | X |  |
| Documentação |  | X | X | X | X | X | X |  |
| Desenvolvimento |  |  |  | X | X | X | X |  |
| Entrega |  |  |  |  |  |  | X |  |
| Apresentação |  |  |  |  |  |  |  | X |

# Documentação do Sistema

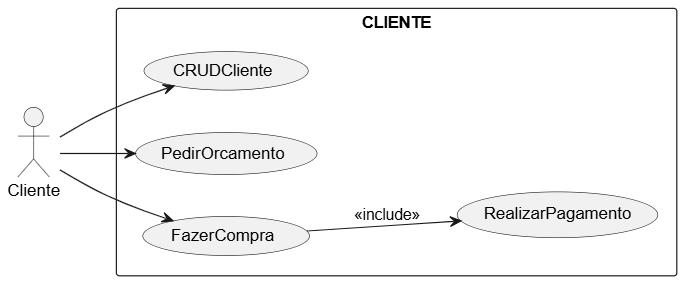
Este capítulo reúne os documentos técnicos que descrevem os principais aspectos do sistema desenvolvido, servindo como referência para sua compreensão, manutenção e evolução. A documentação desempenha um papel fundamental no ciclo de vida do software, pois registra de forma estruturada as decisões de projeto, a arquitetura adotada, as funcionalidades implementadas e demais características relevantes do sistema. Assim, ela garante a continuidade do desenvolvimento, facilita a colaboração entre equipes e contribui para a qualidade do produto final.

## Metodologia de Desenvolvimento

No desenvolvimento deste sistema, foi adotada a metodologia **ágil Scrum**, visando maior flexibilidade, colaboração e entrega contínua de valor. O projeto foi dividido em **sprints quinzenais**, com planejamento, execução e revisão ao final de cada ciclo. As tarefas foram organizadas em um quadro visual (Kanban), o que permitiu o acompanhamento do progresso em tempo real. Essa abordagem facilitou a adaptação às mudanças de requisitos, promoveu maior engajamento da equipe e garantiu entregas incrementais e testáveis ao longo do processo.

* 1. **Diagramas UML**

## 3.3.1 Diagrama de caso de uso (1º)

****

**Caso de Uso: CRUDCliente**

**Atores:** Cliente

**Descrição:** Permite ao cliente criar, visualizar, editar e excluir seu próprio cadastro no sistema.

**Fluxo Principal:**

1. O cliente acessa a interface de cadastro.
2. O sistema solicita os dados pessoais e de login.
3. O cliente preenche ou edita as informações.
4. O sistema valida e salva os dados.

**Fluxo Alternativo:**

Se o cliente tentar cadastrar um CPF/CNPJ já existente, o sistema exibirá uma mensagem de erro.

**Caso de Uso: PedirOrcamento**

**Atores:** Cliente

**Descrição:** Permite que o cliente selecione produtos e quantidades desejadas para gerar uma solicitação de orçamento.

**Fluxo Principal:**

1. O cliente acessa a funcionalidade de orçamento.
2. Seleciona os produtos e define as quantidades.
3. Confirma a solicitação.
4. O sistema registra o pedido e notifica os fornecedores relacionados.

**Caso de Uso: FazerCompra**

**Atores:** Cliente

**Descrição:** Após a aprovação de um orçamento, o cliente realiza a compra dos produtos selecionados.

**Fluxo Principal:**

1. O cliente visualiza os orçamentos aprovados.
2. Seleciona aquele que deseja comprar.
3. O sistema direciona para o processo de pagamento (include: RealizarPagamento).
4. A compra é registrada e seu status atualizado.

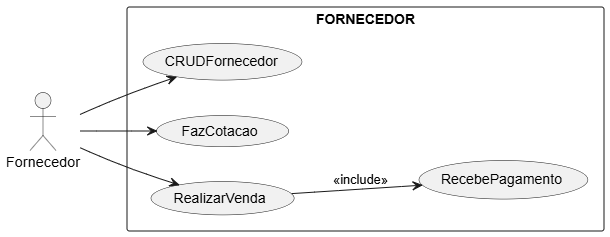
**Caso de Uso: RealizarPagamento *(include)***

**Atores:** Cliente

**Descrição:** Caso incluído automaticamente durante a finalização da compra. Trata do processo de pagamento.

**Fluxo Principal:**

1. O sistema apresenta as opções de pagamento ao cliente.
2. O cliente escolhe o método e insere os dados.
3. O sistema confirma o pagamento e gera um comprovante.

****

**Caso de Uso: CRUDFornecedor**

**Atores:** Fornecedor

**Descrição:** Permite ao fornecedor realizar o cadastro, edição, visualização e exclusão de seus dados no sistema.

**Fluxo Principal:**

1. O fornecedor acessa a interface de perfil.
2. Visualiza seus dados e escolhe entre atualizar ou excluir.
3. Ao cadastrar-se, preenche CNPJ, contato, categoria e login.
4. O sistema valida e armazena as informações.

**Caso de Uso: FazCotacao**

**Atores:** Fornecedor

**Descrição:** Permite que o fornecedor visualize solicitações de orçamento e preencha os valores unitários dos produtos.

**Fluxo Principal:**

1. O fornecedor acessa a lista de solicitações pendentes.
2. Visualiza os produtos e quantidades solicitadas.
3. Preenche os campos com valores unitários.
4. Confirma a cotação.
5. O sistema salva os dados e envia para avaliação.

**Caso de Uso: RealizarVenda**

**Atores:** Fornecedor

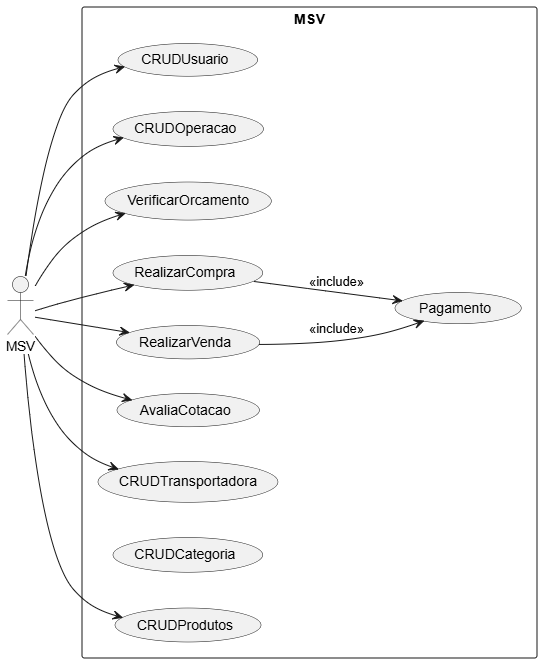
**Descrição:** Após a aprovação do orçamento, o fornecedor registra a venda no sistema.

