

**CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE FRANCA
“Dr. THOMAZ NOVELINO”**

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

**GUSTAVO HENRIQUE MARTINS PEREIRA
MATEUS PEREIRA CINTRA SILVA**

GM SOLUÇÕES

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Fausto Gonçalves Cintra

FRANCA/SP

2021

GM SOLUÇÕES

Gustavo Henrique Martins Pereira¹

Mateus Pereira Cintra Silva²

Resumo

Este sistema *Enterprise Resource Planning* (ERP) tem como propósito atender as necessidades básicas de gestão de pequenas empresas, proporcionando um auxílio no gerenciamento da empresa e a modernizando-a, sendo fornecido gratuitamente em uma plataforma *web*. Levando isso em conta, durante o levantamento de requisitos, priorizamos as principais funcionalidades que uma pequena empresa irá fazer uso, as quais são citadas na documentação apresentada. O desenvolvimento consistiu na utilização de diversas ferramentas, dentre as principais: Node.js no *back-end*, Bootstrap como *framework* no *front-end* e PostgreSQL como banco de dados. Assim, teve como resultado um Produto Mínimo Viável, atendendo as necessidades básicas das pequenas empresas, deixando espaço para aprimoramento de novas funcionalidades.

Palavras-chave: Análise e Desenvolvimento de Sistemas. Engenharia de *software*. Pequenas empresas. Planejamento de Recursos Empresariais. Sistema de gestão empresarial.

Abstract

This Enterprise Resource Planning (ERP) system aims to meet the basic management needs of small businesses, aiding in managing the company and modernizing it, being provided free of charge on a web platform. Taking this into account, during the requirements survey, we prioritize the main features that a small business will use, which are mentioned in the documentation presented. The development consisted in the use of several tools, among the main ones: Node.js in the back-end, Bootstrap as a framework in the front-end and PostgreSQL as a database. Thus, it resulted in a Minimum Viable Product, meeting the basic needs of small businesses, leaving room for improvement of new features.

Keywords: *Business management system. Enterprise Resource Planning. Small businesses. Software engineering. Systems analysis and development.*

¹ Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr. Thomaz Novelino – Franca/SP. Endereço eletrônico: gustavohenrique18031998@gmail.com

² Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr. Thomaz Novelino – Franca/SP. Endereço eletrônico: mateus_cintra@hotmail.com

1 Introdução

O termo ERP vem de uma sigla em inglês, que significa *Enterprise Resource Planning*, ou, em português, Planejamento dos Recursos Empresariais. Seu propósito é auxiliar os processos de uma empresa, facilitando e melhorando sua gestão, controle, e eficiência, entre outros aspectos.

Os ERP crescentemente vêm fazendo parte do cotidiano de várias empresas. No entanto, muitas das vezes um sistema desses tem consideráveis custos de utilização e manutenção, os quais grandes e médias empresas conseguem sustentar. Para um pequeno negócio, por outro lado, manter um sistema desses pode ser inviável ou até impossível, mesmo com todos seus benefícios.

Pensando nesta questão, concebemos este projeto, que visa fornecer um ERP que atenda às necessidades de uma pequena empresa. Com essa ideia em mente, levantamos as principais funcionalidades que uma empresa desse porte necessitaria. Concluímos que o sistema deveria contar com uma página dedicada a cada um dos seguintes itens: vendas, gerenciamento de clientes, produtos, pedidos e estoque. Além delas, também um *dashboard*, contendo informações processadas e exibidas de forma a facilitar a compreensão do conteúdo, tudo isso desenvolvido em uma plataforma *web*, de forma a ter uma maior aceitabilidade dos usuários.

Com isso, durante o desenvolvimento fomos aplicando essas características ao sistema, iniciando pelo back-end onde implementamos o banco de dados no qual armazenamos as devidas informações, que seria necessário no projeto, para realizar esta função utilizamos o PostgreSQL. Referente ao front-end dedicando uma página para cada tópico deixando todos os campos de entrada de dados necessário para o registro, com exibição das informações em uma tabela para facilitar a visualização, edição, exclusão e para implementar tal elemento foi utilizado o Bootstrap um framework para o front-end que facilita a criação das páginas html. Para o dashboard foi decidido que as informações seriam mostradas de forma gráfica para facilitar a compreensão das informações e sua visualização, assim utilizamos o Charts.js uma API que fornece gráficos responsivos.

2 Levantamento de Requisitos

O levantamento de requisitos em uma empresa é muito válido, pois eles

correspondem ao ponto de partida do desenvolvimento. A primeira etapa é determinar quais as funcionalidades e o escopo que serão executadas.

Um dos fatores que dificulta a comunicação entre o cliente e o analista é a falta de compreensão do que o cliente realmente necessita. Cabendo ao analista saber interpretar e identificar o real problema. Muitas vezes, o ponto de vista muda conforme o usuário. Sendo assim, é necessário colher dados e informações de todas as pessoas que utilizarão o produto na empresa.

O levantamento de requisitos é a etapa mais importante para o desenvolvimento de um sistema, pois é a base sobre a qual tudo o mais será construído. Todas as lacunas do desenvolvimento devem ser preenchidas de forma cuidadosa e consistente.

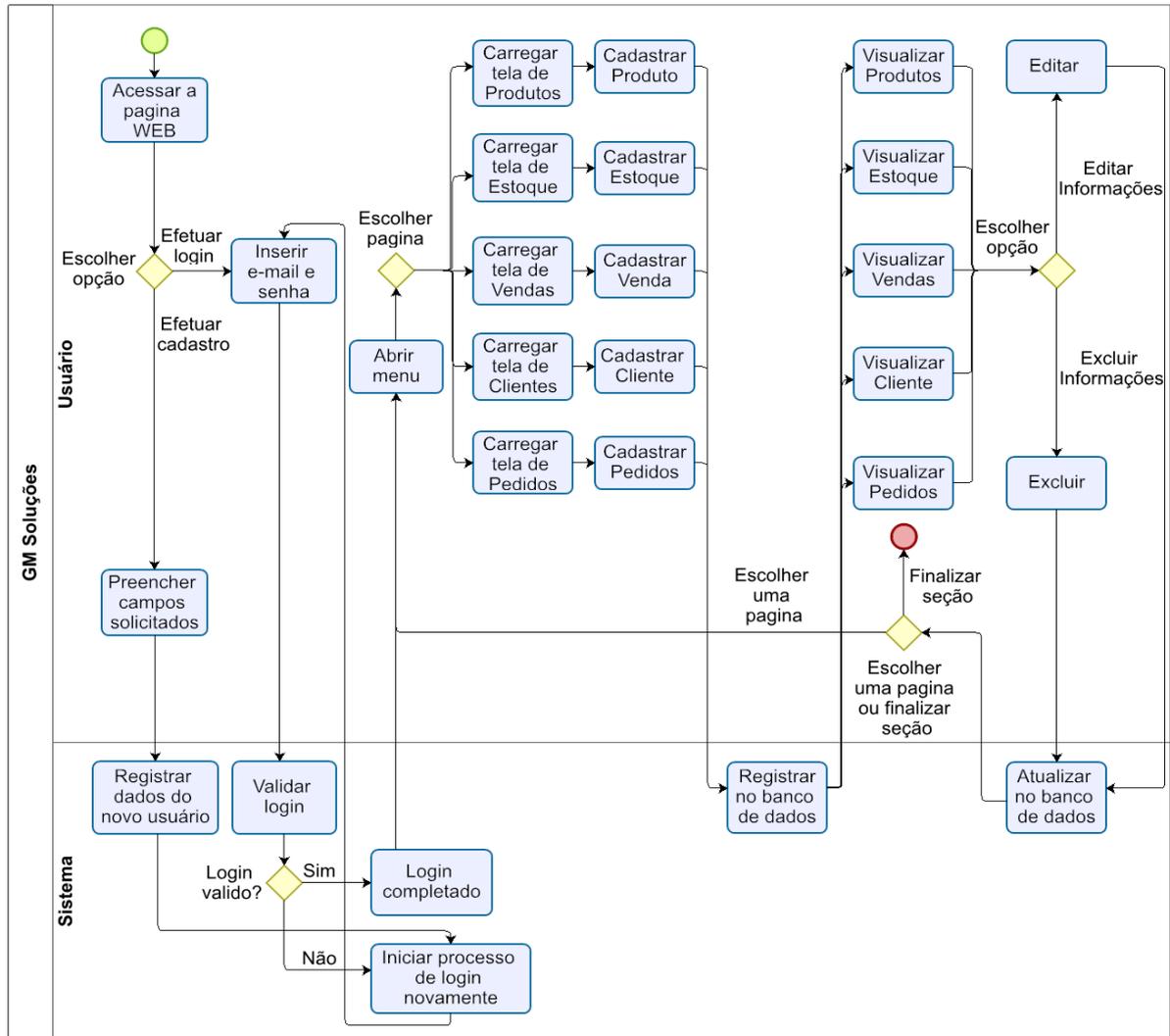
2.1 Elicitação e especificação de requisitos

O levantamento dos requisitos teve como foco priorizar elementos essenciais para um pequeno negócio. Tendo em vista este ponto, consideramos as ações de maior relevância na parte de elicitação dos requisitos, como registro de vendas, pedidos, produtos, clientes e estoque. Outro ponto que influenciou no levantamento foi a aceitabilidade, um dos atributos essenciais de um bom *software*, como citado por Sommerville (2011, p. 5): “o *software* deve ser aceitável para o tipo de usuário para o qual foi projetado. Isso significa que deve ser compreensível, usável e compatível com outros sistemas usados por ele”. Assim, priorizamos a escolha de uma plataforma *web* para receber o sistema, a qual, por ser de fácil acesso, facilitaria que os usuários a utilizarem, sendo apenas requerido acesso à Internet e um navegador *web* atualizado, com possibilidade de adaptação para funcionamento *offline*.

