**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**NÚCLEO DE EDUCAÇÃO A DISTÂNCIA

Pós-graduação *Lato Sensu* em Desenvolvimento Web Full Stack

**Allan Cezar Almeida Chaves**

**SISTEMA DE VENDAS EMASA MINERADORA**

Belo Horizonte

2023

**SUMÁRIO**

[1. Apresentação 3](#_Toc116590715)

[1.1 Contexto 3](#_Toc116590716)

[1.2 Público Alvo 7](#_Toc116590717)

[1.3 Requisitos 8](#_Toc116590718)

[2 Modelagem 17](#_Toc116590719)

[2.1 Diagrama de casos de uso 17](#_Toc116590720)

[2.2 Atores 17](#_Toc116590721)

[2.3 Detalhamento dos casos de uso 18](#_Toc116590722)

[2.4 Projeto de Interface 26](#_Toc116590723)

[2.5 Diagrama de classes 29](#_Toc116590724)

[3 Projeto 30](#_Toc116590725)

[3.1 Arquitetura de *software* 30](#_Toc116590726)

[3.2 Arquitetura da informação 33](#_Toc116590727)

[4 Testes 34](#_Toc116590728)

[5 URLs 35](#_Toc116590729)

[5.1 Aplicação web 35](#_Toc116590730)

[5.2 Repositório código-fonte 35](#_Toc116590731)

[5.3 Vídeo de apresentação do trabalho 35](#_Toc116590732)

[6 REFERÊNCIAS 36](#_Toc116590733)

# Apresentação

Este trabalho implementa uma solução WEB completa para substituir o sistema legado de vendas utilizado por uma empresa de mineração. Neste projeto são documentados todos os artefatos necessários para os processos de modelagem e concepção do produto, sob um ponto de vista da gestão ágil e boas práticas de desenvolvimento.

## Contexto

Desde o final da década de 80, a empresa de mineração EMASA Mineração S/A utiliza um sistema legado de vendas e faturamento. O sistema foi criado com o uso do Clipper 5, um compilador de 16 bits que implementava a linguagem de programação (baseada na linguagem de programação dBASE) como descrito por (Prestwood 1993). O xBASE além de ser uma linguagem de programação, também era um padrão de banco de dados muito utilizado nas décadas de 80 e 90 segundo (Rupp 1991).

O sistema está implementado em um computador antigo de 1994, um *Pentium* 100, arquitetura x86 de 32 bits, com 16MB de memória RAM instalados e um Disco rígido de 2GB. O sistema operacional utilizado no computador é o *Windows 98*, com uma arquitetura de 32 bits e com compatibilidade com sistemas de 16bits. Essa versão do *Windows* disponibiliza a versão 7 do *MS-DOS* de 16/32 bits (Lowe 1999), que é o sistema operacional que é a base para a execução do sistema atual, conforme mostrado na Figura 1.

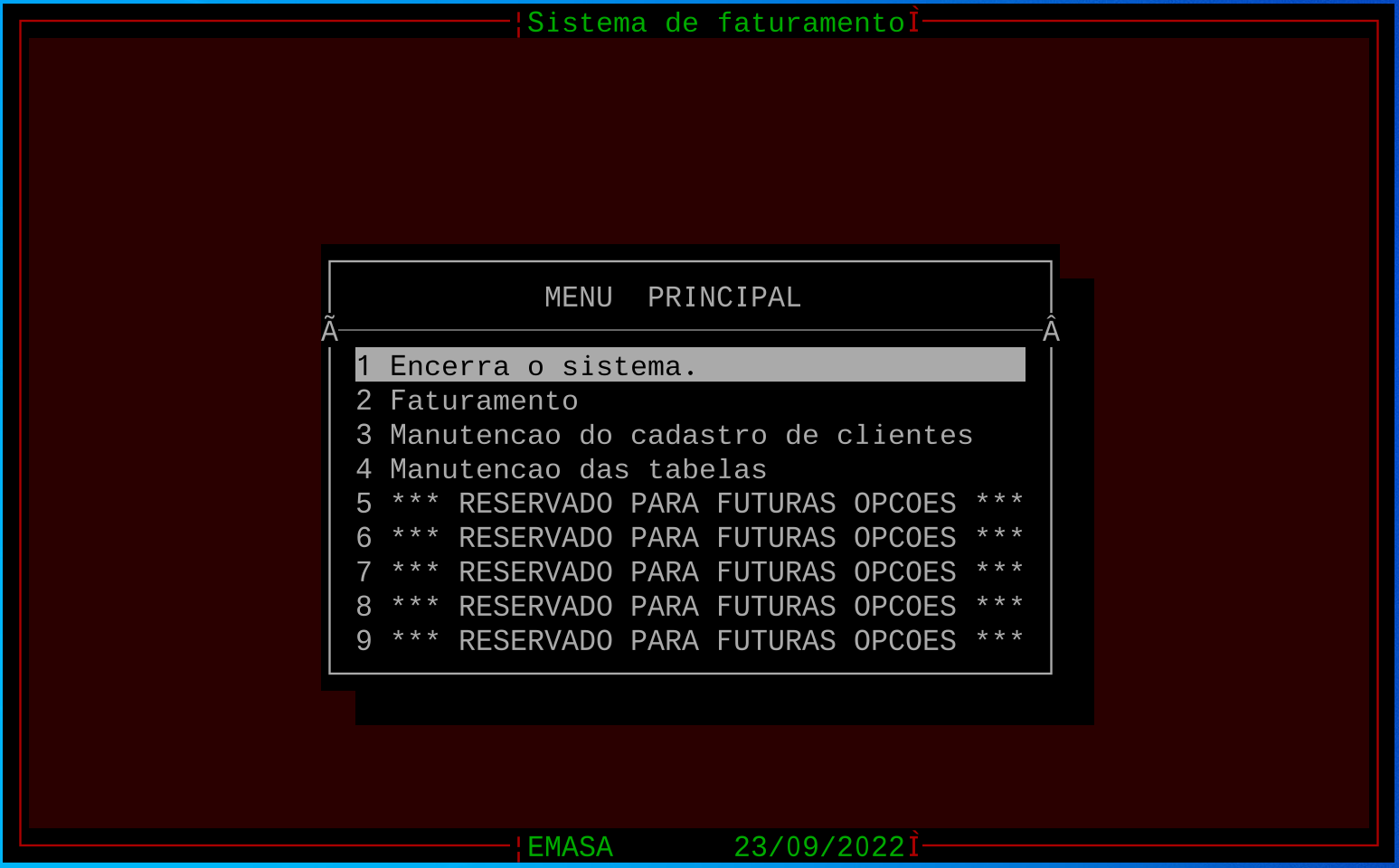


Figura 1 - Tela inicial do sistema legado.

O sistema legado é centralizado, ou seja, não opera em uma rede de computadores e não permite a execução por mais de um computador simultaneamente. Como a aplicação acessa uma base de dados não gerenciada por um SGBD, não é permitida a concorrência aos dados, conforme mensagem de erro exibida pelo sistema na Figura 2Figura 4.

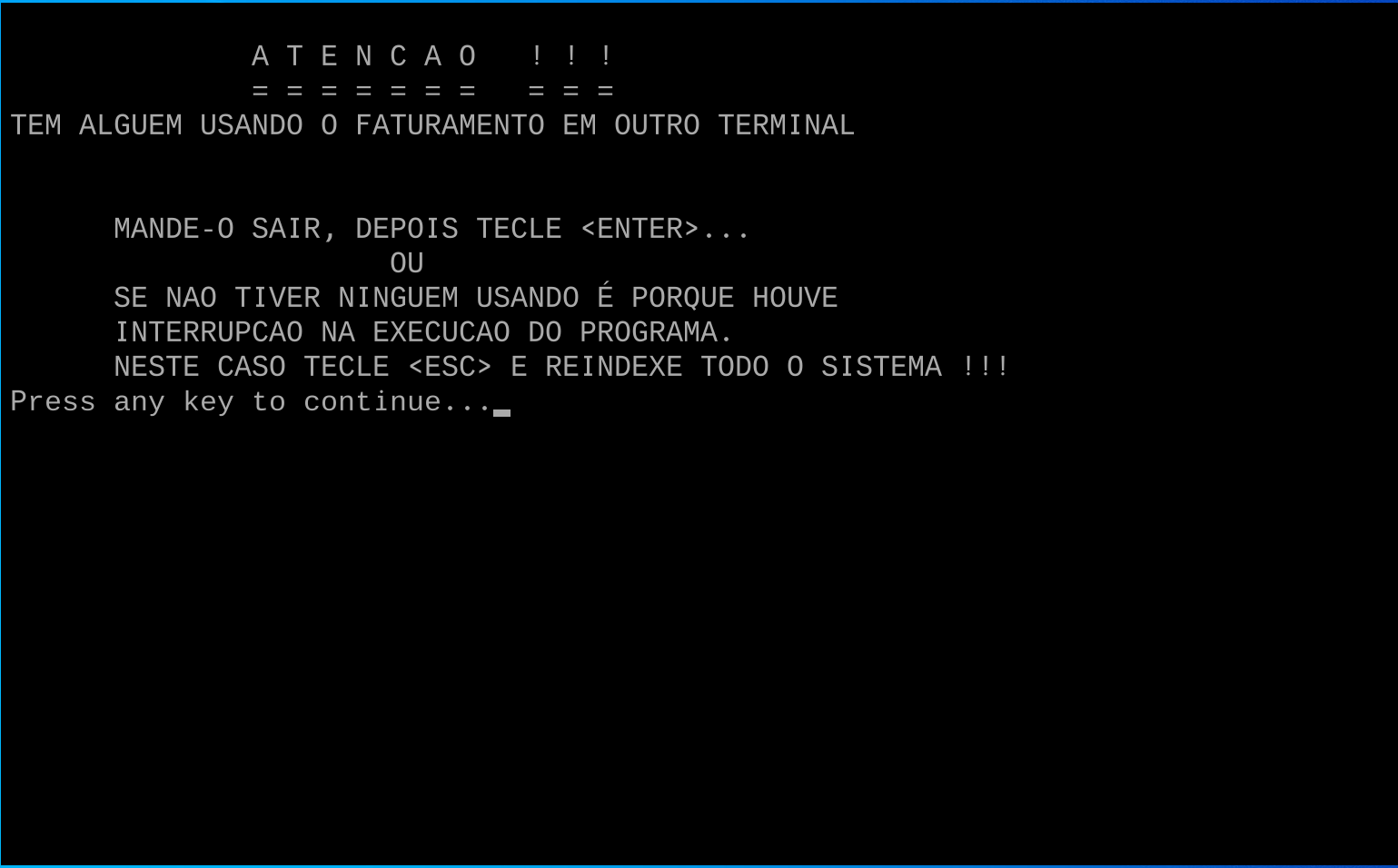


Figura 2- Mensagem de erro por uso simultâneo do Sistema

Além da falta de concorrência, outras funcionalidades que existem em um SGBD e não são implementadas pelo sistema atual, são a segurança e confiabilidade dos dados (Date 2004). Os arquivos não possuem restrições de acesso. Portanto, podem ser abertos e manipulados com o uso qualquer utilitário de banco de dados que interprete arquivos .DBX (Rupp 1991) , conforme apresentado na Figura 3.