**Fluxo Principal:**

1. O fornecedor acessa os orçamentos aprovados.
2. Seleciona a cotação escolhida.
3. Registra a venda vinculada ao orçamento aprovado.
4. O sistema direciona automaticamente para o recebimento do pagamento (include: RecebePagamento).

**Caso de Uso: RecebePagamento *(include)***

1. **Atores:** Fornecedor
2. **Descrição:** Caso incluído automaticamente no processo de venda, representa o recebimento de pagamento por parte do cliente.
3. **Fluxo Principal:**
4. O sistema apresenta os dados do pagamento (valor, data, método).
5. O fornecedor confirma o recebimento.
6. O status da venda é atualizado como concluído.

****

**Caso de Uso: CRUDUsuario**

**Atores:** MVS (Administrador)

**Descrição:** Permite ao administrador realizar a criação, consulta, atualização e exclusão de registros de usuários (clientes, fornecedores, transportadoras).

**Fluxo Principal:**

1. O administrador acessa o menu de usuários.
2. Visualiza a lista atual e escolhe uma ação (novo, editar, excluir).
3. O sistema executa a operação e retorna uma confirmação.

**Caso de Uso: CRUDOperacao**

**Atores:** MVS

**Descrição:** Permite ao administrador gerenciar informações gerais sobre operações realizadas no sistema.

**Fluxo Principal:**

1. O administrador acessa a tela de operações.
2. Visualiza, atualiza dados como tipo, status ou datas.
3. O sistema atualiza os registros conforme as alterações.

**Caso de Uso: VerificarOrcamento**

**Atores:** MVS

**Descrição:** Permite ao administrador visualizar todos os orçamentos enviados por clientes e avaliá-los.

**Fluxo Principal:**

1. O administrador acessa a lista de orçamentos.
2. Seleciona um orçamento para análise.
3. O sistema exibe todos os dados relacionados (cliente, produtos, cotações).
4. O administrador aprova ou rejeita.

**Caso de Uso: AvaliaCotacao**

**Atores:** MVS

**Descrição:** Permite que o administrador compare cotações dos fornecedores e escolha a mais vantajosa.

**Fluxo Principal:**

1. O administrador acessa a cotação de um orçamento.
2. Visualiza preços de diferentes fornecedores.
3. Seleciona a cotação a ser aprovada.
4. O sistema registra a decisão e prepara o processo de compra.

**Caso de Uso: RealizarCompra**

**Atores:** MVS

**Descrição:** Após aprovação de orçamento e cotação, o administrador finaliza a compra dos produtos.

**Fluxo Principal:**

1. O administrador inicia o processo de compra.
2. Escolhe fornecedor, produto e quantidade.
3. O sistema inclui automaticamente o caso de uso **Pagamento**.
4. Compra é registrada no sistema.

**Caso de Uso: RealizarVenda**

**Atores:** MVS

**Descrição:** Permite ao administrador registrar uma venda aprovada ao cliente.

**Fluxo Principal:**

1. O administrador acessa a lista de vendas aprovadas.
2. Confirma a operação.
3. O sistema registra a venda e aciona o caso de uso **Pagamento**.

**Caso de Uso: Pagamento *(include)***

**Atores:** MVS

**Descrição:** Representa a finalização de qualquer transação comercial com o respectivo pagamento.

**Fluxo Principal:**

1. O sistema solicita dados do pagamento (valor, forma, data).
2. O administrador registra o pagamento.
3. O sistema atualiza o status da compra/venda.

**Caso de Uso: CRUDTransportadora**

**Atores:** MVS

**Descrição:** Permite ao administrador gerenciar os dados das transportadoras.

**Fluxo Principal:**

1. O administrador acessa o módulo de transportadoras.
2. Adiciona, edita ou remove registros.
3. O sistema confirma e exibe a alteração.

**Caso de Uso: CRUDCategoria**

**Atores:** MVS

**Descrição:** Permite o gerenciamento das categorias de produtos associadas aos fornecedores.

**Fluxo Principal:**

1. O administrador acessa o menu de categorias.
2. Realiza as operações desejadas (criar, editar, remover).
3. O sistema atualiza os dados conforme solicitado.

**Caso de Uso: CRUDProdutos**

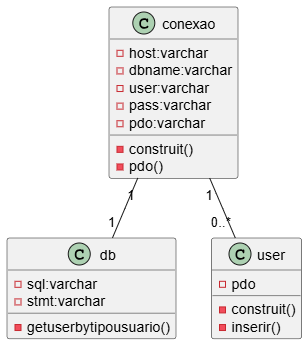
**Atores:** MVS

**Descrição:** Permite ao administrador gerenciar o catálogo de produtos.

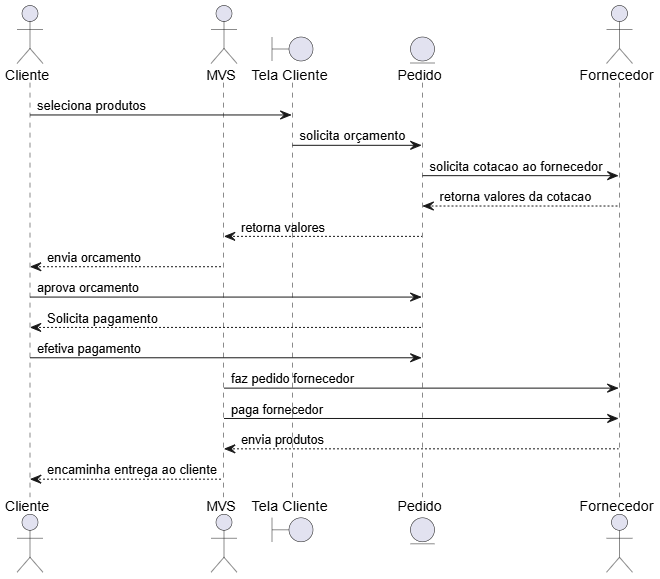
**Fluxo Principal:**

1. O administrador acessa a lista de produtos.
2. Adiciona um novo item ou edita/exclui existentes.
3. O sistema salva e disponibiliza os produtos atualizados.