3 BPMN

O *Business Process Model and Notation* (BPMN) é um diagrama formado por ícones que exibem um fluxo de processos, informando o modo como irão funcionar, passo a passo. O principal objetivo do BPMN é a modelagem dos processos. A Figura 1 exibe o BPMN desenvolvido para o sistema proposto.

Figura 1– Diagrama BPMN



Fonte: Elaborada pelos autores

4 Requisitos funcionais e não funcionais

O Quadro 1 apresenta os requisitos funcionais, que servem para mostrar as regras que o sistema obedece, tendo em vista as funções que ele exerce.

Quadro 1 – Requisitos Funcionais do sistema

RF001 -Cadastro de clientes	Categoria: () Oculto (X)Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O usuário deverá cadastrar os clientes e o sistema deverá registrar todas as informações no banco de dados.		
RF002 -Cadastro de produtos	Categoria: () Oculto	Prioridade: (X) Altíssima

	(X) Evidente	() Alta () Média () Baixa
Descrição: O usuário deverá cadastrar os produtos e o sistema deverá registrar todas as informações no banco de dados.		
RF003- Controle de estoque	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O usuário deverá cadastrar os produtos que deseja colocar em estoque e o sistema deverá registrar todas as informações no banco de dados.		
RF004- Cadastrar vendas	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O usuário deverá cadastrar as vendas selecionando o cliente referido na venda e o sistema deverá registrar todas as informações no banco de dados.		
RF005- Cadastrar Pedidos	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deve cadastrar os pedidos selecionando a venda e o produto referentes ao pedido e o sistema deverá registrar todas as informações no banco de dados.		
RF006- Editar	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá atualizar todas as informações no banco de dados caso o usuário necessite editar as informações.		
RF007- Excluir	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deverá excluir todas as informações no banco de dados caso o usuário necessite excluir as informações.		

O Quadro 2 representa os requisitos não funcionais, que são os requisitos relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas.

Quadro 2 – Requisitos não funcionais do sistema

RNF001- Framework	O <i>framework</i> a ser utilizado será o Bootstrap.	Tipo: Usabilidade	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RNF002- Exibição de dados	Todos os dados do banco de dados terão de ser exibidos em formato de tabela.	Tipo: Visual	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RNF003- Dashboard	A tela de <i>dashboard</i> deverá exibir informações gerais	Tipo: Visual	(X) Desejável () Obrigatório	(X) Permanente () Transitório

	simplificadas em formato de gráficos.			
RNF004-Notificação	Após efetuar alguma ação (cadastro, exclusão e edição) o sistema deverá exibir uma notificação de ação concluída e outra em casos de falha.	Tipo: Visual	(X) Desejável () Obrigatório	(X) Permanente () Transitório
RNF005-Validação	É previsto que todos os <i>inputs</i> tenham uma validação dos dados inseridos.	Tipo: Usabilidade	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RNF006-Entrada de dados	Todos os campos que necessitarem de dados já existentes no banco de dados, deverão ser exibidos por um <i>dropdown</i> com apenas opções tiradas do próprio banco.	Tipo: Usabilidade	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RNF006-Plataforma WEB	O sistema deverá ser desenvolvido em uma plataforma <i>web</i>	Tipo: Plataforma	() Desejável (X) Obrigatório	(X) Permanente () Transitório

Matriz de rastreabilidade entre requisitos funcionais:

	RF01	RF02	RF03	RF04	RF05	RF06	RF07
RF01	x			x		x	x
RF02		x	x		x	x	x
RF03		x	x			x	x
RF04	x			x	x	x	x
RF05		x		x	x	x	x
RF06						x	
RF07							x

Matriz de Rastreabilidade entre Requisitos Funcionais e Requisitos Não Funcionais:

	RF01	RF02	RF03	RF04	RF05	RF06	RF07
RNF01							
RNF02	X	X	X	X	X	X	X
RNF03	X	X	X	X	X		
RNF04	X	X	X	X	X	X	X
RNF05	X	X	X	X	X		
RNF06			X		X		
RNF07							

5 Casos de Uso

Os casos de uso descrevem os diversos passos entre os atores e o sistema na forma de um diálogo entre eles. Sendo assim, uma unidade de um trabalho significativa. Por exemplo: o "Visualizar Produto", "Cadastrar Produto" e "Editar Produto" são todos casos de uso.

Índice de casos de Uso:

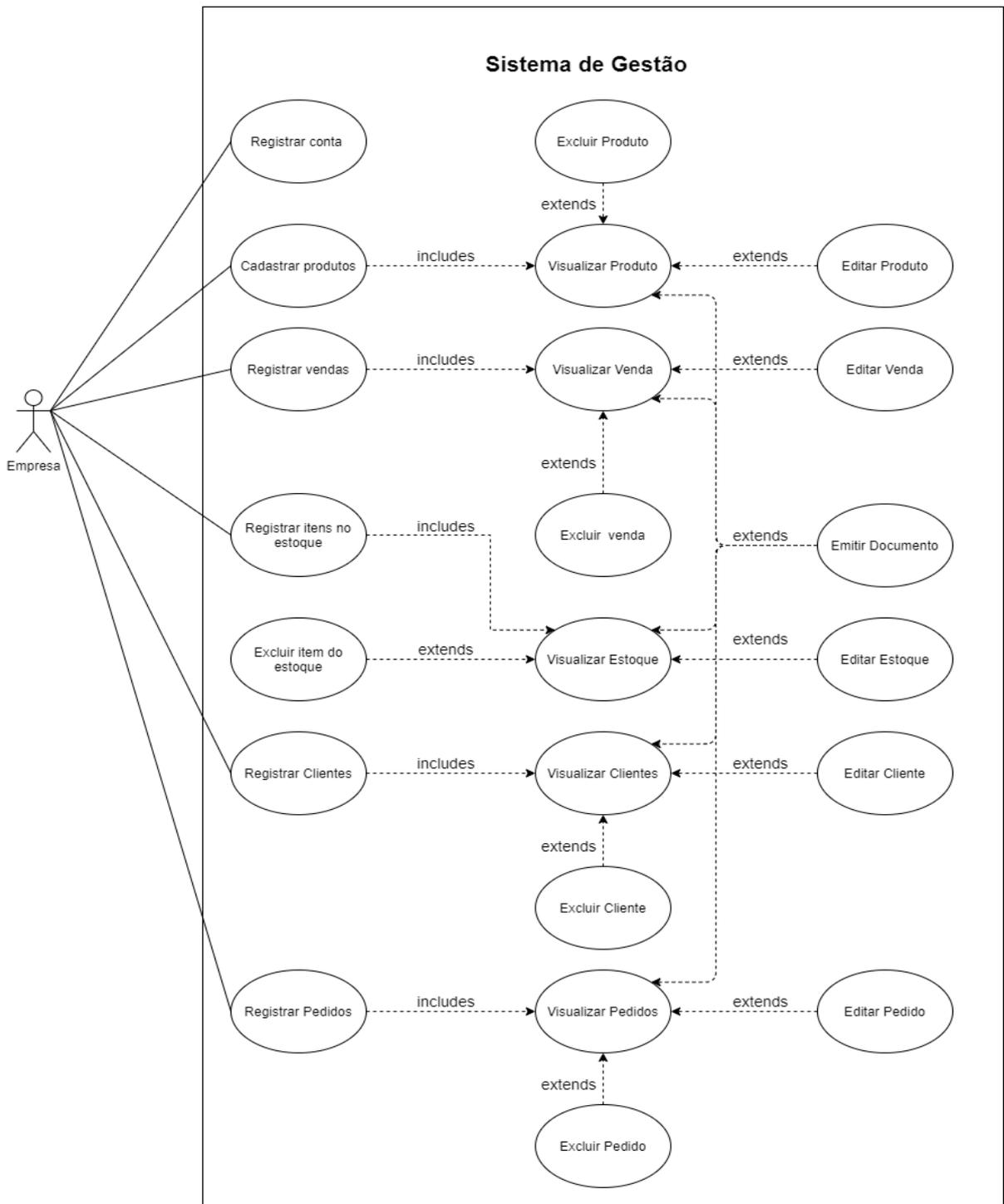
- UC - 001 Registrar conta
- UC - 002 Emitir documento
- UC - 003 Visualizar Produto
- UC - 004 Registrar Produto
- UC - 005 Editar Produto
- UC - 006 Excluir Produto
- UC - 007 Registrar Pedido
- UC - 008 Visualizar Pedido
- UC - 009 Editar Pedidos
- UC - 010 Excluir Pedido
- UC - 011 Registrar Venda
- UC - 012 Visualizar Venda
- UC - 013 Editar Venda
- UC - 014 Excluir Venda
- UC - 015 Registrar itens no estoque
- UC - 016 Visualizar Estoque
- UC - 017 Editar Estoque
- UC - 018 Excluir item do Estoque
- UC - 019 Registrar Cliente
- UC - 020 Visualizar Cliente
- UC - 021 Editar Cliente
- UC - 022 Excluir Cliente

Indicação dos autores do sistema:

- Empresa: é o usuário que faz todas as funcionalidades do sistema.