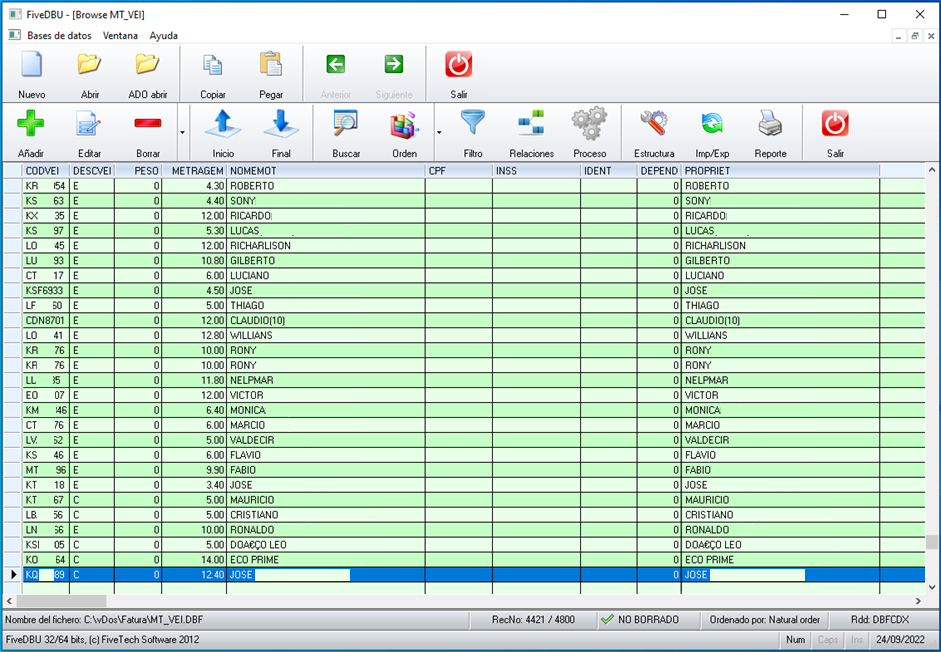


Figura 3- Utilitário para manipulação de base de dados no formato .DBX

A usabilidade do sistema não é intuitiva. Por ser executado no sistema operacional *MS-DOS*, não utiliza uma interface gráfica de usuário, ou em Inglês *GUI* (*Graphics User Interface*) (MYERS e ROSSON 1992). Sendo assim, não provê interação com um periférico auxiliar como o mouse. A interação do usuário com o sistema é realizada através do teclado, conforme mostrado na Figura 4.

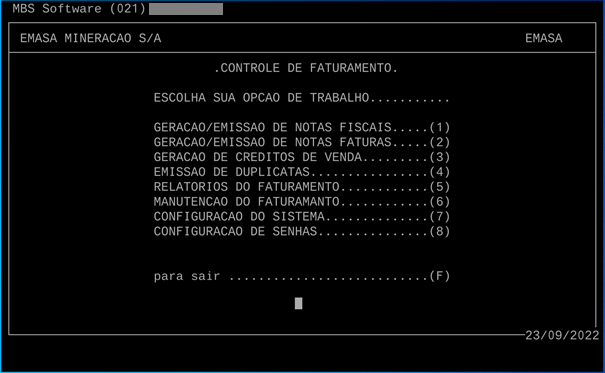


Figura 4- Tela de controle de faturamento do sistema legado

Ademais, há uma alta complexidade implementada no fluxo de vendas, conforme mostrado na Figura 5. Toda complexidade, interfere no tempo total de cada atendimento, causando insatisfação dos clientes com o tempo de espera de cada venda.

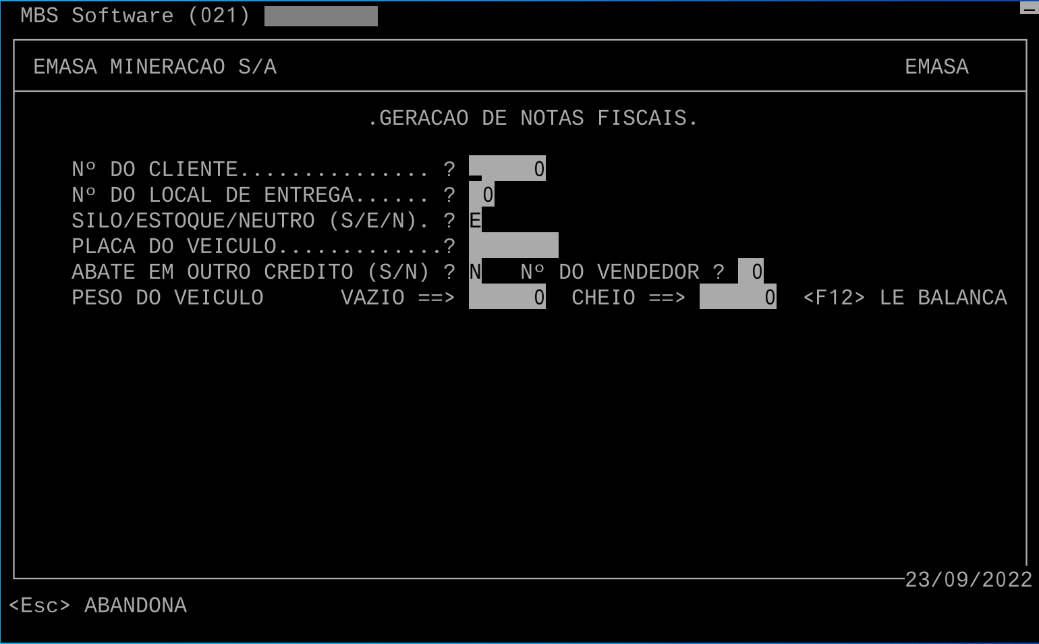


Figura 5- Tela de geração de vendas com notas Fiscais

Como exemplo podemos citar algumas dificuldades no fluxo de vendas, conforme abaixo:

* As regras de negócios devem ser conhecidas e não há uma abstração dessas regras para o usuário;
* Não há tratamento de erros durante a interação do usuário;
* Não é permitido navegação entre telas sem perda de informações já preenchidas;
* Necessário um treinamento extenso do operador do sistema;
* Erros na modelagem de entidades, relacionamentos e atributos

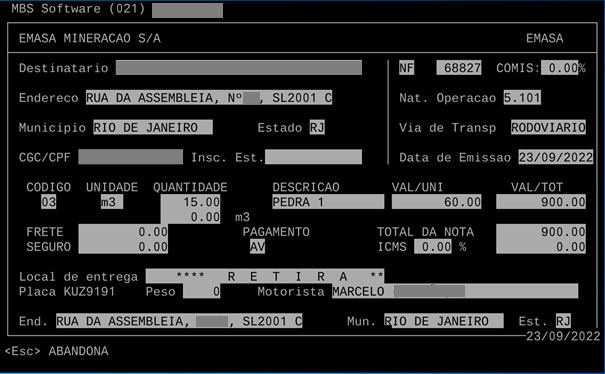


Figura 6- Tela principal de informações de vendas

### Lean Canvas MVP

Visando a validar a ideia do produto, este trabalho utilizou a ferramenta Lean Canvas (MAURYA 2012), conforme a Figura 7.