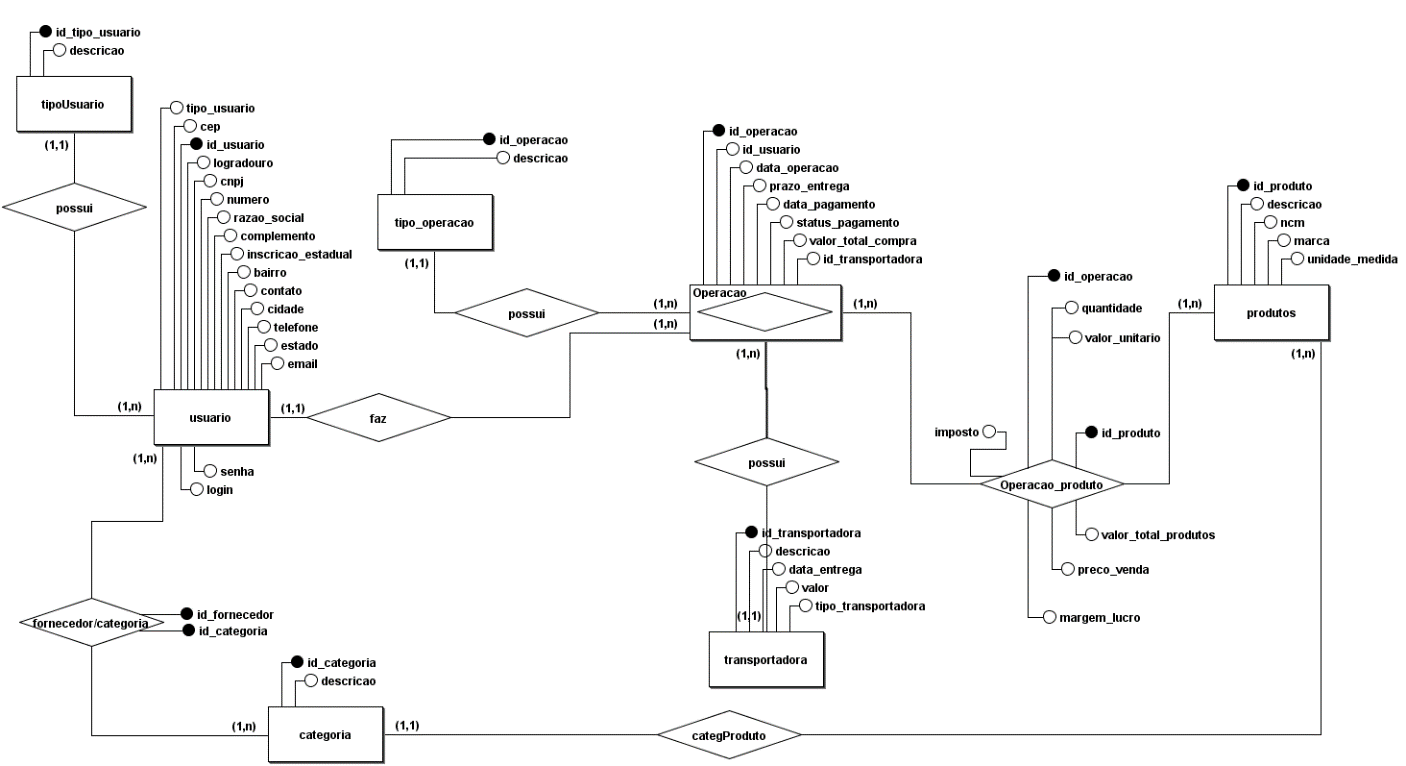
## 3.3.2 Diagrama de classes (2º)

****

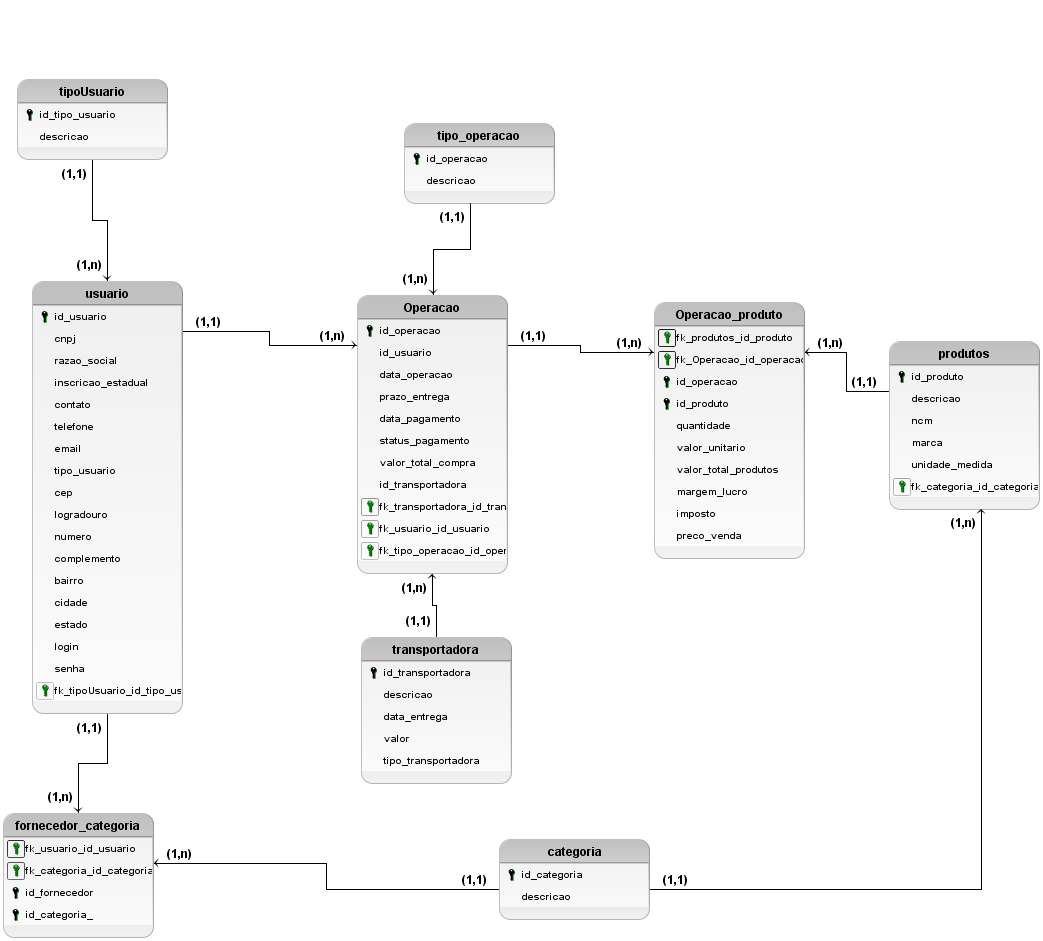
## 3.3.3 Diagrama de sequência (2º)