A Figura 2 exibe o diagrama dos casos de uso.

Figura 2– Diagrama de Caso de Uso



Fonte: Elaborada pelos autores

Especificação dos casos de uso:

Caso de Uso – Registrar conta	
ID	UC - 001
Descrição	O usuário irá se registrar no sistema para poder ter acesso a ele.
Ator Primário	Usuário do sistema
Pré-condição	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário clicar para se registrar. 2. O usuário irá preencher os campos requeridos e finalizar o cadastro. 3. O sistema irá validar os dados e fornecer acesso ao usuário com os dados do cadastro quando fizer <i>login</i>.
Pós-condição	Nenhuma
Cenário Alternativo	*a – Em qualquer momento o usuário pode sair do sistema.
Inclusão	
Extensão	

Caso de Uso – Emitir documento	
ID	UC – 002
Descrição	O sistema irá criar um arquivo com as informações presentes na tabela
Ator Primário	Usuário do sistema
Pré-condição	UC - 003 Visualizar Produto, UC - 008 Visualizar Pedido, UC - 012 Visualizar Venda, UC - 016 Visualizar Estoque, UC - 020 Visualizar Cliente
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário clicar na opção de emitir documentos. 2. O sistema irá criar um arquivo contendo todos os dados visíveis
Pós-condição	Nenhuma
Cenário Alternativo	*a – Em qualquer momento o usuário pode sair do sistema.
Inclusão	
Extensão	

Caso de Uso – Visualizar Produto	
ID	UC - 003
Descrição	Deixar todos os dados de produtos atualizados visíveis em tabelas
Ator Primário	Usuário do sistema
Pré-condição	UC - 004 Registrar Produto

Cenário Principal	<ol style="list-style-type: none"> 1. O use case inicia quando o usuário clica na página do produto. 2. O sistema irá exibir todos os dados registrados em formato de tabela. 3. O sistema sempre irá mostrar dados atualizados.
Pós-condição	Nenhuma
Cenário Alternativo	<p>*a – Em qualquer momento o usuário pode sair do sistema.</p> <p>2.a Não conter nenhum dado no banco de dados. Como consequência não será exibido nada.</p> <p>3.a Após efetuar alguma ação de edição ou exclusão o sistema irá atualizar os dados exibidos.</p>
Inclusão	
Extensão	

Caso de Uso – Registrar Produto	
ID	UC - 004
Descrição	Este caso de uso tem por objetivo o usuário fazer o cadastro no sistema e utilizá-lo após isso.
Ator Primário	Usuário do sistema Administrador
Pré-condição	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> 1. O use case inicia quando o usuário seleciona a opção cadastro. 2. O sistema carrega as informações necessárias para realizar essa operação. 3. O sistema recebe os dados e verifica a integridade dos dados no banco. 4. O sistema cadastra o usuário com sucesso. 5. O sistema encerra a operação.
Pós-condição	UC - 003 Visualizar Produto
Cenário Alternativo	<p>*a – Em qualquer momento o usuário pode sair do sistema.</p> <p>2a - Falha no cadastro.</p> <p>2b - O sistema irá pedir para que todos os campos obrigatórios sejam preenchidos.</p>
Inclusão	
Extensão	

Caso de Uso – Editar Produto	
ID	UC - 005
Descrição	Realizar uma atualização nos dados de um produto
Ator Primário	Usuário do sistema
Pré-condição	UC - 004 Registrar Produto

Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário clica na opção editar. 2. O sistema irá puxar todos os dados já existentes para os <i>inputs</i>. 3. O sistema irá registrar todos os novos dados passados.
Pós-condição	UC - 003 Visualizar Produto
Cenário Alternativo	<p>*a – Em qualquer momento o usuário pode sair do sistema.</p> <p>2.a Caso algum campo obrigatório for apagado a atualização não será efetuada.</p>
Inclusão	
Extensão	

Caso de Uso – Excluir Produto	
ID	UC - 006
Descrição	Caso seja necessário a exclusão do produto do sistema.
Ator Primário	Usuário do sistema
Pré-condição	UC - 004 Registrar Produto
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário clica na opção excluir. 2. O sistema irá criar uma notificação quando o produto for excluído. 3. O sistema irá deletar toda informação sobre o produto.
Pós-condição	UC - 003 Visualizar Produto
Cenário Alternativo	<p>*a – Em qualquer momento o usuário pode sair do sistema.</p> <p>3.a Caso o produto esteja registrado em estoque não será permitido a exclusão.</p>
Inclusão	
Extensão	

Caso de Uso – Registrar Pedido	
ID	UC - 007
Descrição	Este caso de uso tem por objetivo o usuário fazer o cadastro no sistema e utilizá-lo após isso.
Ator Primário	Usuário do sistema, Administrador
Pré-condição	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário seleciona a opção cadastro. 2. O sistema carrega as informações necessárias para realizar essa operação. 3. O sistema recebe os dados e verifica a integridade dos dados no banco. 4. O sistema cadastra o usuário com sucesso. 5. O sistema encerra a operação.
Pós-condição	UC - 008 Visualizar Pedido

Cenário Alternativo	*a – Em qualquer momento o usuário pode sair do sistema. 2a - Falha no cadastro. 2.a 1 O sistema irá pedir para que todos os campos obrigatórios sejam preenchidos.
Inclusão	
Extensão	

Caso de Uso – Visualizar Pedido	
ID	UC - 008
Descrição	Deixar todos os dados de pedidos atualizados visíveis em tabelas
Ator Primário	Usuário do sistema
Pré-condição	UC - 004 Registrar Pedido
Cenário Principal	1. O use case inicia quando o usuário clica na página pedido. 2. O sistema irá exibir todos os dados registrados em formato de tabela. 3. O sistema sempre irá mostrar dados atualizados.
Pós-condição	Nenhuma
Cenário Alternativo	*a – Em qualquer momento o usuário pode sair do sistema. 2.a Não conter nenhum dado no banco de dados. Com consequência não será exibido nada. 3.a Após efetuar alguma ação de edição ou exclusão o sistema irá atualizar os dados exibidos.
Inclusão	
Extensão	

Caso de Uso – Editar Pedido	
ID	UC – 009
Descrição	Realizar uma atualização nos dados de um pedido
Ator Primário	Usuário do sistema
Pré-condição	UC - 007 Registrar Pedido
Cenário Principal	1. O use case inicia quando o usuário clica na opção editar. 2. O sistema irá puxar todos os dados já existentes para os inputs. 3. O sistema irá registrar todos os novos dados passados.
Pós-condição	UC - 008 Visualizar Pedido
Cenário Alternativo	*a – Em qualquer momento o usuário pode sair do sistema. 2.a Caso algum campo obrigatório for apagado a atualização não será efetuada.
Inclusão	
Extensão	

Caso de Uso – Excluir Pedido	
ID	UC – 010
Descrição	Caso seja necessário a exclusão do pedido do sistema
Ator Primário	Usuário do sistema
Pré-condição	UC - 007 Registrar Pedido
Cenário Principal	<ol style="list-style-type: none"> 1. O use case inicia quando o usuário clica na opção excluir. 2. O sistema irá criar uma notificação quando o pedido for excluído. 3. O sistema irá deletar toda informação sobre o pedido.
Pós-condição	UC - 008 Visualizar Pedido
Cenário Alternativo	*a – Em qualquer momento o usuário pode sair do sistema.
Inclusão	
Extensão	

Caso de Uso – Registrar venda	
ID	UC - 011
Descrição	Este caso de uso tem por objetivo o usuário fazer o cadastro no sistema e utilizá-lo após isso.
Ator Primário	Usuário do sistema Administrador
Pré-condição	UC – 019
Cenário Principal	<ol style="list-style-type: none"> 1. O use case inicia quando o usuário seleciona a opção cadastro. 2. O sistema carrega as informações necessárias para realizar essa operação. 3. O sistema recebe os dados e verifica a integridade dos dados no banco. 4. O sistema cadastra o usuário com sucesso. 5. O sistema encerra a operação.
Pós-condição	UC - 012 - Visualizar venda
Cenário Alternativo	*a – Em qualquer momento o usuário pode sair do sistema. 2a - Falha no cadastro. 2b -O sistema irá pedir para que todos os campos obrigatórios sejam preenchidos.
Inclusão	
Extensão	