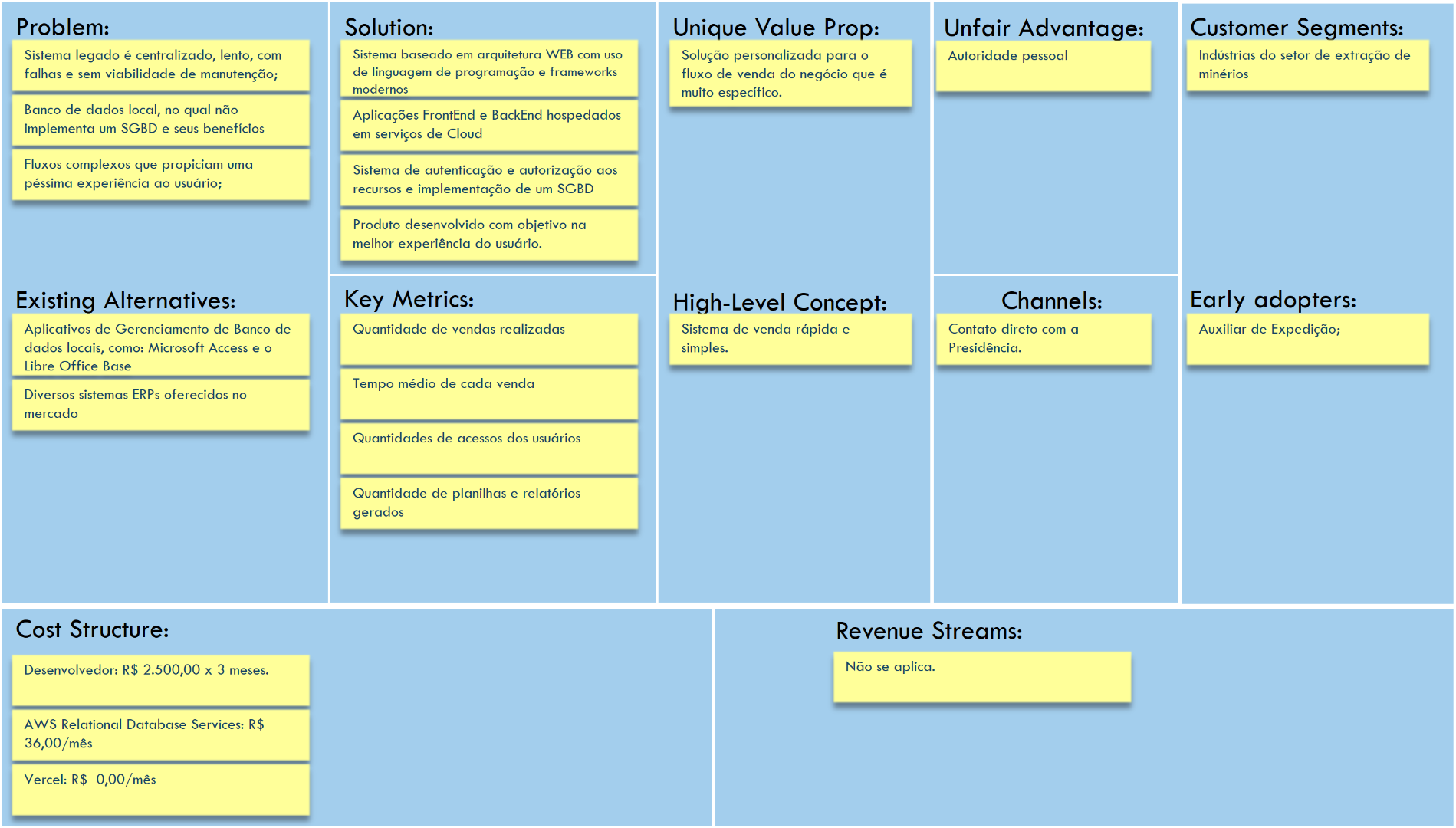


Figura 7 Lean Canvas

## Público Alvo

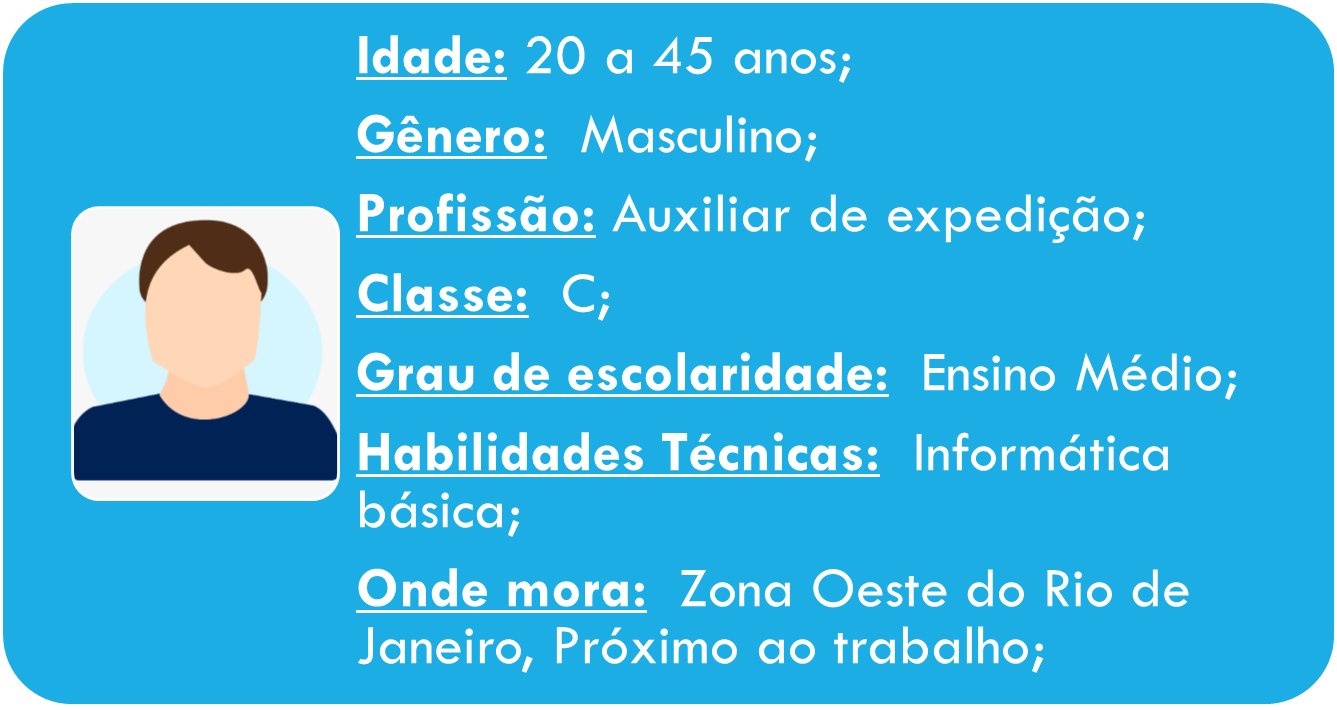


Figura 8- Público alvo do sistema

## Funcionalidades e Requisitos

Este trabalho buscou elucidar os requisitos funcionais através da elaboração do backlog de produto (COHN 2004). Foi proposto partir do princípio que não teríamos os requisitos definidos no início do projeto. As soluções resolvidas por um novo sistema seriam descobertas, em conjunto, com base nas funcionalidades que agregassem maior valor à empresa.

### Construção do *backlog* de produto.

Para a definição das funcionalidades, aplicamos um processo da metodologia ágil conhecido como *PBB – Product Backlog Building* (Aguiar e Caroli 2021)*.* Buscamos, nesse processo, compreender um pouco mais das características dos usuários do sistema e suas necessidades.

O primeiro item do backlog do produto foi a definição do épico, como mostrado na Figura 9. Após reuniões com os stakeholders conseguimos obter uma síntese das necessidades dos principais usuários do sistema.

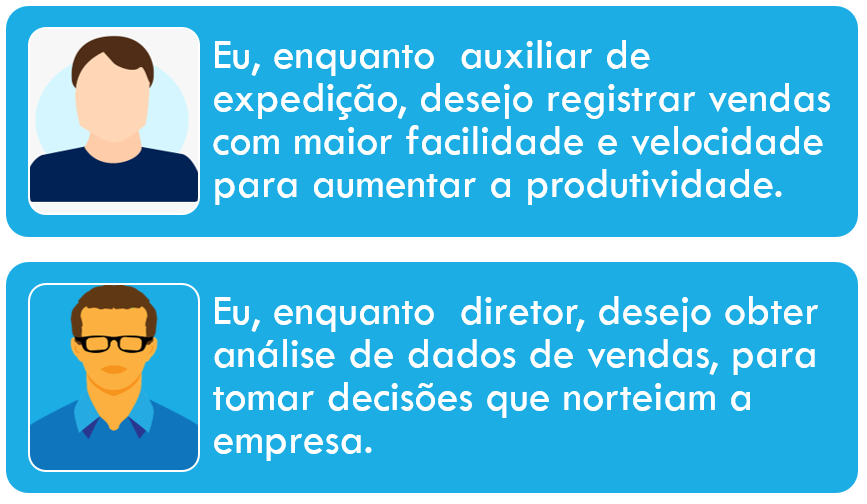


Figura 9 - Escrita dos épicos

Ademais, buscamos compreender melhor o comportamento do usuário. Para tal finalidade, criamos uma persona, conforme a Figura 10. A criação da persona foi baseada em informações obtidas sobre os últimos funcionários do setor de expedição/venda de produtos.

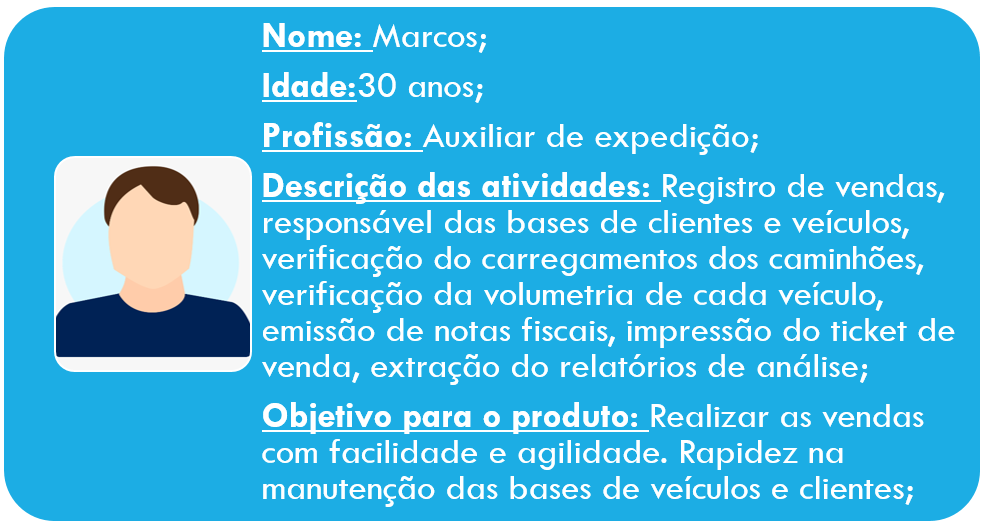


Figura 10 – Persona

Para definirmos os objetivos dos usuários com o sistema, buscamos entender quais seriam as interações do usuário com o sistema. Para isso, desenhamos a jornada comumente realizada do usuário e seus pontos de contato com o sistema, como mostram a Figura 11 e a Figura 12. Os itens em cinza são passos onde não há interação com o sistema. Os itens em verde escuro são os pontos de contato do usuário com o sistema. Os Passos em verde claro são passos dos usuários que foram identificadas como funcionalidades futuras.

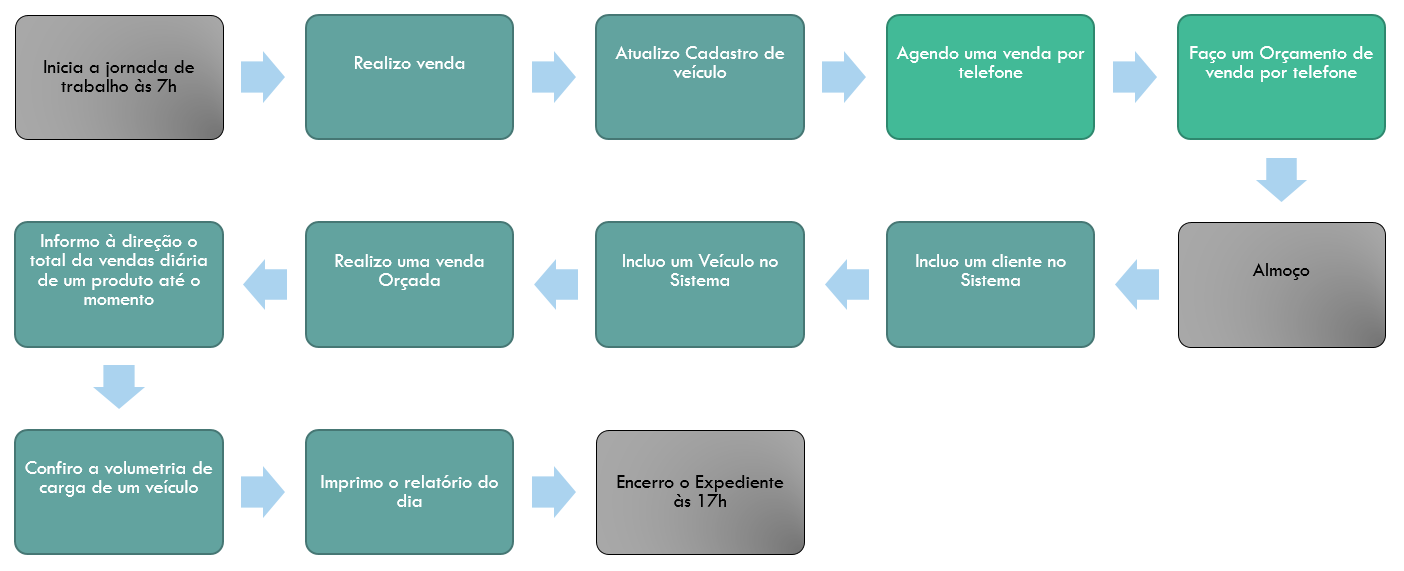


Figura 11 - Jornada do usuário auxiliar de expedição

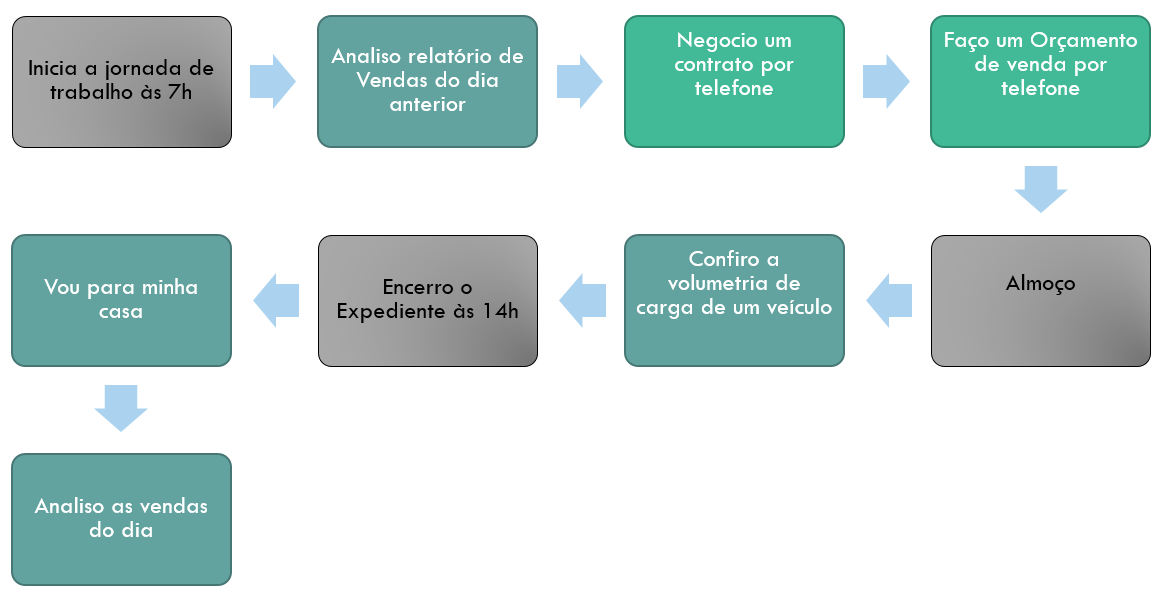


Figura 12 - Jornada do usuário Diretor

Ao observar a jornada dos usuários, podemos identificar as principais funcionalidades exigidas para o sistema novo. A seguir, na Tabela 1mostraremos cada funcionalidade identificada, os problemas nas quais ela se propõe resolver e seus benefícios.