****

* 1. **Modelo Conceitual (1º)**

****

* 1. **Modelo lógico (1º)**

****

* 1. **Modelo Físico (2º)**

*-- Script de criação do banco de dados*

*-- Criando o banco*

**CREATE** **DATABASE** lojasistema;

*-- chamando o banco--*

**USE** lojasistema;

*-- Tabela: tipoUsuario*

**CREATE** **TABLE** tipoUsuario (

id\_tipo\_usuario **INT** **PRIMARY** **KEY** **AUTO\_INCREMENT**,

descricao **VARCHAR**(50) **NOT** **NULL**

);

*-- Tabela: usuario*

**CREATE** **TABLE** usuario (

id\_usuario **INT** **PRIMARY** **KEY** **AUTO\_INCREMENT**,

cnpj **VARCHAR**(18),

razao\_social **VARCHAR**(255),

inscricao\_estadual **VARCHAR**(20),

contato **VARCHAR**(100),

telefone **VARCHAR**(20),

email **VARCHAR**(100),

tipo\_usuario **INT**,

cep **VARCHAR**(10),

logradouro **VARCHAR**(100),

numero **INT**,

complemento **VARCHAR**(50),

bairro **VARCHAR**(50),

cidade **VARCHAR**(50),

estado **VARCHAR**(2),

login **VARCHAR**(50),

senha **VARCHAR**(100),

**FOREIGN** **KEY** (tipo\_usuario) **REFERENCES** tipoUsuario(id\_tipo\_usuario)

);

*-- Tabela: categoria*

**CREATE** **TABLE** categoria (

id\_categoria **INT** **PRIMARY** **KEY** **AUTO\_INCREMENT**,

descricao **VARCHAR**(100) **NOT** **NULL**

);

*-- Tabela: produtos*

**CREATE** **TABLE** produtos (

id\_produto **INT** **PRIMARY** **KEY** **AUTO\_INCREMENT**,

descricao **VARCHAR**(100),

ncm **VARCHAR**(20),

marca **VARCHAR**(50),

unidade\_medida **VARCHAR**(20),

fk\_categoria\_id\_categoria **INT**,

**FOREIGN** **KEY** (fk\_categoria\_id\_categoria) **REFERENCES** categoria(id\_categoria)

);

*-- Tabela: transportadora*

**CREATE** **TABLE** transportadora (

id\_transportadora **INT** **PRIMARY** **KEY** **AUTO\_INCREMENT**,

descricao **VARCHAR**(100),

data\_entrega **DATE**,

valor **DECIMAL**(10,2),

tipo\_transportadora **VARCHAR**(50)

);

*-- Tabela: tipo\_operacao*

**CREATE** **TABLE** tipo\_operacao (

id\_tipo\_operacao **INT** **PRIMARY** **KEY** **AUTO\_INCREMENT**,

descricao **VARCHAR**(100)

);

*-- Tabela: operacao*

**CREATE** **TABLE** operacao (

id\_operacao **INT** **PRIMARY** **KEY** **AUTO\_INCREMENT**,

fk\_usuario\_id\_usuario **INT**,

data\_operacao **DATE**,

prazo\_entrega **VARCHAR**(50),

data\_pagamento **DATE**,

status\_pagamento **VARCHAR**(50),

valor\_total\_compra **DECIMAL**(10,2),

fk\_transportadora\_id\_transportadora **INT**,

fk\_tipo\_operacao\_id\_tipo\_operacao **INT**,

**FOREIGN** **KEY** (fk\_usuario\_id\_usuario) **REFERENCES** usuario(id\_usuario),

**FOREIGN** **KEY** (fk\_transportadora\_id\_transportadora) **REFERENCES** transportadora(id\_transportadora),

**FOREIGN** **KEY** (fk\_tipo\_operacao\_id\_tipo\_operacao) **REFERENCES** tipo\_operacao(id\_tipo\_operacao)

);

*-- Tabela: fornecedor\_categoria*

**CREATE** **TABLE** fornecedor\_categoria (

id\_fornecedor **INT**,

id\_categoria **INT**,

**PRIMARY** **KEY** (id\_fornecedor, id\_categoria),

**FOREIGN** **KEY** (id\_fornecedor) **REFERENCES** usuario(id\_usuario),

**FOREIGN** **KEY** (id\_categoria) **REFERENCES** categoria(id\_categoria)

);

*-- Tabela: operacao\_produto*

**CREATE** **TABLE** operacao\_produto (

id\_operacao **INT**,

id\_produto **INT**,

quantidade **INT**,

valor\_unitario **DECIMAL**(10,2),

valor\_total\_produtos **DECIMAL**(10,2),

margem\_lucro **DECIMAL**(5,2),

imposto **DECIMAL**(5,2),

preco\_venda **DECIMAL**(10,2),

**PRIMARY** **KEY** (id\_operacao, id\_produto),

**FOREIGN** **KEY** (id\_operacao) **REFERENCES** operacao(id\_operacao),

**FOREIGN** **KEY** (id\_produto) **REFERENCES** produtos(id\_produto)

);