Caso de Uso – Visualizar venda	
ID	UC – 012
Descrição	Deixar todos os dados de vendas atualizados visíveis em tabelas.
Ator Primário	Usuário do sistema

Pré-condição	UC - 011 Registrar venda
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário clica na página de venda. 2. O sistema irá exibir todos os dados registrados em formato de tabela. 3. O sistema sempre irá mostrar dados atualizados.
Pós-condição	Nenhuma
Cenário Alternativo	<p>*a – Em qualquer momento o usuário pode sair do sistema.</p> <p>2.a Não conter nenhum dado no banco de dados. Com consequência não será exibido nada.</p> <p>3.a Após efetuar alguma ação de edição ou exclusão o sistema irá atualizar os dados exibidos.</p>
Inclusão	
Extensão	

Caso de Uso – Editar venda	
ID	UC – 013
Descrição	Realizar uma atualização nos dados de uma venda
Ator Primário	Usuário do sistema
Pré-condição	UC - 011 Registrar Pedido
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário clica na opção editar. 2. O sistema irá puxar todos os dados já existentes para os <i>inputs</i>. 3. O sistema irá registrar todos os novos dados passados.
Pós-condição	UC - 011 Visualizar Pedido
Cenário Alternativo	<p>*a – Em qualquer momento o usuário pode sair do sistema.</p> <p>2.a Caso algum campo obrigatório for apagado a atualização não será efetuada.</p>
Inclusão	
Extensão	

Caso de Uso – Excluir venda	
ID	UC – 014
Descrição	Caso seja necessário a exclusão da venda do sistema.
Ator Primário	Usuário do sistema, Administrador.
Pré-condição	UC - 011 Registrar venda
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário clica na opção excluir. 2. O sistema irá criar uma notificação quando o pedido for excluído. 3. O sistema irá deletar toda informação sobre o pedido.
Pós-condição	UC - 012 Visualizar Pedido
Cenário Alternativo	*a – Em qualquer momento o usuário pode sair do sistema.

Inclusão	
Extensão	

Caso de uso - Registrar estoque	
ID	UC – 015
Descrição	Este caso de uso tem como objetivo realizar a inserção dos produtos dentro do estoque.
Ator Primário	Usuário do sistema Administrador
Pré-condição	UC - 004 - Registrar produto
Cenário Principal	<ol style="list-style-type: none"> 1. O use case tem início quando é cadastrado o produto no sistema UC - 004, sendo assim. O registro do produto no estoque estará disponível, será solicitado a quantidade de produtos que entraram no estoque. 2. O sistema carrega as informações necessárias para realizar essa operação. 3. O sistema recebe os dados e verificar a integridade dos dados no banco. 4. O sistema cadastra o produto com sucesso. 5. O sistema encerra a operação.
Pós-condição	Sem condição
Cenário Alternativo	<p>*a – Em qualquer momento o usuário pode sair do sistema</p> <p>2a - Falha no cadastro.</p> <p>2b - O sistema irá pedir para que todos os dados obrigatórios sejam fornecidos de maneira correta.</p>
Inclusão	
Extensão	

Caso de uso - Visualizar estoque	
ID	UC – 016
Descrição	Este caso de uso tem como objetivo realizar apenas a visualização dos dados de estoque.
Ator Primário	Usuário do sistema, Administrador
Pré-condição	UC - 14 - Registrar estoque
Cenário Principal	<ol style="list-style-type: none"> 1. O use case tem início quando o usuário do sistema precisa verificar o que está em registro no estoque, impossibilitando realizar qualquer modificação.
Pós-condição	Sem condição

Cenário Alternativo	*a – Em qualquer momento o usuário pode sair do sistema.
Inclusão	
Extensão	

Caso de uso - Editar estoque	
ID	UC – 017
Descrição	Este caso de uso tem como objetivo realizar alguma edição no cadastro do estoque.
Ator Primário	Usuário do sistema
Pré-condição	UC 4 - Registrar Produto
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário clica na opção editar. 2. O sistema irá puxar todos os dados já existentes para os inputs. 3. O sistema irá registrar todos os novos dados.
Pós-condição	Sem condição
Cenário Alternativo	*a – Em qualquer momento o usuário pode sair do sistema
Inclusão	
Extensão	

Caso de Uso –Excluir item estoque	
ID	UC - 018
Descrição	Este caso de uso tem como objetivo excluir os produtos dentro do estoque.
Ator Primário	Usuário do sistema
Pré-condição	Excluir produto do estoque apenas quando for retirado do mesmo. Apenas sendo com alguma compra.
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário clicar no botão de excluir 2. Excluir produtos do estoque quando for realizado uma venda com algum cliente 3. Após realizar a exclusão, sempre checar se foi removido a quantidade correta
Pós-condição	UC - 17 - Visualizar estoque
Cenário Alternativo	*a – Em qualquer momento o usuário pode sair do sistema. 2a - Caso seja excluído item a mais que o necessário, será preciso incluir todos os itens que foram removidos de forma errada.
Inclusão	
Extensão	

Caso de Uso – Registrar Cliente	
ID	UC - 019
Descrição	Este caso de uso tem por objetivo o usuário fazer o cadastro no sistema e utilizá-lo após isso.
Ator Primário	Usuário do sistema Administrador
Pré-condição	Ter em mãos documentos que serão necessários para o cadastro do cliente
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário seleciona a opção cadastro. 2. O sistema carrega as informações necessárias para realizar essa operação. 3. O sistema recebe os dados e verifica a integridade dos dados no banco. 4. O sistema cadastra o usuário com sucesso. 5. O sistema encerra a operação
Pós-condição	UC - 020 - Visualizar Cliente
Cenário Alternativo	<p>*a – Em qualquer momento o usuário pode sair do sistema.</p> <p>2a - Falha no cadastro</p> <p>2b -O sistema irá pedir para que todos os campos obrigatórios sejam preenchidos</p>
Inclusão	
Extensão	

Caso de Uso – Visualizar Cliente	
ID	UC – 020
Descrição	Deixar todos os dados de clientes atualizados visíveis em tabelas
Ator Primário	Usuário do sistema
Pré-condição	UC - 019 - Cadastrar Cliente
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário clica na página cliente. 2. O sistema irá exibir todos os dados registrados em formato de tabela. 3. O sistema sempre irá mostrar dados atualizados.
Pós-condição	Nenhuma
Cenário Alternativo	<p>*a – Em qualquer momento o usuário pode sair do sistema.</p> <p>2.a Não conter nenhum dado no banco de dados. Como consequência, não será exibido nada</p> <p>3.a Após efetuar alguma ação de edição ou exclusão o sistema irá atualizar os dados exibidos.</p>
Inclusão	

Extensão	
-----------------	--

Caso de Uso – Editar Cliente	
ID	UC – 021
Descrição	Realizar uma atualização nos dados de um cliente
Ator Primário	Usuário do sistema
Pré-condição	UC - 019 - Cadastrar Cliente
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário clica na opção editar. 2. O sistema irá puxar todos os dados já existentes para os <i>inputs</i>. 3. O sistema irá registrar todos os novos dados passados.
Pós-condição	UC - 020 - Visualizar Cliente
Cenário Alternativo	<p>*a – Em qualquer momento o usuário pode sair do sistema</p> <p>2.a Caso algum campo obrigatório for apagado a atualização não será efetuada.</p>
Inclusão	
Extensão	

Caso de Uso – Excluir Cliente	
ID	UC - 022 - Excluir Cliente
Descrição	Caso seja necessário a exclusão do cliente do sistema
Ator Primário	Usuário do sistema
Pré-condição	UC – 020 - Visualizar cliente
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário clica na opção excluir. 2. O sistema irá criar uma notificação quando o cliente for excluído. 3. O sistema irá deletar toda informação sobre o cliente.
Pós-condição	UC - 020 - Visualizar Cliente
Cenário Alternativo	<p>*a – Em qualquer momento o usuário pode sair do sistema.</p> <p>3.a Caso o cliente esteja registrado em alguma venda não será permitido a exclusão.</p>
Inclusão	
Extensão	