|  |  |  |
| --- | --- | --- |
| Funcionalidade | Problemas a serem resolvidos | Benefícios |
| Gerenciar Registro de Clientes | Modelo de dados atual não contempla informações úteis dos Clientes. | Nova modelagem de dados para a entidade Clientes. |
| Não é possível listagem de Usuários na tela, sendo possível somente a impressão da lista. | A implementação de um SGBD permite filtrar e listar usuários e exibi-los em páginas. |
| Gerenciar Registro de Veículos | Modelo de dados atual não contempla informações úteis dos Veículos. | Nova modelagem de dados para a entidade Veículos. |
| Não é possível listagem de Veículos na tela, sendo possível somente a impressão da lista. | A implementação de um SGBD permite filtrar e listar informações dos veículos e exibi-los em páginas. |
| Realizar Acesso ao Sistema | Não existe controle de acesso ao Sistema legado | Sistema com autenticação e autorização aos recursos |
| Realizar Cadastro de Produtos | O sistema não contempla a volumetria de cada produto por estoque | Nova modelagem de dados contemplará controle dos estoques. |
| Realizar Geração de Planilhas | Relatórios impressos em uma impressora matricial | A arquitetura WEB permite que qualquer relatório seja impresso em qualquer tipo de impressora ou exportado para arquivo no formato .PDF |
| Alto consumo e gasto com papel | Relatórios são exibidos em planilhas. Facilitando a visualização e excluindo a obrigação da impressão. |
| O processo de consultas nas tabelas é lento | A implementação de um SGBD fornece uma consulta relacional entre as tabelas otimizada, provendo resultados mais rápidos para o usuário. |
| Informações que não fornecem dados analíticos | A possibilidade de geração de planilhas em Excel, possibilita que o usuário visualiza, filtre e organize os dados da melhor maneira. Sendo assim, permite a melhor extração de informações úteis. |
| Indisponibilidade das informações | A arquitetura WEB fornece disponibilidade dos dados em qualquer lugar e em qualquer dispositivo com acesso à internet. |
| Realizar Registro de Vendas | Usabilidade do sistema legado é ruim. | Interface otimizada para melhor experiência do usuário. |
| Alta complexidade no processo atual. | Novo fluxo para facilitar a experiência do usuário. |
| Sistema legado apresenta constantes erros de indexação das tabelas após travamentos | Base de dados gerenciada por um SGBD, permitindo concorrência de acesso, confiabilidade e segurança. |
| Impressão do comprovante de pagamento é demorada e trava constantemente o sistema. | Por ser um sistema WEB, o navegador dispara o pedido de impressão para o Sistema Operacional. |

Tabela 1 - Funcionalidades do Sistema

Visando a entrega de um produto mínimo viável(MVP) para experimentação do cliente, este trabalho buscou priorizar as funcionalidades. Para este fim, foram aplicadas diversas ferramentas de priorização do backlog do produto.

A primeira técnica de priorização aplicada foi a MoSCoW, conforme mostra a Tabela 2. As funcionalidades foram classificadas pelo valor à empresa, riscos às entregas e suas dependências.

|  |  |  |
| --- | --- | --- |
| Funcionalidades | MoSCoW | Prioridade |
| Realizar registro de Clientes | Tem que ter | 1 |
| Realizar registro de Veículos | Deveria ter | 1 |
| Realizar registro de Vendas | Tem que ter | 1 |
| Realizar Geração de Planilhas | Deveria ter | 2 |
| Realizar Acesso ao Sistema | Poderia ter | 4 |
| Realizar Cadastro de Produtos | Poderia ter | 3 |

Tabela 2 - Técnica de priorização MoSCoW

Após a priorização foi feito, em conjunto com a equipe, uma revisão dos níveis de confiança para o desenvolvimento de cada funcionalidade, como mostra a Figura 13.

O eixo horizontal representa a confiança da equipe em como criar a funcionalidade. Já o eixo vertical representa a confiança da equipe em conhecer o que o usuário espera dessa funcionalidade.

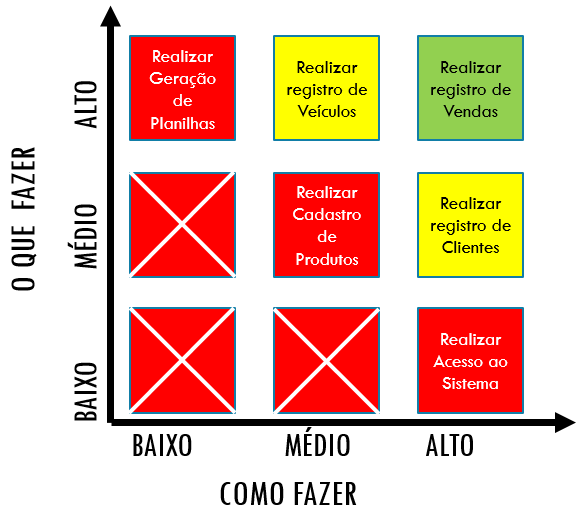


Figura 13 - Gráfico do nível de confiança de desenvolvimento das funcionalidades.

Como podemos observar na Figura 13, a funcionalidade “Realizar Registro e Vendas” está posicionado mais à direita do gráfico. Ou seja, a equipe tem um alto nível de confiança em como desenvolver a atividade. Isto se deve ao fato de ter ocorrido diversas entrevistas com os usuários e um bom entendimento de suas necessidades. Também é possível observar que esta funcionalidade está mais acima no eixo vertical. Ou seja, a equipe tem uma confiança alta em como desenvolver essa funcionalidade de acordo com a necessidade do usuário.

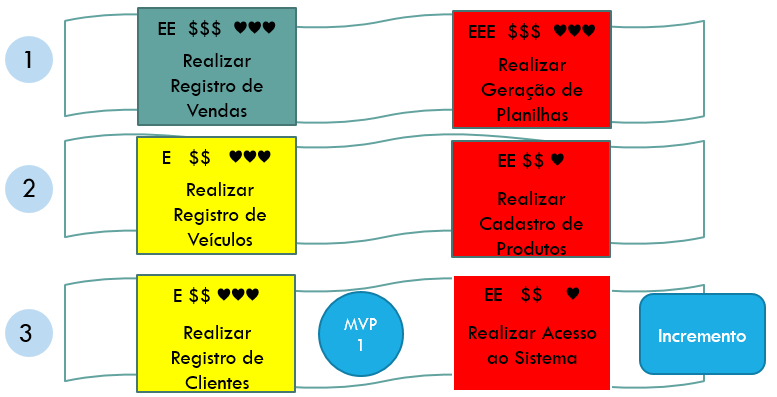
Outro processo de priorização é avaliar as funcionalidades quanto ao esforço técnico, valor para negócio e valor de experiência do usuário. A Tabela 3 demonstra essa análise qualitativa para cada funcionalidade.

|  |  |  |  |
| --- | --- | --- | --- |
| Funcionalidades | Esforço Técnico | Valor para o Negócio | Valor de experiência do usuário |
| Realizar registro de Clientes | E | $ $ | ♥♥ ♥ |
| Realizar registro de Veículos | E | $ $ | ♥♥♥ |
| Realizar registro de Vendas | E E | $ $ $ | ♥♥♥ |
| Realizar Geração de Planilhas | E E E | $ $ $ | ♥♥♥ |
| Realizar Acesso ao Sistema | E E | $ $ | ♥ |
| Realizar Cadastro de Produtos | E E | $ $ | ♥ |

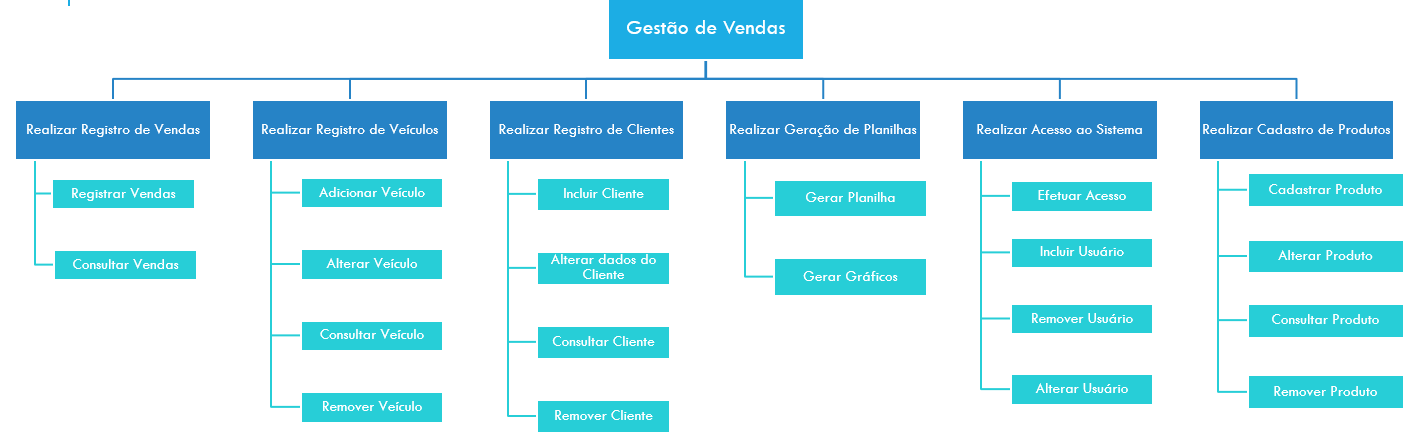
Tabela 3 - Tabela de esforço, negócio e UX das funcionalidades.

A letra “E”, na coluna “Esforço Técnico”, representa o quão uma funcionalidade requer de esforço técnico da equipe. Já o símbolo “$”, na coluna “Valor para o Negócio”, representa o valor de uma funcionalidade para o negócio/usuário. Por fim, o símbolo “♥” representa o quão importante será a experiência de usuário em cada funcionalidade.