*-- Script de Inserção dos dados*

*-- Inserções: tipoUsuario*

**INSERT** **INTO** tipoUsuario (descricao) **VALUES**

('Cliente'),

('Fornecedor'),

('Administrador');

*-- Inserções: categoria*

**INSERT** **INTO** categoria (descricao) **VALUES**

('Informática'),

('Iluminação'),

('Cabos Elétricos'),

('Tubos e PEAD'),

('Pré-moldados');

*-- Inserções: tipo\_operacao*

**INSERT** **INTO** tipo\_operacao (descricao) **VALUES**

('Venda'),

('Compra'),

('Orçamento'),

('Cotação');

*-- Inserções: usuario*

**INSERT** **INTO** usuario (cnpj, razao\_social, inscricao\_estadual, contato, telefone, email, tipo\_usuario, cep, logradouro, numero, complemento, bairro, cidade, estado, login, senha) **VALUES**

('12.345.678/0001-01', 'Comercial Tech LTDA', '1234567890', 'João Lima', '(11)98765-4321', 'joao@comtech.com', 1, '01001-000', 'Rua Alfa', 123, 'Sala 5', 'Centro', 'São Paulo', 'SP', 'joaotech', 'senha123'),

('98.765.432/0001-99', 'Distribuidora Elétrica SA', '9876543210', 'Maria Souza', '(21)99876-5432', 'maria@disteletrica.com', 2, '20000-000', 'Av. Beta', 456, '', 'Bairro Industrial', 'Rio de Janeiro', 'RJ', 'mariasouza', 'segura321'),

('22.333.444/0001-55', 'Admin Serviços Ltda', '2233445566', 'Carlos Mendes', '(31)91234-5678', 'carlos@adminsrv.com', 3, '30000-000', 'Rua Gama', 789, 'Conj. C', 'Centro', 'Belo Horizonte', 'MG', 'carluxo', 'adm123');

*-- Inserções: produtos*

**INSERT** **INTO** produtos (descricao, ncm, marca, unidade\_medida, fk\_categoria\_id\_categoria) **VALUES**

('Notebook Acer Aspire 5', '84713012', 'Acer', 'un', 1),

('Lâmpada LED 9W', '85392900', 'Philips', 'un', 2),

('Cabo Flexível 4mm Azul', '85444900', 'Sil', 'm', 3),

('Tubo PEAD 60mm', '39172100', 'Tigre', 'm', 4),

('Bloco Estrutural 39x19x14', '68109100', 'Premol', 'un', 5);

*-- Inserções: transportadora*

**INSERT** **INTO** transportadora (descricao, data\_entrega, valor, tipo\_transportadora) **VALUES**

('TransVia Express', '2025-07-01', 120.00, 'Terrestre'),

('Carga Rápida LTDA', '2025-07-03', 75.50, 'Rodoviária'),

('Entrega Aérea Brasil', '2025-07-02', 250.00, 'Aérea');

*-- Inserções: operacao*

**INSERT** **INTO** operacao (fk\_usuario\_id\_usuario, data\_operacao, prazo\_entrega, data\_pagamento, status\_pagamento, valor\_total\_compra, fk\_transportadora\_id\_transportadora, fk\_tipo\_operacao\_id\_tipo\_operacao) **VALUES**

(1, '2025-06-20', '7 dias', '2025-06-27', 'Pago', 6400.00, 1, 1),

(2, '2025-06-22', '10 dias', **NULL**, 'Pendente', 3400.00, 2, 2),

(1, '2025-06-25', '5 dias', '2025-06-30', 'Pago', 1200.00, 3, 3);

*-- Inserções: fornecedor\_categoria*

**INSERT** **INTO** fornecedor\_categoria (id\_fornecedor, id\_categoria) **VALUES**

(2, 1),

(2, 3),

(2, 4);

*-- Inserções: operacao\_produto*

**INSERT** **INTO** operacao\_produto (id\_operacao, id\_produto, quantidade, valor\_unitario, valor\_total\_produtos, margem\_lucro, imposto, preco\_venda) **VALUES**

(1, 1, 2, 3200.00, 6400.00, 20.00, 15.00, 3680.00),

(2, 3, 500, 6.00, 3000.00, 25.00, 10.00, 7.50),

(2, 2, 40, 10.00, 400.00, 30.00, 5.00, 13.00),

(3, 5, 300, 4.00, 1200.00, 15.00, 8.00, 4.83);

*-- Procedure , trigger, viem e função --*

*-- PRocedure*

DELIMITER $$

**CREATE** **PROCEDURE** inserir\_usuario (

**IN** p\_cnpj **VARCHAR**(18),

**IN** p\_razao\_social **VARCHAR**(255),

**IN** p\_inscricao\_estadual **VARCHAR**(20),

**IN** p\_contato **VARCHAR**(100),

**IN** p\_telefone **VARCHAR**(20),

**IN** p\_email **VARCHAR**(100),

**IN** p\_tipo\_usuario **INT**,

**IN** p\_cep **VARCHAR**(10),

**IN** p\_logradouro **VARCHAR**(100),

**IN** p\_numero **INT**,

**IN** p\_complemento **VARCHAR**(50),

**IN** p\_bairro **VARCHAR**(50),

**IN** p\_cidade **VARCHAR**(50),

**IN** p\_estado **VARCHAR**(2),

**IN** p\_login **VARCHAR**(50),

**IN** p\_senha **VARCHAR**(100)

)

**BEGIN**

**INSERT** **INTO** usuario (

cnpj, razao\_social, inscricao\_estadual, contato, telefone,

email, tipo\_usuario, cep, logradouro, numero, complemento,

bairro, cidade, estado, login, senha

)

**VALUES** (

p\_cnpj, p\_razao\_social, p\_inscricao\_estadual, p\_contato, p\_telefone,

p\_email, p\_tipo\_usuario, p\_cep, p\_logradouro, p\_numero, p\_complemento,

p\_bairro, p\_cidade, p\_estado, p\_login, p\_senha

);

**END**$$

DELIMITER ;