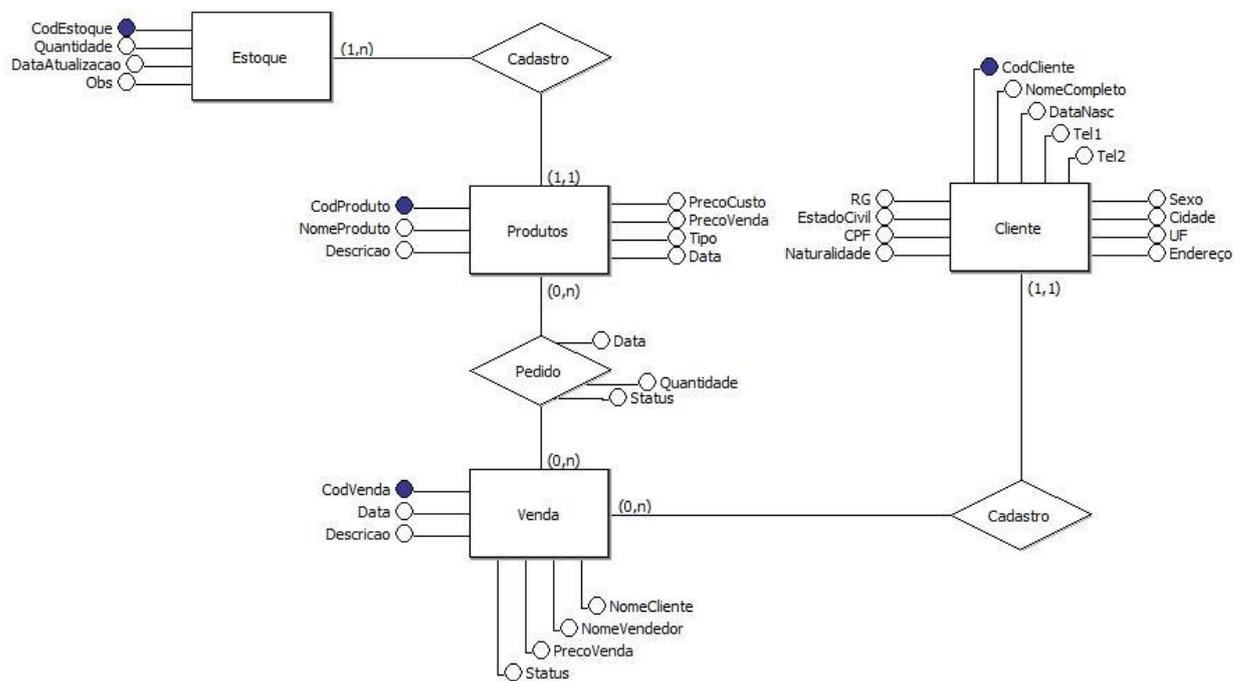
6 Diagrama Entidade-Relacionamento

Utilizando-se da representação do Diagrama Entidade-Relacionamento (DER), serão mostrados as entidades, seus atributos e relacionamentos com suas cardinalidades, referentes ao banco de dados do sistema.

A técnica de modelagem de dados mais difundida e utilizada é a abordagem entidade-relacionamento (ER). Nesta técnica, o modelo de dados é representado através de um modelo entidade-relacionamento (Modelo ER). Geralmente, um modelo ER é representado graficamente através de um diagrama entidade-relacionamento (DER). A abordagem ER foi criada em 1976 por Peter Chen, podendo ser considerada como um padrão de fato para a modelagem conceitual. (HEUSER, 2009)

A Figura 3 a seguir mostra a modelagem do banco de dados proposta para o desenvolvimento do sistema.

Figura 3– Diagrama Entidade-Relacionamento



Fonte: Elaborada pelos autores

7 Ferramentas e métodos de desenvolvimento

Os tópicos a seguir serão citados todas as ferramentas e métodos de desenvolvimento adotados no sistema.

7.1 Ferramentas

A utilização das ferramentas em uma aplicação é indispensável. Nos tópicos a seguir, iremos pormenorizar todas as ferramentas que utilizamos para realizar o desenvolvimento do sistema.

7.1.1 Sistema de gerenciamento de banco de dados: PostgreSQL

Banco de dados é uma coleção de informações persistentes referentes a um tema específico. Possui a obrigação de atender às necessidades de informação de seus usuários, sendo elas a de uma simples consulta ou a de geração de informações ou conhecimento.

Os dados possuem relevância crucial para o funcionamento não apenas do sistema, como também do próprio negócio. Por isso, além de zelar por sua proteção e bom uso, é necessário que, desde o início, sua modelagem tenha sido cuidadosa em refletir as características particulares das aplicações em que serão utilizados.

Os dados de uma empresa, se não forem o elemento mais precioso, estão entre eles. Uma informação armazenada incorretamente, ou de forma desordenada, pode custar todo o negócio. Sabendo disso, não tenha medo de desenhar esquemas, testar os esquemas das tabelas, trocar opiniões com outros desenvolvedores na hora de modelar um banco de dados (CARVALHO, 2017, p. 3).

Dentro de uma empresa, o sistema de banco de dados ou Sistema de Gerenciamento de Banco de Dados (SGBD) dá a possibilidade de realizar inclusão, alteração e até exclusão dos dados. Todavia, é recomendável que a exclusão seja substituída por uma marcação (*flag*) no registro, o que ainda permitirá consultas futuras a ele.

Para o presente projeto, foi escolhido o PostgreSQL como SGBD. A ferramenta foi criada em 1973 na Universidade de Berkeley na Califórnia, quando dois estudantes começaram a realizar um projeto com banco de dados relacional. A primeira versão disponível para usuários foi em 1989. Daí em diante a popularidade foi apenas aumentando conforme o tempo foi passando e novas funcionalidades foram sendo adicionadas (HESLEY, 2007).

O PostgreSQL suporta diversas funcionalidades, como:

- Comandos complexos
- Chaves estrangeiras

- Gatilhos
- Visões
- Integridade transaccional
- Controle de simultaneidade multiversão

E pode ser ampliado pelo usuário de muitas maneiras, como :

- Tipos de dado
- Funções
- Operadores
- Função de agregação
- Métodos de Índice
- Linguagem Procedurais

7.1.2 Visual Studio Code

A Microsoft lançou o editor de código em 2015, tendo em vista o suporte a várias linguagens e plataformas de codificação, estando disponível para os sistemas operacionais Windows, Linux e MacOS. Uma das principais características do editor de código é oferecer o suporte para mais de 30 linguagens de programação diferentes, como, por exemplo, JavaScript, Java, HTML, CSS, Java, entre outras. Além de ser gratuito, tem o código aberto (*open source*), com o código disponível no GitHub (STUDIO, 2021).

7.1.3 JavaScript

A linguagem de programação confere interatividade no *layout*. Com sua implementação, a página *web* faz muito mais do que apenas mostrar para os usuários as informações básicas, ganhando poder de processamento. JavaScript permite a um *site* ter mapas, barra de notificação, telas interativas etc.

Criada em 1995 por Brendan Eich, a linguagem originalmente iria rodar apenas no Netscape Navigator, e tinha como propósito oferecer processos de desenvolvimento mais dinâmicos para os desenvolvedores, tornando o uso mais rentável e agradável. A linguagem hoje em dia evoluiu muito e consegue atender a todo o ciclo de vida de uma aplicação *web* (*front-end* e *back-end*). Hoje em dia, é

utilizada também para aplicativos ricos para *desktop*, programas para celulares, entre outras aplicabilidades.

7.1.4 Node.js

O Node.js tem como característica oferecer um ambiente de execução da linguagem JavaScript sem necessidade de um *browser*. Com alto desempenho e baixo custo, o Node.js é uma ótima opção para ser utilizado entre os programadores *back-end* na criação de seus projetos. A criação de API é um ponto forte do Node.js

A ferramenta foi desenvolvida tendo como base o *engine* V8, utilizado pelo navegador Google Chrome para processar *scripts* JavaScript. Isso possibilitou que a linguagem pudesse ser manipulada também na programação de servidores, além de sua utilização no *front-end*.

7.1.5 Bootstrap

É um *framework front-end* de código-aberto, que consiste em um conjunto de sintaxes que definem comportamentos específicos. Confere aos desenvolvedores um meio de construir *sites* mais rapidamente, pois não precisam se preocupar com a complexidade do CSS puro (BOOTSTRAP, 2021)

O Bootstrap contém *templates* baseados em HTML e CSS. Por exemplo, possui a possibilidade de se colocar em uma página *web* as funções de grades, carrosséis, imagens com botões, entre outras opções para utilizar. Permite, ainda, que o usuário utilize o *site* com mais otimização para qualquer tamanho de tela, desde *smartphones* até telas maiores de *desktop*.

7.1.6 CANVA

É um editor de imagens *online*. A ferramenta tem disponibilidade em celulares Android ou IOS, porém na versão *web* tem recursos extras, mas as duas são úteis para situações diferentes. Mas é válido ressaltar que o Canva não é totalmente gratuito, existindo também a versão profissional que oferece mais fotos e elementos com personalização, podendo redimensionar o projeto e salvá-los em formato PNG (isto é, imagens com o fundo transparente) (CANVA, 2021).

7.1.7 GITHUB

A plataforma é uma rede social de desenvolvedores, onde usuários compartilham projetos que precisam ser públicos para mais pessoas. Sendo assim, cada pessoa que realizar alguma modificação no projeto, necessariamente precisa fazer *upload* do projeto no GitHub para realizar as alterações. Os outros usuários que estão utilizando o projeto devem fazer o *download* da nova versão e sempre atualizar o projeto, sendo assim todos estarão sempre com versões mais recentes.