Mais uma ferramenta de priorização aplicada, foi o sequenciador. Nele, as funcionalidades foram priorizadas relativamente umas às outras. Essa ferramenta serve para a identificar e validar os MVPs (Produto mínimo viável) e incrementos do produto.



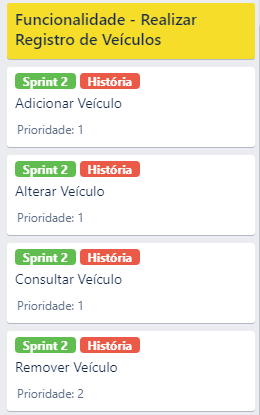
Após a priorização foi criado um diagrama em árvore para compreender a organização e granularidade do épico, das funcionalidades e dos itens do backlog do produto

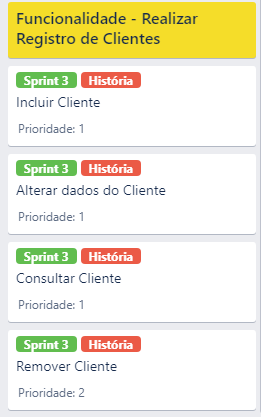


### *Backlog* do produto.

A Figura abaixo mostra um exemplo do backlog do produto, criado em uma plataforma que adota o paradigma Kanban.





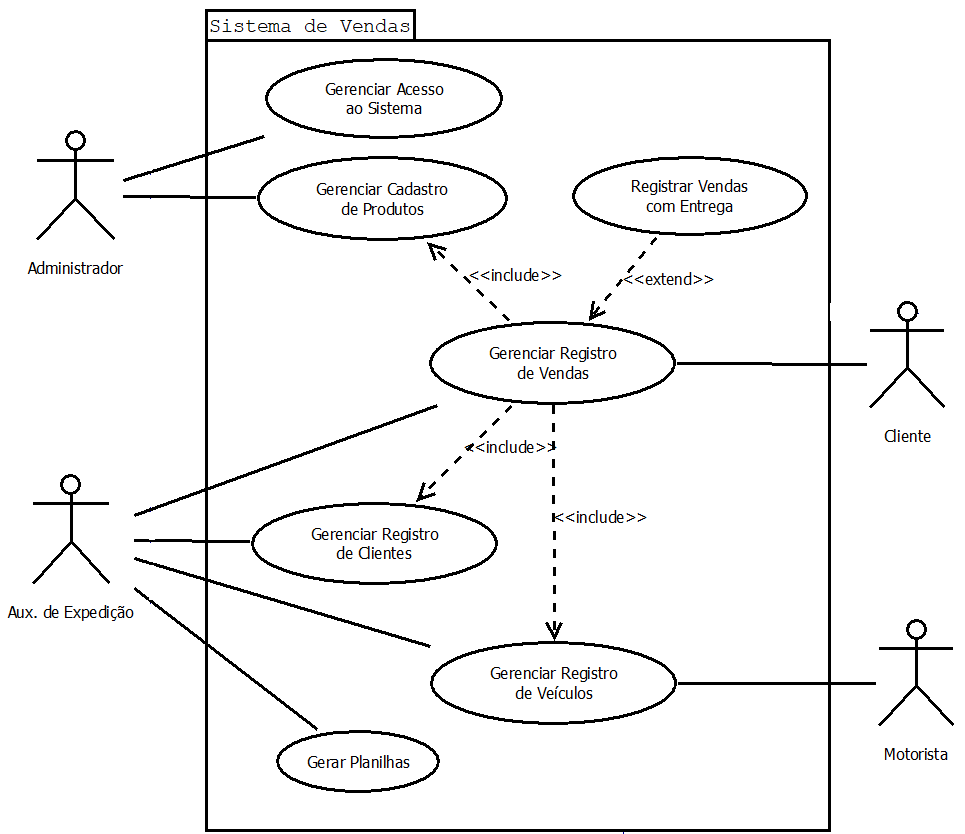


### *Lista de Requisitos Funcionais*.

|  |  |
| --- | --- |
| ID | Descrição do Requisito |
| RF01 | O sistema deve registrar as vendas |
| RF02 | O sistema deve imprimir um ticket da venda |
| RF03 | O sistema deve permitir buscar e consultar uma venda |
| RF04 | O sistema deve cancelar uma venda somente por um usuário administrador |
| RF05 | O sistema deve cadastrar veículos |
| RF06 | O sistema deve permitir buscas e consultar veículos |
| RF07 | O sistema deve permitir alterar os dados de um veículo |
| RF08 | O sistema deve permitir excluir o registro de um veículo |
| RF09 | O sistema deve cadastrar clientes |
| RF10 | O sistema deve permitir buscas e consultar clientes |
| RF11 | O sistema deve permitir alterar os dados de um cliente |
| RF12 | O sistema deve permitir excluir o registro de um cliente |
| RF13 | O sistema deve cadastrar produtos |
| RF14 | O sistema deve permitir buscas e consultar produtos |
| RF15 | O sistema deve permitir alterar os dados de um produto |
| RF16 | O sistema deve permitir excluir o registro de um produto |
| RF17 | O sistema deve permitir cadastrar usuários |
| RF18 | O sistema deve permitir buscar e consultar dados dos usuários |
| RF19 | O sistema deve permitir alterar os dados e permissões de um usuário |
| RF20 | O sistema deve permitir excluir um usuário |
| RF21 | O sistema deve restringir o acesso ao sistema apenas para usuários cadastrados |
| RF22 | O sistema deve gerar planilhas de vendas no formato .XLS |

# Modelagem

## Diagrama de casos de uso



## Atores

|  |  |
| --- | --- |
| Ator | Especializações |
| Usuário | Auxiliar de Expedição;  Administrador; |
| Cliente | - |
| Motorista | - |

## 

## Detalhamento dos casos de uso

Esta seção exibe tabelas com o detalhamento dos principais casos de uso do sistema apresentados no diagrama de casos de uso.

|  |  |
| --- | --- |
| ID: | UC01 |
| Nome: | Registrar Vendas! |
| Escopo: | Sistema de vendas EMASA S.A. |
| Nível: | Objetivo |
| Objetivo: | Registrar a venda de um Produto na base de dados e gerar um ticket da venda; |
| Requisitos: | Conexão com a *internet*; Aparelho com navegador WEB; |
| Atores: | Usuário Auxiliar de Expedição; Cliente; |
| Pré-Condições: | Cliente deve ser cadastrado previamente;  Veículo deve estar cadastrado previamente; |
| Pós-Condições: | Ticket da venda impresso;  As informações das vendas serão armazenadas em um banco de dados relacional; |
| Fluxo Principal: | 1. O caso de uso inicia quando o Usuário Auxiliar de Expedição digita o ID do Cliente e confirma apertando Enter no teclado ou clicando no botão de "Avançar". 2. O sistema verifica se o ID do Cliente é válido e exibe uma lista de veículos vinculados ao Cliente. 3. O Usuário Auxiliar de Expedição seleciona o veículo utilizado para o transporte do produto e confirma os dados do Cliente e do veículo clicando no botão de "Avançar". 4. O sistema exibe a lista de produtos disponíveis na tela. 5. O Usuário Auxiliar de Expedição seleciona o produto carregado pelo Cliente. 6. De acordo com o produto selecionado, o sistema exibe uma lista de origens no estoque. 7. O Usuário Auxiliar de Expedição escolhe a origem do estoque e confirma a venda clicando no botão de "Confirmar Venda". 8. O sistema verifica se a nota fiscal eletrônica foi emitida e, se sim, exibe uma tela para imprimir o ticket da venda. Caso contrário, o sistema exibe uma mensagem informando que a nota fiscal eletrônica ainda não foi emitida e o caso de uso é finalizado. 9. O Usuário Auxiliar de Expedição confirma a impressão do ticket de venda clicando no botão de "Imprimir Ticket" e finaliza o caso de uso. |
| Fluxos Alternativos: | 1a - O cliente não sabe o número do ID  1a.1 O cliente informa o nome;  1a.2 O usuário busca pelo nome do cliente;  1a.3 O Sistema exibe uma lista de clientes de acordo com a busca;  1a.4 O usuário escolhe o cliente correto;  1a.5 Retorna ao fluxo principal no passo 3; |
| Erros/Exceções: | 4a - O Veículo não está cadastrado  4a.1 O cliente deverá agendar uma análise de volumetria do veículo para cadastrá-lo;  4a.2 O caso de uso é finalizado; |
| Mensagens: | * ID do Cliente inválido. Por favor, digite um ID válido. * Cliente não encontrado. Por favor, verifique se o nome está correto e tente novamente. * Veículo não cadastrado. Por favor, agende uma análise de volumetria da caçamba do veículo para cadastrá-lo. * Produto não encontrado. Por favor, verifique se o produto está disponível e tente novamente. * Quantidade de produto selecionada não disponível. Por favor, escolha outra quantidade de produto ou tente novamente mais tarde. * Origem do estoque não selecionada. Por favor, escolha a origem do estoque para continuar. * Nota fiscal eletrônica não emitida. Aguarde a emissão da nota fiscal eletrônica para continuar. * Ticket de venda impresso com sucesso. * Erro ao imprimir o ticket de venda. Por favor, tente novamente mais tarde ou entre em contato com o suporte técnico. * Venda registrada com sucesso. * Erro ao registrar a venda na base de dados. Por favor, tente novamente mais tarde ou entre em contato com o suporte técnico. |
| Regras de negócio: | [RN01] – O ticket deve ser emitido em 4 vias;  [RN02] – O processo só é iniciado após a emissão da nota fiscal eletrônica; |

|  |  |
| --- | --- |
| ID: | UC02 |
| Nome: | Registrar Veículos ! |
| Escopo: | Sistema de vendas EMASA S.A. |
| Nível: | Objetivo |
| Objetivo: | Cadastrar na base de dados as informações de identificação de um veículo de transporte e sua capacidade do volume de carga. |
| Requisitos: | Conexão com a *internet*; Aparelho com navegador WEB; |
| Atores Principais: | Usuário Auxiliar de Expedição; Motorista; |
| Pré-Condições: | Agendamento prévio do cadastramento do veículo;  Dimensões da caçamba de carga aferidas por um funcionário autorizado da empresa; |
| Pós-Condições: | As informações serão armazenadas em um banco de dados relacional; |
| Fluxo Principal: | 1. O caso de uso inicia quando o usuário insere a placa do veículo. O usuário confirma apertando Enter no teclado ou clicando no botão de "Avançar". 2. O sistema verifica se a placa do veículo já está cadastrada no banco de dados. Se a placa já existir, o sistema exibe uma mensagem informando que a placa já está cadastrada e solicita que o usuário verifique se a informação está correta. Caso contrário, o sistema exibe um formulário com informações de identificação do veículo. 3. O usuário preenche o formulário com informações do veículo, incluindo a marca, o modelo, a altura, a largura e o comprimento da caçamba de carga. O usuário confirma clicando no botão de "concluir". 4. O sistema verifica se todas as informações obrigatórias foram preenchidas corretamente e, se sim, armazena as informações no banco de dados. O sistema exibe uma mensagem informando que o cadastro do veículo foi realizado com sucesso e finaliza o caso de uso. Caso contrário, o sistema exibe uma mensagem informando que as informações estão incompletas e solicita que o usuário preencha todas as informações obrigatórias antes de prosseguir. |
| Fluxos Alternativos: | Não se aplica |
| Erros/Exceções: | 1a - O cliente não agendou o cadastramento.  1a.1 O caso de uso é encerrado e o motorista ou o cliente são orientados a agendar o cadastramento do veículo; |
| Mensagens: | * Placa já cadastrada. Por favor, verifique se a informação está correta. * Cadastro do veículo realizado com sucesso. |
| Regras de negócio: | [RN01] – O Cadastramento do veículo pode ser feito por um celular ou tablet com acesso à internet ou rede WI-FI da empresa, permitindo a inserção das informações na área externa da empresa; |