*-- Trigger --*

DELIMITER $$

**CREATE** **TRIGGER** trg\_atualiza\_valor\_total\_operacao

**AFTER** **INSERT** **ON** operacao\_produto

**FOR** **EACH** **ROW**

**BEGIN**

**UPDATE** operacao

**SET** valor\_total\_compra = (

**SELECT** **SUM**(valor\_total\_produtos)

**FROM** operacao\_produto

**WHERE** id\_operacao = **NEW**.id\_operacao

)

**WHERE** id\_operacao = **NEW**.id\_operacao;

**END**$$

DELIMITER ;

*-- Função*

DELIMITER $$

**CREATE** **FUNCTION** resumo\_operacoes\_por\_usuario(

p\_id\_usuario **INT**,

p\_tipo\_operacao **VARCHAR**(50)

) **RETURNS** **VARCHAR**(255)

**DETERMINISTIC**

**BEGIN**

**DECLARE** tipo\_id **INT**;

**DECLARE** total\_operacoes **INT** **DEFAULT** 0;

**DECLARE** valor\_total **DECIMAL**(10,2) **DEFAULT** 0.00;

**DECLARE** resultado **VARCHAR**(255);

*-- Buscar o ID do tipo de operação*

**SELECT** id\_tipo\_operacao **INTO** tipo\_id

**FROM** tipo\_operacao

**WHERE** descricao = p\_tipo\_operacao

**LIMIT** 1;

*-- Calcular total de operações e valor total*

**SELECT**

**COUNT**(\*),

**IFNULL**(**SUM**(valor\_total\_compra), 0.00)

**INTO**

total\_operacoes,

valor\_total

**FROM** operacao

**WHERE** fk\_usuario\_id\_usuario = p\_id\_usuario

**AND** fk\_tipo\_operacao\_id\_tipo\_operacao = tipo\_id;

*-- Montar string de resposta*

**SET** resultado = **CONCAT**('Operações: ', total\_operacoes, ', Total: R$ ', **FORMAT**(valor\_total, 2));

**RETURN** resultado;

**END**$$

DELIMITER ;

*-- View --*

**CREATE** **VIEW** historico\_precos\_produto **AS**

**SELECT**

p.id\_produto,

p.descricao **AS** nome\_produto,

o.id\_operacao,

o.data\_operacao,

op.preco\_venda,

op.valor\_unitario,

op.quantidade

**FROM**

produtos p

**JOIN** operacao\_produto op **ON** p.id\_produto = op.id\_produto

**JOIN** operacao o **ON** o.id\_operacao = op.id\_operacao

**ORDER** **BY**

p.id\_produto, o.data\_operacao **DESC**;

* 1. **Recursos e ferramentas (1º e 2º)**

Para o desenvolvimento do sistema, foram adotadas diversas ferramentas que contribuíram tanto para a implementação quanto para a modelagem, teste e documentação do projeto. A seguir, estão listadas as principais:

**XAMPP**  
Ferramenta que integra servidor **Apache**, banco de dados **MySQL** e interpretador PHP. Foi usada para simular o ambiente de produção localmente, permitindo o desenvolvimento e testes do sistema de forma prática e centralizada.

**Visual Studio Code (VS Code)**

Editor de código utilizado no desenvolvimento das funcionalidades do sistema. Suportou as linguagens:

**PHP**: backend e regras de negócio;

**HTML e CSS**: construção da interface do usuário;

**PlantUML**: criação de diagramas de caso de uso e outros artefatos UML.

**HeidiSQL**  
Utilizado para gerenciamento da base de dados, execução de scripts SQL, inspeção de tabelas e manipulação de dados. Facilitou a verificação de consistência e execução dos testes de integridade relacional.

**brModelo**  
Ferramenta empregada para a criação do **modelo lógico e físico do banco de dados**. Permitiu a construção do DER (Diagrama Entidade-Relacionamento), auxiliando na organização e normalização das tabelas, campos e relacionamentos antes da implementação no MySQL.

Essas ferramentas permitiram um fluxo de trabalho eficiente, desde a concepção até a validação do sistema, garantindo maior organização, qualidade e produtividade no desenvolvimento.

**GitHub:** Utilizado como plataforma de versionamento de código e colaboração em equipe. Permitimos o controle de versões, criação de branches para desenvolvimento paralelo e registro do histórico de alterações. O GitHub também facilitou a integração entre os membros da equipe, permitindo revisões de código e resolução eficiente de conflitos.

## Etapas / Sprints realizados (1º e 2º)

| **Sprint** | **Objetivo** | **Principais Atividades** | **Observações** |
| --- | --- | --- | --- |
| **1º Sprint** | Planejamento e Estruturação Inicial | - Levantamento dos requisitos funcionais e não funcionais  - Criação dos diagramas de caso de uso (cliente, fornecedor, administrador)  - Modelagem do DER no brModelo  - Estruturação do banco MySQL  - Telas iniciais de cadastro e login  - Testes de mesa | Foco na definição da base do sistema e validação das regras iniciais. |
| **2º Sprint** | Desenvolvimento das Funcionalidades e Integração | - Solicitação de orçamento pelo cliente  - Cotação pelo fornecedor  - Aprovação e realização da compra  - Visualização e gestão de pedidos  - CRUDs de transportadora, categoria e produto  - Testes e refatorações | Etapa marcada pela integração das funcionalidades centrais e ajustes de performance. |
| **3º Sprint** | Refinamento de Funcionalidades e Validações Finais | - Correções baseadas nos testes manuais  - Aprimoramento da usabilidade nas interfaces  - Ajustes no modelo de dados conforme feedback  - Validação dos fluxos principais de navegação e uso | Consolidou os principais fluxos do sistema, priorizando a estabilidade e navegabilidade. |
| **Sprint Futuro** | Acompanhamento de Pedido e Conclusão de Integrações Complementares | - Implementação do acompanhamento de status de pedidos em tempo real  - Finalização do módulo de pagamento completo  - Integração com notificações e relatórios automatizados | Estas funcionalidades foram previstas para sprints futuros, pois o foco atual foi atender os requisitos das disciplinas trabalhadas neste semestre. |

# Testes e Qualidade

## Estratégia de Testes:

A estratégia de testes adotada foi inicialmente fundamentada em **testes de mesa**, com o objetivo de validar o fluxo lógico de cada funcionalidade antes mesmo da execução em ambiente real. Esse processo permitiu identificar incoerências lógicas e comportamentais nas funcionalidades, garantindo que cada módulo estivesse alinhado com os requisitos definidos no escopo do projeto. Após essa etapa, foram realizados testes manuais em ambiente controlado, simulando diferentes perfis de usuário (cliente, fornecedor e administrador), a fim de verificar a integridade e integração entre os módulos do sistema.