Um dos pontos positivos é a equipe poder trabalhar, ao mesmo tempo, de qualquer lugar do mundo. A automatização dos fluxos de trabalho é fundamental e o GitHub tem como objetivo auxiliar no desenvolvimento dos projetos (GITHUB, 2021).

7.1.8 CHART.JS

A ferramenta é uma biblioteca na linguagem JavaScript de código aberto para visualização dos dados, auxiliando na criação de gráficos, com a utilização das linguagens HTML, CSS e JS para renderizar os gráficos. A ferramenta dispõe de vários formatos diferentes para a criação de gráficos, sendo alguns deles: dispersão, bolha, rodar, área, pizza, linha, barra, polar. (CHART.JS, 2021)

7.1.9 KNEX.JS

É um construtor de consultas a bancos de dados (*query builder*) que, usado em conjunto com o Node.js, facilita a integração da aplicação com alguns bancos de dados, como: Microsoft SQL Server, PostgreSQL e, Mysql. O Knex.js faz a transição dos dados que estão em diferentes bancos de dados, contando também com o sistema de *migration* (KNEX.JS, 2021).

7.2 Métodos ou Desenvolvimento

Nas imagens exibidas a seguir, apresenta-se os esquemas de banco de dados de cada tabela, seus campos e relacionamentos, de acordo com o Diagrama Entidade-Relacionamento, tendo como fonte de dados os requisitos do usuário.

A Figura 4 apresenta o esquema tabela “cliente”, na qual o usuário deverá inserir os dados referentes a cada campo, tendo como obrigatórios apenas os campos NomeCompleto, Tel1 e Endereco.

Figura 4 – Esquema de banco de dados da tabela cliente

```

1  exports.up = function(knex, Promise) {
2    return knex.schema.createTable('cliente', table => {
3      table.increments('CodCliente').primary()
4      table.string('NomeCompleto').notNull()
5      table.string('DataNasc')
6      table.string('Tel1', 1000).notNull()
7      table.string('Tel2', 1000)
8      table.string('CPF', 1000)
9      table.string('CNPJ', 1000)
10     table.string('Sexo', 1000)
11     table.string('Cidade', 1000)
12     table.string('UF', 1000)
13     table.string('Endereco', 2000).notNull()
14     table.string('Naturalidade', 1000)
15     table.string('EstadoCivil', 1000)
16   })
17 }
18 };
19
20 exports.down = function(knex, Promise) {
21   return knex.schema.dropTable('cliente')
22 };

```

Fonte: Elaborada pelos autores

A Figura 5 apresenta o esquema da tabela “venda”, em que o usuário deverá inserir os dados referentes a cada campo, sendo obrigatórios os campos NomeVendedor, data, Status e ValorTotal, além do clienteld, que deverá já estar previamente cadastrado.

Figura 5 - Esquema de banco de dados da tabela venda

```

1 exports.up = function(knex, Promise) {
2   return knex.schema.createTable('venda', table => {
3     table.increments('CodVenda').primary()
4     table.string('NomeVendedor').notNull()
5     table.string('Descricao', 1000)
6     table.string('data').notNull()
7     table.string('Status', 1000).notNull()
8     table.decimal('ValorTotal').notNull()
9     table.integer('clienteId').references('CodCliente').inTable('cliente').notNull()
10  })
11 };
12
13 exports.down = function(knex, Promise) {
14   return knex.schema.dropTable('venda')
15 };

```

Fonte: Elaborada pelos autores

A Figura 6 apresenta o esquema da tabela “produto”, na qual o usuário deverá inserir os dados referentes a cada campo, tendo como obrigatoriedade para o efetuar o cadastro apenas os campos NomeProduto, PrecoCusto e PrecoVenda.

Figura 6 - Esquema de banco de dados da tabela produto

```

1 exports.up = function(knex, Promise) {
2   return knex.schema.createTable('produto', table => {
3     table.increments('CodProduto').primary()
4     table.string('NomeProduto').notNull()
5     table.string('Descricao', 1000)
6     table.decimal('PrecoCusto').notNull()
7     table.decimal('PrecoVenda').notNull()
8     table.string('Tipo', 1000)
9     table.string('Data')
10  })
11 };
12
13 exports.down = function(knex, Promise) {
14   return knex.schema.dropTable('produto')
15 };

```

Fonte: Elaborada pelos autores

A Figura 7 apresenta o esquema da tabela “pedido”, em que o usuário deverá inserir os dados referentes a cada campo, tendo como obrigatórios os campos Quantidade, Data e Status, além de CodVenda e CodProduto que deverão já estar previamente cadastrados.

Figura 7 - Esquema de banco de dados da tabela pedido

```

1  exports.up = function(knex, Promise) {
2      return knex.schema.createTable('pedido', table => {
3          table.increments('CodPedido').primary()
4          table.integer('Quantidade').notNull()
5          table.string('Data').notNull()
6          table.string('Status').notNull()
7          table.integer('CodVenda').references('CodVenda').inTable('venda').notNull()
8          table.integer('CodProduto').references('CodProduto').inTable('produto').notNull()
9      })
10 };
11
12 exports.down = function(knex, Promise) {
13     return knex.schema.dropTable('pedido')
14 };

```

Fonte: Elaborada pelos autores

A Figura 8 apresenta o banco de dados da tabela “estoque”, em que o usuário deverá inserir os dados referentes a cada campo, tendo como obrigatoriedade para o efetuar o cadastro os campos NomeItem, precoCusto e PrecoVenda, Quantidade e DataAT, além do CodProduto que deverá já estar previamente cadastrado.

Figura 8 - Esquema de banco de dados da tabela estoque

```

1  exports.up = function(knex, Promise) {
2      return knex.schema.createTable('estoque', table => {
3          table.increments('CodEstoque').primary()
4          table.string('Obs', 1000)
5          table.integer('Quantidade').notNull()
6          table.string('DataAT').notNull()
7          table.integer('CodProduto').references('CodProduto').inTable('produto')
8      })
9  };
10
11 exports.down = function(knex, Promise) {
12     return knex.schema.dropTable('estoque')
13 };

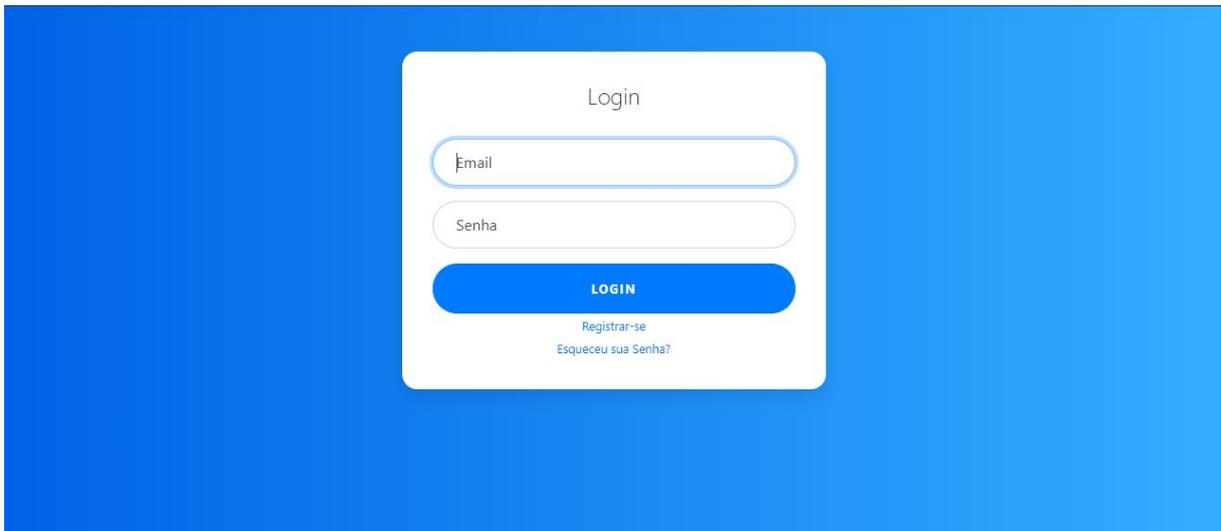
```

Fonte: Elaborada pelos autores

8 Resultados e Discussão

Neste tópico serão apresentados todos os passos do sistema, concentrando todas as informações relevantes que foram obtidas após a fundamentação do projeto.

A Figura 9 apresenta o layout da página de *login*, para conseguir acessar o sistema. É necessário inserir dados como *e-mail* e senha. Caso o usuário não possua cadastro para acessar, deverá entrar em contato com o administrador.