|  |  |
| --- | --- |
| ID: | UC03 |
| Nome: | Registrar Clientes ! |
| Escopo: | Sistema de vendas EMASA S.A. |
| Nível: | Objetivo |
| Objetivo: | Cadastrar um Cliente; |
| Requisitos: | Conexão com a *internet*; Aparelho com navegador WEB; |
| Atores: | Usuário Auxiliar de Expedição;  Usuário cliente; |
| Pré-Condições: | CNPJ do cliente ser válido; o cliente deve apresentar documentação válida da empresa para conferência. |
| Pós-Condições: | As informações serão armazenadas em um banco de dados relacional; |
| Fluxo Principal: | 1. O caso de uso inicia quando o usuário insere o CNPJ do cliente no sistema e confirma apertando Enter no teclado ou clicando no botão de "Avançar". 2. O sistema verifica se o CNPJ do cliente é válido e exibe um formulário com outras informações de identificação do cliente, incluindo nome, telefone, e-mail e data de registro. 3. O usuário preenche o formulário com as informações do cliente e confirma clicando no botão de "concluir". 4. O sistema verifica se todas as informações obrigatórias foram preenchidas corretamente e, se sim, armazena as informações no banco de dados. O sistema exibe uma mensagem informando que o cadastro do cliente foi realizado com sucesso e finaliza o caso de uso. Caso contrário, o sistema exibe uma mensagem informando que as informações estão incompletas e solicita que o usuário preencha todas as informações obrigatórias antes de prosseguir. |
| Fluxos Alternativos: | Não se aplica |
| Erros/Exceções: | Não se aplica |
| Mensagens: | * As informações do cliente estão incompletas. Por favor, preencha todos os campos obrigatórios antes de prosseguir. * O CNPJ do cliente é inválido. Por favor, verifique se o número foi digitado corretamente. * Não foi possível realizar o cadastro do cliente. Por favor, tente novamente mais tarde. * O cliente já está cadastrado em nosso sistema. Por favor, verifique se não há duplicidade de cadastro. * O cliente não pode ser cadastrado. Por favor, entre em contato com o suporte técnico. |
| Regras de negócio: | [RN01] – O Cadastramento do cliente é restrito para pessoa jurídica;  [RN02] – O CNPJ do cliente deve ser válido;  [RN03] – O cliente deve apresentar documentação válida da empresa para conferência; |

|  |  |
| --- | --- |
| ID: | UC04 |
| Nome: | Gerar Planilhas ! |
| Escopo: | Sistema de vendas EMASA S.A. |
| Nível: | Objetivo |
| Objetivo: | Gerar Planilhas para análise de negócios da empresa; |
| Requisitos: | Conexão com a *internet*; Aparelho com navegador WEB; |
| Atores: | Usuário Auxiliar de Expedição; |
| Pré-Condições: | Não se aplica |
| Pós-Condições: | Geração de planilha e gráficos em diferentes formatos (XLS, CSV, PDF) e envio por e-mail. |
| Fluxo Principal: | 1. O caso de uso inicia quando o sistema solicita ao usuário o período de análise desejado. 2. O usuário escolhe o período de interesse na análise das vendas. 3. O sistema solicita ao usuário o cliente de interesse na análise das vendas. 4. O usuário escolhe o cliente de interesse na análise das vendas. 5. O sistema solicita ao usuário o produto de interesse na análise das vendas. 6. O usuário escolhe o produto de interesse na análise das vendas. 7. O sistema apresenta as informações na forma de planilha e gráficos, permitindo que o usuário visualize as informações de diferentes maneiras. 8. O usuário seleciona o formato de arquivo desejado para exportar a planilha e os gráficos (XLS, CSV ou PDF). 9. O usuário pode salvar o arquivo no computador ou enviar por e-mail diretamente pelo sistema. |
| Fluxos Alternativos: | 1a – O usuário não escolhe o período específico  1a.1 - O sistema aplica o primeiro dia do ano corrente e a data atual como início e fim do período de busca;  1a.2 - Retorna ao fluxo principal no passo 2;  2a – O usuário não escolhe um cliente específico  2a.1 – O sistema aplica todos clientes da base para a geração da planilha;  2a.2 – Retorna ao fluxo principal no passo 3;  3a – O usuário não escolhe um produto específico  3a.1 – O sistema aplica todos os produtos como filtro para a geração da planilha;  3a.2 – Retorna ao fluxo principal no passo 4; |
| Erros/Exceções: | Não se aplica |
| Mensagens: | * Planilha e gráficos gerados com sucesso. * Arquivo exportado com sucesso. * E-mail enviado com sucesso. |
| Regras de negócio: | * O sistema permite a geração de planilhas e gráficos em diferentes formatos (XLS, CSV ou PDF). * O usuário pode salvar o arquivo gerado no computador ou enviar por e-mail diretamente pelo sistema. * Os filtros de busca podem ser salvos para geração automática da planilha sempre que houver atualizações nas informações de vendas. |

|  |  |
| --- | --- |
| ID: | UC05 |
| Nome: | Gerenciar Cadastro de Produtos ! |
| Escopo: | Sistema de vendas EMASA S.A. |
| Nível: | Objetivo |
| Objetivo: | Gerenciar os registros de produtos vendidos pela empresa; |
| Requisitos: | Conexão com a *internet*; Aparelho c om navegador WEB; |
| Atores: | Usuário Administrador; |
| Pré-Condições | Não se aplica |
| Pós-Condições: | As informações dos produtos serão armazenadas em um banco de dados relacional; |
| Fluxo Principal: | 1. O sistema solicita o nome do produto; 2. O usuário informa o nome do produto; 3. O sistema solicita o tipo do produto; 4. O usuário informa o tipo do produto; 5. O sistema solicita a descrição do produto; 6. O usuário informa a descrição do produto; 7. O sistema solicita o preço do produto; 8. O usuário informa o preço do produto; 9. O sistema salva as informações do produto no banco de dados. |
| Fluxos Alternativos: | Não se aplica |
| Erros/Exceções: | Não se aplica |
| Mensagens: | * Produto salvo com sucesso! * O nome do produto já existe. Por favor, informe um nome diferente.; * O preço informado é inválido. Por favor, informe um preço válido.; |
| Regras de negócio: |  |

|  |  |
| --- | --- |
| ID: | UC06 |
| Nome: | Gerenciar Acesso ao Sistema! |
| Escopo: | Sistema |
| Nível: | Objetivo |
| Objetivo: | Gerar Planilhas para análise das vendas; |
| Requisitos: | Conexão com a *internet*; Aparelho c om navegador WEB; |
| Atores: | Usuário Auxiliar de Expedição; |
| Pré-Condições: | As informações dos usuários serão armazenadas em um serviço de autenticação que utilize o protocolo OAuth; |
| Pós-Condições: |  |
| Fluxo Principal: | 1. O usuário fornece o e-mail cadastrado no serviço de OAuth; 2. O sistema verifica se o e-mail está cadastrado; 3. O usuário escolhe o tipo de permissão para acesso e clica em confirmar; 4. O sistema armazena o perfil do usuário com status pendente de aprovação pelo administrador; 5. O Administrador visualiza os perfis com status pendentes e aprova de acordo com as regras de acesso da empresa para o uso do sistema; 6. O sistema concede acesso ao usuário de acordo com as permissões aprovadas pelo administrador. |
| Fluxos Alternativos: | 1a – O usuário não tem e-mail cadastrado no serviço de Oauth  1a.1 - O usuário é redirecionado para a página do serviço de Oauth para criar um e-mail.  1a.2 - Retorna ao fluxo principal no passo 1; |
| Erros/Exceções: | Não se aplica |
| Mensagens: | * "Seu perfil foi enviado para análise do administrador"; * "Seu perfil foi aprovado e seu acesso ao sistema está liberado". |
| Regras de negócio: |  |

## Projeto de Interface

### **Wireframes**

O desenvolvimento de wireframes é uma etapa crucial no processo de criação de projetos de interface. Eles servem como uma representação visual da estrutura e do layout do sistema, ajudando a definir a arquitetura de informação e a disposição dos elementos na tela.

No projeto de interface do frontend do "Sistemas de Vendas EMASA", uma empresa de mineração, foram criados wireframes utilizando o software Concepts, conforme mostrado na Figura 14. Através desses wireframes, foi possível visualizar como seria a disposição dos elementos na tela e como o usuário iria interagir com o sistema.

Os wireframes foram desenvolvidos com base em uma análise detalhada dos requisitos do sistema e das necessidades do usuário. Foram definidas as funcionalidades que seriam apresentadas em cada tela, assim como a disposição dos elementos e a hierarquia visual de cada página.

A criação dos wireframes permitiu realizar testes de usabilidade, identificando problemas de navegação e interação, bem como oportunidades de melhoria no design da interface. As telas foram refinadas até que a estrutura final do sistema ficasse clara e intuitiva para o usuário final.

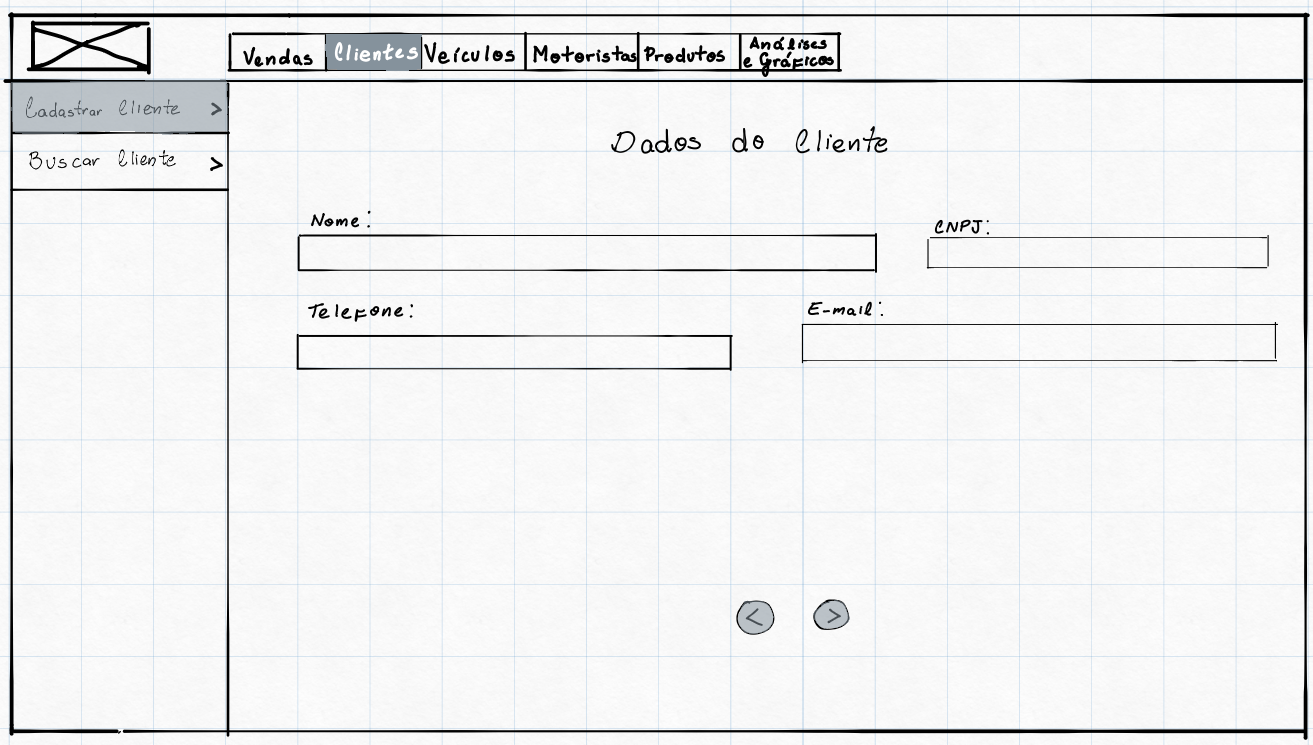


Figura 14 Wireframes para o Frontend

Todos os Wireframes desenvolvidos no projeto estão disponíveis em <https://docs.google.com/presentation/d/1eubGOBqWL-05Y1t4eV0VnKaDzt-zuzSvogoCBZsukWc/edit?usp=sharing>