## Resultados dos Testes:

Durante os testes realizados, foi identificada uma **inconsistência no banco de dados**, especialmente em consultas que envolviam múltiplos relacionamentos e caminhos alternativos para alcançar os mesmos dados (como pedidos vinculados a diferentes orçamentos e fornecedores). Essa redundância exigiu ajustes no modelo relacional, com padronização de consultas SQL e revisão das chaves estrangeiras. Após as correções, os testes foram reexecutados com sucesso, validando o correto funcionamento das operações de cadastro, orçamento, compra e pagamento.

* 1. **Garantia da Qualidade:**

Para garantir a qualidade do sistema, foram adotadas as seguintes práticas:

**Testes de mesa** para análise lógica de cada funcionalidade antes da implementação prática;

**Refatoração contínua do código-fonte**, com ajustes pontuais baseados nos resultados dos testes;

**Validação cruzada entre os módulos**, testando interações como: solicitação de orçamento → geração de cotação → realização de compra → processamento do pagamento;

**Revisão da modelagem do banco de dados**, com foco em eliminar redundâncias e inconsistências relacionais.

Essas práticas asseguraram a estabilidade e integridade funcional do sistema desenvolvido.

## Requisitos mínimos de hardware e software para o sistema

Hardware mínimo:

Processador: Intel Core i3 ou equivalente

Memória RAM: 4 GB

Armazenamento: 1 GB livre para instalação do sistema e banco de dados

Resolução mínima de tela: 1280x720 pixels

Software necessário:

Sistema Operacional: Windows 10 ou superior / Linux Ubuntu 20.04+

Servidor Web: Apache ou Nginx

Interpretador: PHP 7.4 ou superior

Banco de Dados: MySQL 5.7 ou superior

Navegador compatível: Google Chrome, Mozilla Firefox, Microsoft Edge (versão atualizada)

## Contrato para desenvolvimento de software

CONTRATO DE PRESTAÇÃO DE SERVIÇOS DE DESENVOLVIMENTO DE SOFTWARE

# 1. PARTES CONTRATANTES

1.1 CONTRATANTE:  
MVS Info, pessoa jurídica de direito privado, inscrita no CNPJ sob nº 15.456.789/0001-01\*, com sede em Rua Jarbas Leme Godoy,875 – Jd. José Ometto II – CEP:130606-389 – Araras – SP doravante denominada “MVS”.  
  
1.2 CONTRATADA:  
Morgan Devs, equipe de desenvolvimento composta por:  
- Rafaela Morais – https://github.com/LemesdeMorais  
- Jamila Cardoso – https://github.com/jmcardoso18  
- Felipe Rocha – https://github.com/FlpRocha236  
- Jefferson – https://github.com/Jefferson434  
  
doravante denominada “Morgan Devs”.

# 2. OBJETO

2.1 O presente contrato tem por objeto o desenvolvimento do sistema de gestão comercial, conforme especificações técnicas acordadas, compreendendo:  
  
- Portal do cliente para orçamentos e pedidos;  
- Portal do fornecedor para propostas e status;  
- Administração (clientes, fornecedores, produtos, relatórios, fretes);  
- Marketing digital (folders, vídeos, e-mails);  
- Banco de dados MySQL, frontend responsivo e backend PHP.  
  
2.2 O projeto está documentado e estruturado no arquivo ZIP entregue e anexado a este contrato como Anexo I.

# 3. PRAZO DE EXECUÇÃO

3.1 O desenvolvimento terá prazo estimado de 180 dias corridos, contados a partir da assinatura deste contrato.  
  
3.2 Caso haja necessidade de prorrogação do prazo, esta deverá ser comunicada por escrito pela Morgan Devs, com antecedência mínima de 5 dias, para avaliação pela MVS.

# 4. REMUNERAÇÃO

4.1 O valor total dos serviços será de R$ 10.000,00, podendo ser pago em:  
  
- 50% na assinatura do contrato;  
- 20% na entrega parcial do frontend;  
- 30% na entrega final, com sistema testado e validado.  
  
4.2 Quaisquer valores adicionais decorrentes de solicitações fora do escopo inicial serão negociados em aditivos contratuais.

# 5. OBRIGAÇÕES DA CONTRATADA (Morgan Devs)

a) Desenvolver o software conforme especificações e prazos acordados;  
b) Disponibilizar código-fonte, scripts do banco de dados e documentação técnica;  
c) Garantir a compatibilidade responsiva (dispositivos móveis e desktops);  
d) Oferecer suporte técnico para eventuais ajustes por um período de 30 dias após entrega.

# 6. OBRIGAÇÕES DA CONTRATANTE (MVS)

a) Fornecer informações, materiais e acessos necessários ao desenvolvimento;  
b) Aprovar etapas do projeto no prazo máximo de 5 dias úteis após recebimento;  
c) Efetuar os pagamentos nos prazos estabelecidos.

# 7. DIREITOS AUTORAIS E LICENÇA

7.1 A Morgan Devs cede à MVS o direito de uso do software, exclusivamente para os fins previstos neste contrato.  
  
7.2 O código fonte permanecerá como propriedade intelectual da Morgan Devs, sendo vedada sua revenda ou cessão a terceiros sem anuência prévia.  
  
7.3 A MVS poderá utilizar o sistema livremente, realizar pequenas customizações e hospedar em ambiente próprio.

# 8. CONFIDENCIALIDADE

8.1 Ambas as partes comprometem-se a manter sigilo sobre todas as informações confidenciais trocadas durante o projeto.

# 9. RESCISÃO

9.1 O contrato poderá ser rescindido:  
  
- Por comum acordo entre as partes;  
- Por descumprimento contratual, mediante notificação prévia de 10 dias.

# 10. FORO

10.1 Para dirimir quaisquer controvérsias oriundas deste contrato, fica eleito o foro da comarca de [Cidade/Estado], renunciando a qualquer outro por mais privilegiado que seja.