Figura 9 – Página de *login*

Login

Email

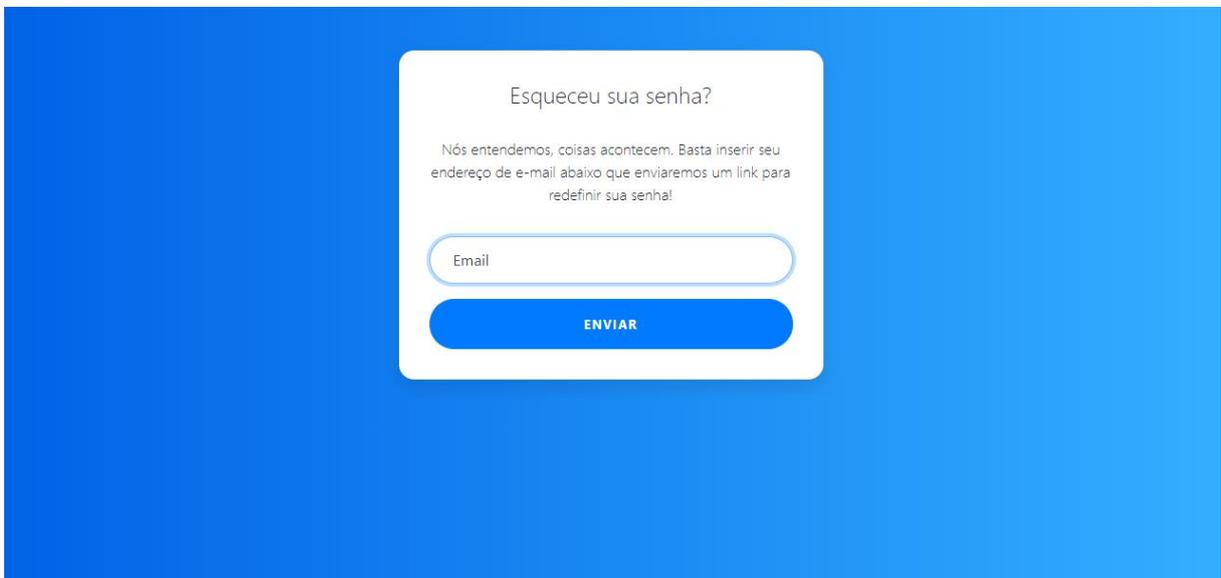
Senha

LOGIN

[Registrar-se](#)
[Esqueceu sua Senha?](#)

Fonte: Elaborado pelos autores

Caso o usuário possua acesso ao sistema, mas acabou esquecendo a senha, na Figura 10 vai solicitar o *e-mail* de cadastro e irá enviar uma mensagem de confirmação para ser alterada a senha.

Figura 10 – Recuperação de senha

Esqueceu sua senha?

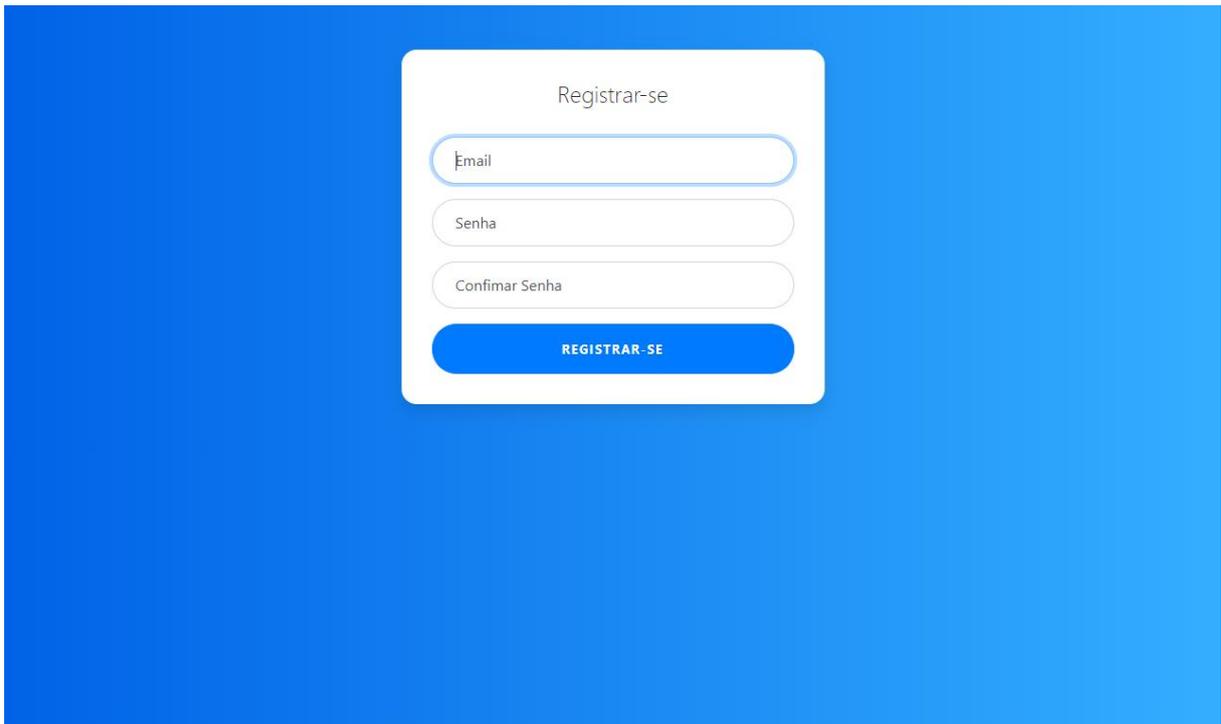
Nós entendemos, coisas acontecem. Basta inserir seu endereço de e-mail abaixo que enviaremos um link para redefinir sua senha!

Email

ENVIAR

Fonte: Elaborado pelos autores

Caso o usuário/a precise de acesso ao sistema deverá registrar-se, informando os dados na página representada na Figura 11 para criar uma nova conta de usuário e conceder-lhe acesso ao sistema.

Figura 11 - Criar conta

Registar-se

Email

Senha

Confirmar Senha

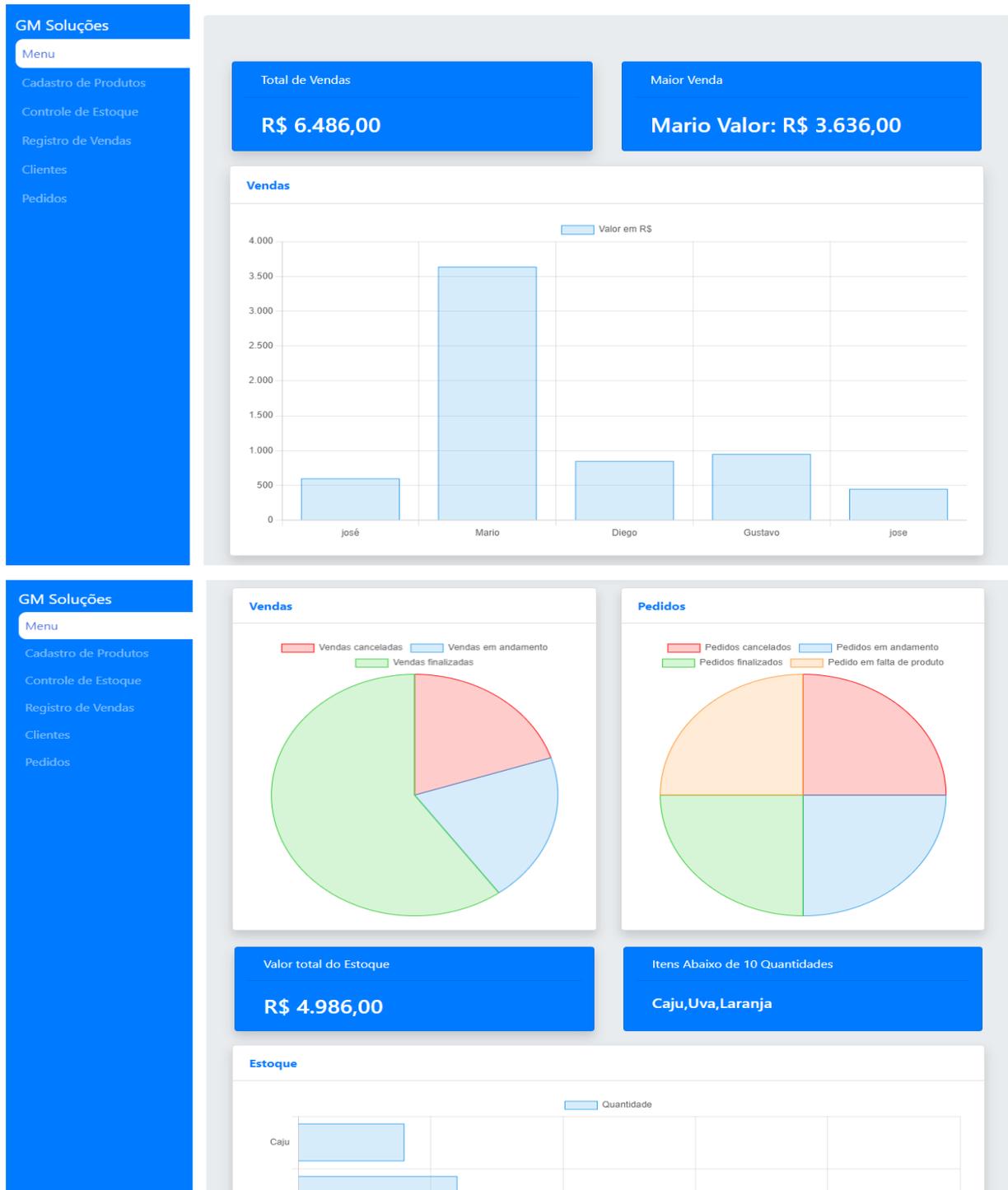
REGISTRAR-SE

O formulário de registro de usuário é exibido em um cartão branco centralizado sobre um fundo azul. O cartão contém o título "Registar-se" no topo, seguido por três campos de entrada de texto rotulados "Email", "Senha" e "Confirmar Senha". Abaixo dos campos, há um botão azul com o texto "REGISTRAR-SE" em letras maiúsculas brancas.