### **Protótipo**

O desenvolvimento de protótipos de interfaces é uma das etapas mais importantes no processo de criação de um sistema ou aplicativo. É através dele que é possível testar a usabilidade, a navegabilidade e a funcionalidade do projeto antes mesmo de iniciar a programação.

Nesse sentido, o Figma é uma ferramenta muito utilizada para a criação de protótipos de interfaces, pois possibilita uma simulação completa da experiência do usuário, incluindo interações, animações e transições.

Neste trabalho de conclusão de curso, foi desenvolvido um protótipo de front end para o sistema de vendas de uma empresa de mineração chamada EMASA SA, o "Sistemas de Vendas EMASA". O objetivo foi criar uma interface intuitiva e funcional para facilitar o trabalho dos vendedores da empresa e, consequentemente, aumentar a eficiência e a produtividade do setor de vendas.

O protótipo foi criado no Figma e pode ser acessado através do link "<https://www.figma.com/proto/CKgX2pHuBsVimZqkP4Oym8/ERP-EMASA?page-id=0%3A1&node-id=16%3A410&viewport=-116%2C609%2C0.47&scaling=min-zoom&starting-point-node-id=16%3A410>". Nele, é possível visualizar todas as telas do sistema, bem como as interações entre elas. A partir do protótipo, foram realizados testes de usabilidade com usuários reais, o que possibilitou identificar pontos de melhoria na interface e fazer ajustes para torná-la ainda mais intuitiva e funcional.

Com o protótipo finalizado e testado, é possível iniciar a fase de desenvolvimento do sistema, com uma base sólida e segura para a criação do front end. O uso do Figma para a criação do protótipo se mostrou extremamente eficiente, pois permitiu a simulação completa da experiência do usuário, facilitando a identificação de possíveis problemas antes mesmo da programação ser iniciada.

### **Mapa do Site**

O mapa do site é uma ferramenta importante no processo de desenvolvimento de um sistema, pois ele representa de maneira clara e organizada a estrutura e a hierarquia das páginas que compõem o sistema. No caso do Sistema de Vendas EMASA, o mapa do site foi estruturado de forma a proporcionar uma experiência de uso intuitiva e eficiente para os usuários.

A tela de login é a primeira página do sistema, onde os usuários devem inserir suas credenciais para ter acesso às funcionalidades do sistema. A tela inicial do sistema apresenta o menu superior, que contém as opções de navegação para as diferentes áreas do sistema. A partir desse menu, os usuários podem acessar as páginas de Vendas, Clientes, Veículos, Motoristas, Produtos e Análises e Gráficos.

Ao clicar na opção de Vendas, os usuários têm acesso a um menu lateral, que apresenta as opções de realizar vendas e consultar vendas. Da mesma forma, ao clicar em Clientes, Veículos, Motoristas ou Produtos, os usuários podem cadastrar e buscar informações relacionadas a cada uma dessas áreas.

Por fim, a opção de Análises e Gráficos permite que os usuários acessem informações estatísticas e gráficos relacionados às vendas e aos demais aspectos do sistema, o que pode ajudar na tomada de decisão dos gestores e usuários do sistema.

Em resumo, o mapa do site do Sistema de Vendas EMASA, conforme a Figura 15, foi estruturado de forma a proporcionar aos usuários uma experiência de uso intuitiva e eficiente, garantindo acesso fácil e rápido às informações e funcionalidades do sistema.

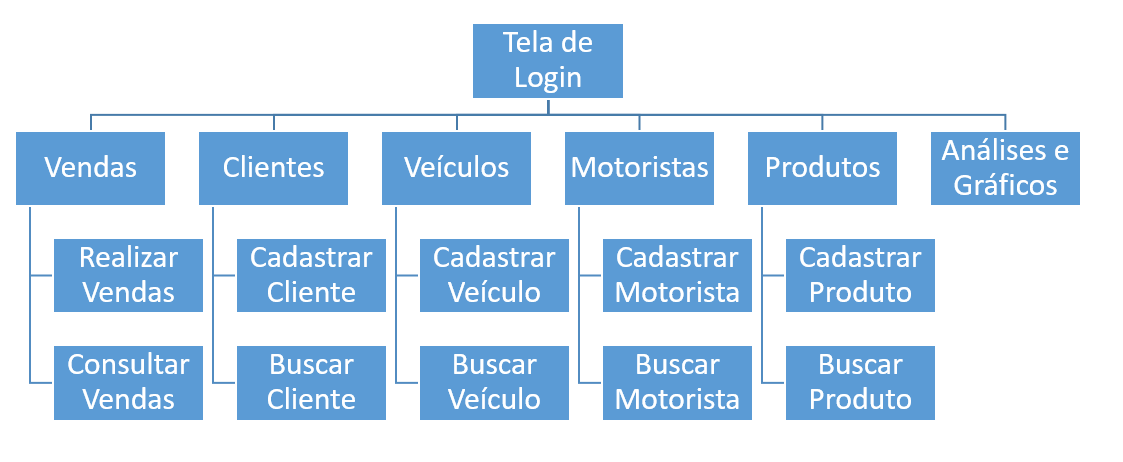
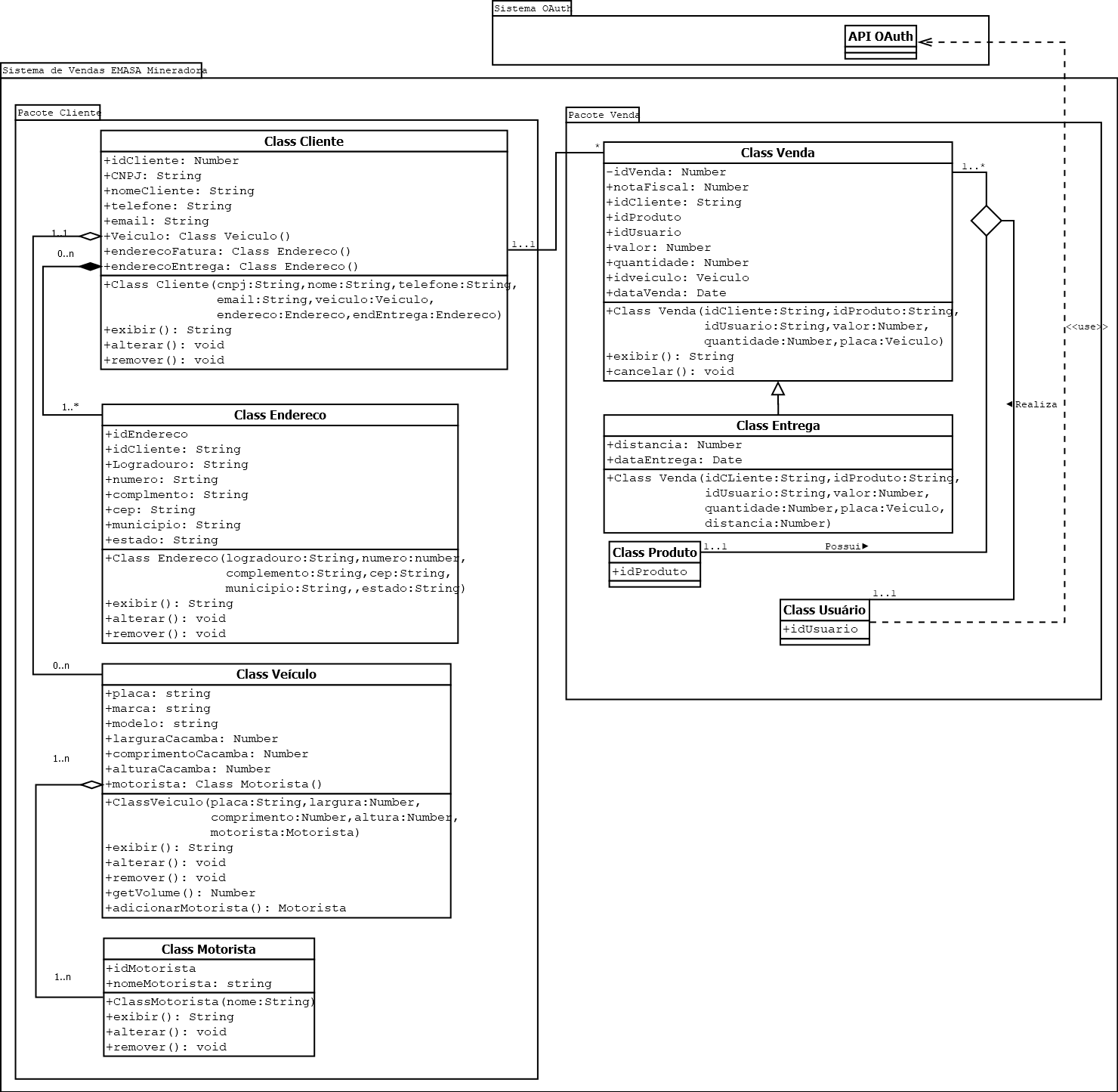


Figura 15 - Mapa do site

## Diagrama de classes



# Projeto

## Arquitetura de *software*

Este trabalho de conclusão de curso apresenta o projeto de uma aplicação web, cuja arquitetura foi cuidadosamente planejada com base em condutores arquiteturais de software, tais como usabilidade, integridade e manutenibilidade.

Para o Front-End, foi escolhido o estilo arquitetural SPA (Single Page Application), que permite uma experiência de usuário mais fluida e interativa. Já para o Back-End, optou-se pelo modelo Cliente-Servidor, utilizando Node.js e Express.js como tecnologias para a construção dos componentes do servidor.

O diagrama da arquitetura da solução foi modelado utilizando o modelo C4, conforme a Figura 16, e conta com os seguintes componentes planejados para a aplicação Web: Front-End SPA, API RESTful construída com Node.js e Express.js, e Banco de Dados MySQL conectado à API RESTful através do ORM Sequelize. A relação entre esses componentes é estabelecida através de requisições HTTP, utilizando o padrão REST para a comunicação entre o Front-End e a API RESTful, e ORM para a comunicação entre a API RESTful e o Banco de Dados como mostra a Figura 17.