## Local e data

Araras,26 de junho de 2025.  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
MVS Info  
Contratante  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Morgan Devs  
Contratada  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Testemunha 1  
Nome:  
CPF:  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Testemunha 2  
Nome:  
CPF:

# Considerações Finais

O desenvolvimento deste sistema permitiu aplicar na prática os conhecimentos adquiridos ao longo do semestre, integrando diversas disciplinas da área de tecnologia. Durante o projeto, foram enfrentados desafios técnicos e de organização que exigiram colaboração, planejamento e adaptação contínua. A entrega final contempla os principais requisitos definidos inicialmente, com estrutura sólida de banco de dados, interfaces funcionais e lógica de negócio implementada de forma consistente. Como aprendizado, destacam-se a importância da divisão clara de tarefas, o uso de metodologias ágeis para organização do trabalho e a vivência prática do ciclo completo de desenvolvimento de software, desde a concepção até os testes.

Contribuição Individual

**Rafaela (Front-end):** Responsável pela criação das interfaces visuais do sistema, garantindo boa usabilidade e responsividade nas telas de login, cadastro e gerenciamento de pedidos.

**Felipe (Back-end):** Atuou na implementação das regras de negócio, conexão com o banco de dados e desenvolvimento das funcionalidades de orçamento, cotação e compra.

**Jefferson (Full Stack):** Contribuiu tanto no front-end quanto no back-end, integrando os módulos, realizando testes de integração e refatorando o código para melhor desempenho.

**Jamila (DBA):** Ficou encarregada da modelagem lógica e física do banco de dados, criação de tabelas e procedures em MySQL, além de garantir a integridade e consistência dos dados.

# Lições aprendidas

Durante o desenvolvimento do sistema, a equipe adquiriu uma série de aprendizados importantes, tanto técnicos quanto relacionados à dinâmica de trabalho em grupo. Entre as principais lições aprendidas, destacam-se:

**Importância do planejamento inicial:** A definição clara dos requisitos e da estrutura do banco de dados desde o início, nos perdemos no meio do projeto com o tempo.

**Comunicação constante:** A troca frequente de informações entre os membros da equipe foi essencial para alinhar as expectativas.

**Adaptação ao uso de ferramentas:** O uso de ferramentas como o brModelo para modelagem, o Heidi para banco de dados e plataformas de versionamento (como GitHub) foi fundamental e exigiu aprendizado prático.

**Gestão de tempo e prioridades:** Faltou uma melhor gestão do tempo.

**Testes e validações contínuos:** A prática de testes de mesa e testes manuais permitiu detectar e corrigir erros rapidamente, reforçando a importância de validar cada etapa antes de avançar.

**Colaboração multidisciplinar:** Trabalhar com diferentes perfis (front-end, back-end, DBA, full stack) mostrou como a colaboração entre áreas técnicas contribui para um produto mais robusto e funcional.

# Referências

**Site oficial do MySQL** – <https://dev.mysql.com>  
Documentação técnica para uso do banco de dados MySQL, linguagem SQL e recursos de administração.

**GitHub Docs** – <https://docs.github.com>  
Guia completo para versionamento de código, gerenciamento de repositórios e colaboração em equipe com Git.

**brModelo** – <http://www.sis4.com/brModelo>  
Ferramenta de modelagem utilizada para criação do Diagrama Entidade-Relacionamento (DER) do banco de dados.

**Trello** – <https://trello.com>  
Plataforma de gerenciamento de tarefas baseada em quadros Kanban, utilizada para organização e acompanhamento das sprints do projeto.

**PHP Manual** – <https://www.php.net/manual/pt_BR/>  
Documentação oficial da linguagem PHP, com exemplos e explicações das funções utilizadas no back-end.

**Zeal** – <https://zealdocs.org>  
Aplicativo offline de consulta à documentação de diversas linguagens e frameworks, utilizado como apoio durante o desenvolvimento.

# Anexo I - Diário de bordo

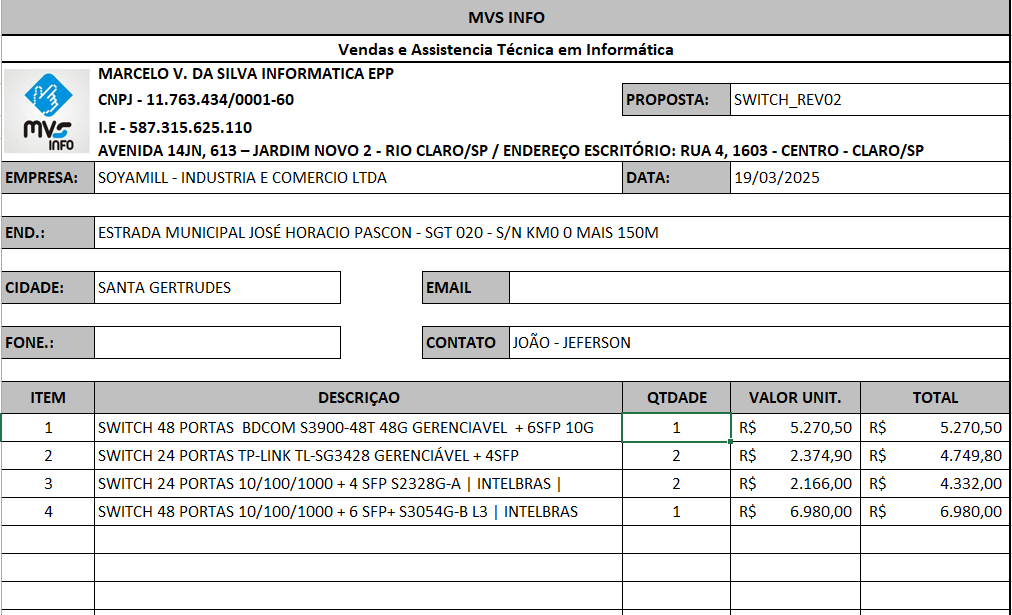
**Físico:** Feito pelo grupo.

**Digital:** individual / Links

# Anexo II – Cronograma efetivo Anexo III –

# 

# Evidências



Print da folha de pedido que o Stakeholder