Fonte: Elaborado pelos autores

Após ter acesso ao sistema, o usuário verá o *dashboard* (Figura 12). A página mostra todos os dados gerais da empresa: onde circulam os lucros, as despesas, melhor e pior mês, melhor e pior ano. É útil para que o gestor possa verificar e fazer as devidas modificações onde julgar pertinente.

Figura 12 - Dashboard



Fonte: Elaborado pelos autores

Na página de cadastro do produto, o usuário deverá informar todos os dados solicitados (Figura 13), uma vez que o sistema não aceitará campos sem preenchimento.

Figura 13 - Cadastro de produto

Cadastro De Produto

Nome do Item: Código do Produto:

Descrição:

Preço de Custo: Preço de Venda:

Tipo: Data:

Cadastrar

Código do Produto	Nome do Item	Descrição	Preço de Custo	Preço de Venda	Tipo	Data de Cadastro	Editar	E
-------------------	--------------	-----------	----------------	----------------	------	------------------	--------	---

Fonte: Elaborado pelos autores

A página de controle de estoque (Figura 14) permite controlar o estoque da empresa. Para tanto, é obrigatório preencher todos os dados corretamente, com exceção do campo observação, que é opcional. As validações visam manter o estoque consistente.

Figura 14 - Controle de Estoque

Controle de Estoque

Nome do Item: Código do Produto:

Quantidade: Data de Atualização:

Observações:

Cadastrar

Código do Produto	Nome Item	Quantidade em Estoque	Preço de Custo	Preço de Venda	Valor total de ver
3	Caju	4	R\$ 25,00	R\$ 55,00	R\$ 220,00
1	Uva	6	R\$ 20,00	R\$ 50,00	R\$ 300,00
6	morango	15	R\$ 85,00	R\$ 100,00	R\$ 1.500,00
2	Banana	23	R\$ 66,00	R\$ 100,00	R\$ 2.300,00
7	Laranja	9	R\$ 45,00	R\$ 74,00	R\$ 666,00

Fonte: Elaborado pelos autores

Toda venda feita precisa ser registrada (Figura 15), fornecendo os valores, o código do cliente, a descrição caso necessário, a data da venda e o *status* da venda

Figura 15 - Registro de venda

The screenshot shows a web interface for 'Registro De Venda'. On the left is a blue sidebar with navigation options: 'GM Soluções', 'Menu', 'Cadastro de Produtos', 'Controle de Estoque', 'Registro de Vendas' (highlighted), 'Clientes', and 'Pedidos'. The main content area has a title 'Registro De Venda' and several input fields: 'Nome vendedor' (text), 'Descrição' (text), 'Valor total de Venda' (text) with a sub-field 'Preço de Venda R\$' (text), 'Data da venda' (calendar), 'Código do Cliente' (dropdown), and 'Status' (dropdown with 'Venda em Andamento' selected). A blue 'Cadastrar' button is below the form. At the bottom, a table header is visible with columns: 'Codigo da Venda', 'Descrição', 'Valor Total Venda', 'Status', 'Nome Vendedor', 'Código do Cliente', 'Data da Venda', and 'Edita'.

Fonte: Elaborado pelos autores

Todo e qualquer cliente que compre algo, precisa ser cadastrado no sistema (Figura 16), pois a carteira de clientes que já adquiriram produtos da empresa pode auxiliar em buscas de novas promoções, de faixa etária etc.

Figura 16 – Cadastro de cliente

The screenshot shows a web interface for 'Cadastro De Clientes'. On the left is a blue sidebar with navigation options: 'GM Soluções', 'Menu', 'Cadastro de Produtos', 'Controle de Estoque', 'Registro de Vendas', 'Clientes' (highlighted), and 'Pedidos'. The main content area has a title 'Cadastro De Clientes' and several input fields: 'Nome Completo' (text), 'Data da nascimento' (calendar), 'Telefone' (text), 'Telefone 2' (text), 'Naturalidade' (text), 'Estado civil' (dropdown), 'Sexo' (dropdown), 'CPF' (text), 'CNPJ' (text), 'UF' (dropdown), 'Endereço' (text), and 'Cidade' (text). A blue 'Cadastrar' button is below the form. At the bottom, a table header is visible with columns: 'Código Cliente', 'Nome Complete', 'Telefone', 'Telefone 2', 'Sexo', 'Estado civil', 'Data Nascimento', 'CPF', 'CNPJ', and 'Endereço'.

Fonte: Elaborado pelos autores

A página de pedidos (Figura 17) permite exibir os pedidos já existentes, incluindo os que foram porventura cancelados.

Figura 17 – Pedidos

Fonte: Elaborado pelos autores

Considerações finais

No início do projeto, pensamos em que público iríamos atingir realizando o sistema, e como poderíamos desenvolver, pois a forma e a plataforma em que seria realizado comprometem muito para a conclusão efetiva. O banco de dados é hospedado no servidor, com possibilidade de outras máquinas na mesma rede terem acesso aos dados. O *back-end* foi idealizado para ser independente; portanto, caso mude a tecnologia, não será necessário realizar muitas alterações. No *front-end*, serão realizadas mudanças caso o cliente necessite de novas funcionalidades de acordo com o uso diário do usuário.

No entanto, mesmo tendo como objetivo pequenas empresas, o foco do projeto sempre foi o público geral. Portanto, caso alguém requisite novas funcionalidades, poderíamos realizar algumas modificações para atender ao usuário.

O objetivo do desenvolvimento do ERP foi simplificar o uso e dar acesso a empresas que estão no início do empreendimento, oferecendo um sistema que esteja ao alcance do pequeno empresário.

Durante a criação do sistema, tivemos algumas dificuldades que seria a inclusão dos dados no front-end. Outra dificuldade que tivemos foi o desenvolvimento

da documentação, ambos não tinham familiaridade em como realizar os processos. Porém, com ajuda de amigos, familiares, e professores isso tudo foi possível para podermos finalizar a documentação.

Temos em mente prosseguir com o projeto, pois pessoas as quais pesquisamos para a realização do sistema precisam de auxílio sistêmico para melhorar suas empresas. Sendo assim, pensamos em continuar e poder ter esses clientes conosco usando o sistema.

Referências

BIZAGI. bizagi.com. **Bizagi**, 2021. Disponível em: <<https://www.bizagi.com/pt>>. Acesso em: 03 maio 2021.

BOOTSTRAP. getbootstrap.com/. **Bootstrap**, 2021. Disponível em: <<https://getbootstrap.com/>>. Acesso em: 03 maio 2021.

CANVA. canva.com. **Canva**, 2021. Disponível em: <<https://www.canva.com/>>. Acesso em: 03 maio 2021.

CHART.JS. www.chartjs.org. **Chart.js**, 2021. Disponível em: <<https://www.chartjs.org/>>. Acesso em: 03 maio 2021.

CODIFICAR. codificar.com.br. **Codificar**, 2021. Disponível em: <<https://codificar.com.br/requisitos-funcionais-nao-funcionais>>. Acesso em: 03 Maio 2021.

GITHUB. GitHub: Where the world builds software. **GITHUB**, 2021. Disponível em: <<https://github.com>>. Acesso em: 03 maio 2021.

HESLEY. Introdução ao PostgreSQL. **devmedia**, 2007. Disponível em: <<https://www.devmedia.com.br/introducao-ao-postgresql/6390>>.

HEUSER, C. A. **Projeto de banco de dados**. Porto Alegre: Bookeman, 2009.

KNEX.JS. <http://knexjs.org/>. **knex.js**, 2021. Disponível em: <<http://knexjs.org/>>.

NODE.JS. nodejs.org. **Node js**, 2021. Disponível em: <<https://nodejs.org/en/>>. Acesso em: 03 maio 2021.

PGADMIN. www.pgadmin.org. **PgAdmin**, 2021. Disponível em: <<https://www.pgadmin.org/>>. Acesso em: 03 maio 2021.

SOMMERVILLE. **Engenharia de Software**. São Paulo: Pearson Education do Brasil, 2011.

STUDIO, V. code.visualstudio.com. **Visual Studio**, 2021. Disponível em: <<https://code.visualstudio.com/>>. Acesso em: 03 maio 2021.