A escolha do Banco de Dados MySQL e do ORM Sequelize permitirá uma maior escalabilidade, garantindo a integridade e segurança dos dados da aplicação. O modelo Cliente-Servidor, por sua vez, permitirá uma maior modularidade e controle da aplicação como um todo.

Assim, este trabalho apresenta um projeto de arquitetura de software robusto e escalável, que visa atender aos condutores arquiteturais definidos e proporcionar uma experiência de usuário satisfatória, garantindo a integridade e segurança das informações.

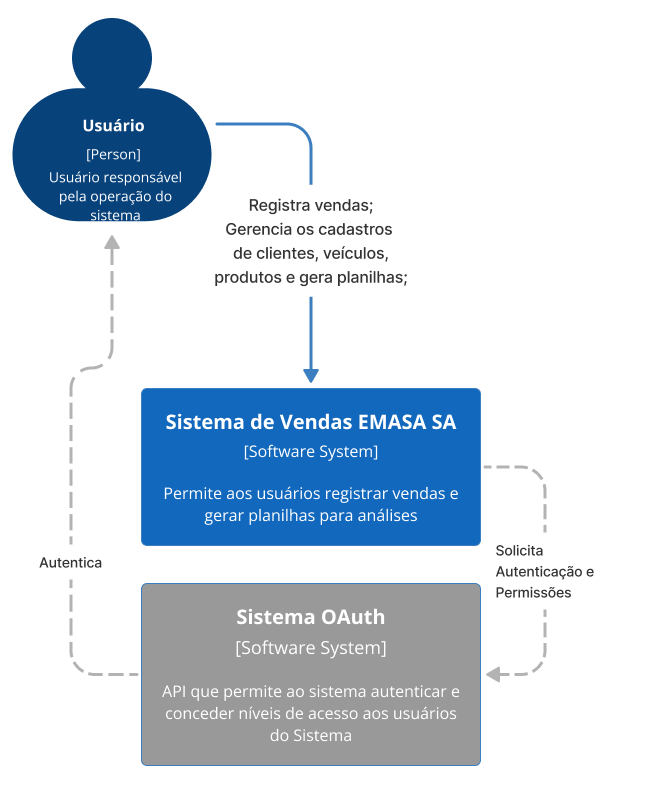


Figura 16 C4 Model Nível 1

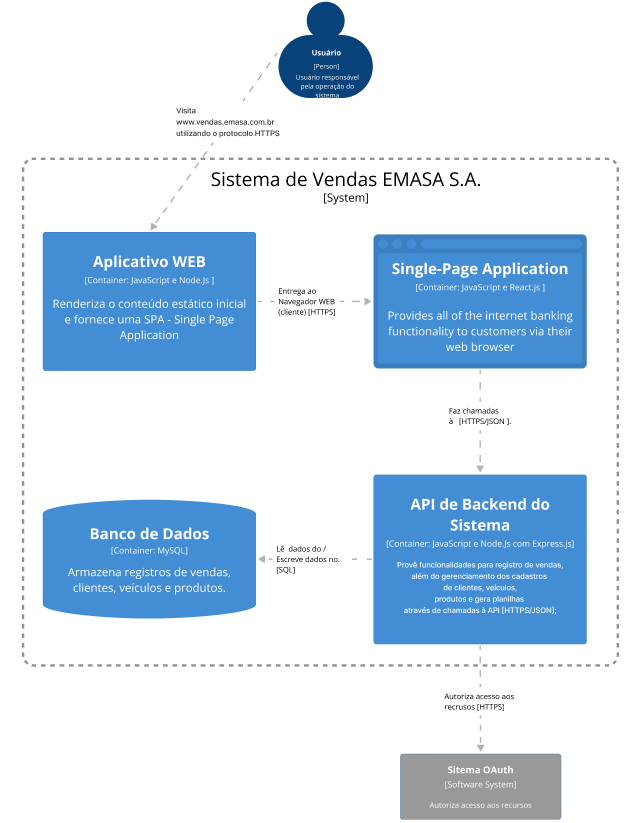


Figura 17 C4 Model Nível 2

## Arquitetura da informação

O projeto do Sistema de Vendas EMASA, desenvolvido neste trabalho, foi criado com base no conceito de "API FIRST" e com uma arquitetura de API RESTFUL, com o objetivo de criar uma interface de programação de aplicativos bem definida e que permita a integração com outras aplicações e serviços externos.

Para especificar e documentar a API, foi utilizada a ferramenta open source Swagger, que possibilitou a criação de uma documentação clara e precisa da API. Isso contribui para uma melhor comunicação entre as equipes de desenvolvimento e usuários, garantindo uma maior transparência e qualidade do projeto como um todo.

A adoção do conceito de "API FIRST" e da arquitetura RESTFUL tem como objetivo tornar a aplicação mais flexível e escalável, permitindo que novas funcionalidades e integrações possam ser adicionadas de forma mais simples e eficiente.

A utilização do Swagger para documentação da API contribui para uma melhor compreensão e uso da API por parte dos desenvolvedores e usuários, o que possibilita uma maior eficiência na integração com outras aplicações e serviços externos.

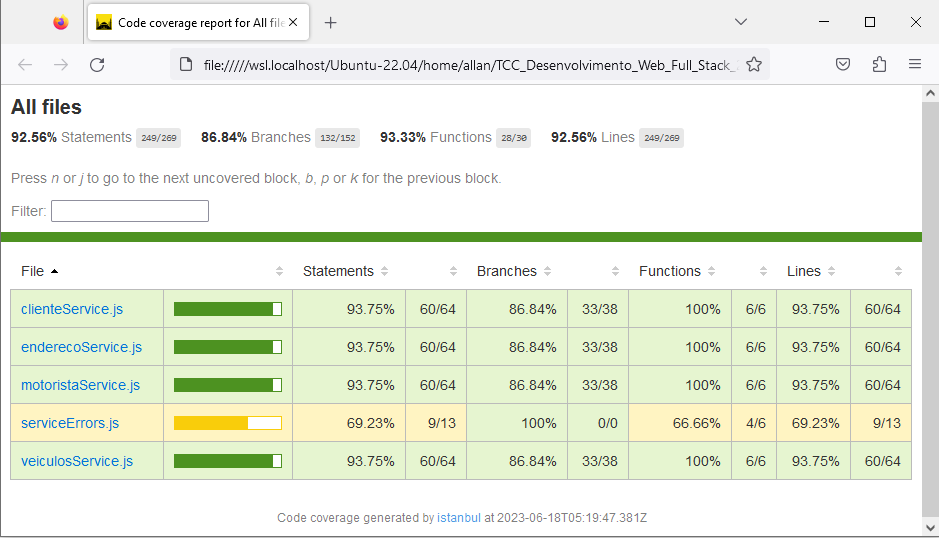
Em resumo, a adoção do conceito de "API FIRST" e da arquitetura RESTFUL, aliada à utilização do Swagger para documentação da API, são importantes medidas que foram tomadas no projeto do Sistema de Vendas EMASA para garantir a flexibilidade, escalabilidade e qualidade do projeto como um todo.

Para testar e validar a API, é possível utilizar um MOCK da API disponível em [https://app.swaggerhub.com/apis/allancac/vendas\_emasa/1.0.0](https://app.swaggerhub.com/apis/allancac/vendas_emasa/1.0.0%20) . Neste link, é possível visualizar a especificação da API no formato YAML, seguindo as especificações da Open API 3.0.0.

Também é possível realizar requisições HTTP para testes através do link do servidor virtual abaixo:

[https://virtserver.swaggerhub.com/allancac/vendas\_emasa/1.0.0./](https://virtserver.swaggerhub.com/allancac/vendas_emasa/1.0.0" \t "_new)

# Testes



# URLs

|  |  |
| --- | --- |
| **Wireframes**  (abrir no Google Chrome) | <https://docs.google.com/presentation/d/1eubGOBqWL-05Y1t4eV0VnKaDzt-zuzSvogoCBZsukWc/edit?usp=sharing> |
| **Protótipo Front End** | <https://www.figma.com/proto/CKgX2pHuBsVimZqkP4Oym8/ERP-EMASA?page-id=0%3A1&node-id=16%3A410&viewport=-116%2C609%2C0.47&scaling=min-zoom&starting-point-node-id=16%3A410> |
| **Mock da API Restful** | [https://app.swaggerhub.com/apis/allancac/vendas\_emasa/1.0.0](https://app.swaggerhub.com/apis/allancac/vendas_emasa/1.0.0%20) |
| **Servidor Virtual da API(Versão Mock)** | [https://virtserver.swaggerhub.com/allancac/vendas\_emasa/1.0.0./](https://virtserver.swaggerhub.com/allancac/vendas_emasa/1.0.0) |

## Aplicação web

|  |  |
| --- | --- |
| **Link** | <https://apptccpucminas.azurewebsites.net/> |
| **Usuário de teste** | admin@admin.com |
| **Senha** | pucminas21@ |

## Repositório código-fonte

A versão final do código está publicada em um repositório público no GitHub, na branch ‘main’. Todavia, é possível acompanhar a evolução das sprints (módulos) nos branchs ‘moduloA’, ‘moduloB’ e ‘moduloC’.

|  |  |
| --- | --- |
| **Link da Versão final – Branch ‘main’** | <https://github.com/allancac/TCC_Desenvolvimento_Web_Full_Stack_2021/> |
| **Link do modulo A – Branch ‘moduloA’** | <https://github.com/allancac/TCC_Desenvolvimento_Web_Full_Stack_2021/tree/moduloA> |
| **Link do modulo B – Branch ‘moduloB’** | <https://github.com/allancac/TCC_Desenvolvimento_Web_Full_Stack_2021/tree/moduloB> |
| **Link do modulo C – Branch ‘moduloC’** | <https://github.com/allancac/TCC_Desenvolvimento_Web_Full_Stack_2021/tree/moduloC> |

## Vídeo de apresentação do trabalho

|  |  |
| --- | --- |
| **Link** | <https://youtu.be/1RA71Q4Yb_w> |

# REFERÊNCIAS

Aguiar, Fábio, e Paulo Caroli. *Product Backlog Building: Um guia prático para criação e refinamento de backlog para produtos de sucesso.* Rio de Janeiro: Editora Caroli, 2021.

COHN, M. *User Stories Applied: For Agile Software Development.* Boston: Addison-Wesley, 2004.

Date, C J. *Introdução a Sistemas de Bancos de Dados. 8. ed.* Rio de Janeiro: Elsevier, 2004.

Lowe, d. *Windows 98 in a Nutshell: A Desktop Quick Reference. 1st ed.* Sebastopol: O'Reilly Media, 1999.

MAURYA, A. *Running Lean: Iterate from Plan A to a Plan That Works.* Sebastopol: O'Reilly Media, 2012.

MYERS, B A, e M B ROSSON. “Survey on User Interface Programming.” *Proceedings of the ACM SIGGRAPH Symposium on User Interface Software and Technology (UIST)*, 1992: 195-204.

Prestwood, M. *Clipper 5 Development Guide. 2. ed.* San Francisco: Sybex, 1993.

Rupp, N A. *Programming: A Beginner's Guide to XBase.* New York: McGraw-Hill, 1991.