

**INSTITUTO FEDERAL DO PARANÁ**

**CRISTIANE MOTIN  
ÉDER DIEGO DE SOUSA**

**SIGIC: SISTEMA DE GESTÃO INTEGRADO COMERCIAL**

**COLOMBO**

**2021**

**INSTITUTO FEDERAL DO PARANÁ**

**CRISTIANE MOTIN  
ÉDER DIEGO DE SOUSA**

**SIGIC: SISTEMA DE GESTÃO INTEGRADO COMERCIAL**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal do Paraná, Campus Colombo, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Ademir Luiz do Prado

**COLOMBO**

**2021**

## TERMO DE APROVAÇÃO

**CRISTIANE MOTIN  
ÉDER DIEGO DE SOUSA**

### **SIGIC: SISTEMA DE GESTÃO INTEGRADO COMERCIAL**

Trabalho de Conclusão de Curso aprovado como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pelo Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, do Instituto Federal do Paraná, Campus Colombo, pela seguinte banca examinadora:



Prof. Me. Ademir Luiz do Prado  
Orientador



Prof. Me. Gustavo Bigetti Guergoletto



Prof. Me. Marcos Dinís Lavarda

Colombo, 15 de outubro de 2021.

## **AGRADECIMENTOS**

Primeiramente agradecemos a Deus por permitir chegar até esse momento. Agradecemos ao professor orientador e coordenador do curso Ademir que com seu conhecimento e experiência, contribuiu imensamente na execução deste projeto. Agradecimento aos nossos colegas pela amizade, pelas trocas de experiências e conhecimentos que contribuíram com nossas trajetórias durante o curso. Por fim, agradecemos aos demais professores e funcionários do IFPR por proporcionarem as condições necessárias para o que ensino e o aprendizado sejam de qualidade nesta instituição

*“A verdadeira motivação vem de  
realização, desenvolvimento pessoal,  
satisfação no trabalho e reconhecimento.”*

Frederick Herzberg

## RESUMO

No presente Trabalho de Conclusão de Curso buscou-se desenvolver um software para controle de pedidos, entradas e saídas de insumos, estoque dos produtos, as entradas e saídas do caixa, vendas, entre outras atividades para médios e pequenos comércios. De início será implantado em uma empresa familiar, com o objetivo de melhorar a gestão do negócio. Para redução de erros manuais, buscou-se a partir do presente estudo, soluções para otimizar o processo. O sistema foi desenvolvido utilizando a plataforma Java, usando framework Spring MVC com *Thymeleaf*. E o banco de dados H2 é o banco em memória e o PostgreSQL o banco em produção. Após a implementação da aplicação obteve-se um resultado positivo, pois com a aplicação a empresa poderá ter controle mais efetivo de suas compras e vendas, controle dos clientes e de suas compras a pagar e a receber.

Palavras-chave: ERP. Java. Spring. Responsivo.

## **ABSTRACT**

In this Course Conclusion Work, we sought to develop a software to control orders, inputs and outputs of inputs, stock of products, cash inputs and outputs, sales, among other activities for medium and small businesses. Initially, it will be implemented in a family business, with the aim of improving business management. To reduce manual errors, solutions to optimize the process were sought from this study. The system was developed using the Java platform, using Spring MVC framework with Thymeleaf. And the H2 database is the in-memory database and PostgreSQL the production database. After the implementation of the application, a positive result was obtained, as with the application the company will be able to have more effective control of its purchases and sales, control of customers and their purchases to be paid and receivable.

Keywords: ERP. Java. Spring. Responsive.

## LISTA DE FIGURAS

FIGURA 1 - IDE INTELLIJ COMMUNITY.....	17
FIGURA 2 - PAINEL DE ACESSO AO BANCO H2 EM NAVEGADOR WEB COM SPRING	20
FIGURA 3 - FLUXOGRAMA PARA FAZER PEDIDO.....	30
FIGURA 4 - CASO DE USO .....	34
FIGURA 5 - DIAGRAMA DE CLASSES .....	45
FIGURA 6 - DIAGRAMA DE ENTIDADE E RELACIONAMENTO .....	47
FIGURA 7 - TELA LOGIN .....	48
FIGURA 8 - TELA DASHBOARD.....	48
FIGURA 9 - TELA CADASTRO .....	49
FIGURA 10 - TELA CADASTRO FORNECEDOR .....	49
FIGURA 11 - TELA CADASTRO CLIENTES .....	50
FIGURA 12 - TELA CADASTRO PRODUTOS.....	50
FIGURA 13 - TELA AJUSTAR ESTOQUE.....	51
FIGURA 14 - TELA EFETUAR VENDAS .....	51
FIGURA 15 - TELA LISTA DE VENDAS.....	52

## LISTA DE QUADROS

QUADRO 1 – DETALHAMENTO DO CASO DE USO FAZER LOGIN.....	35
QUADRO 2 – DETALHAMENTO DO CASO DE USO PEDIDO DE VENDA DO PRODUTO.....	36
QUADRO 3 – DETALHAMENTO DO CASO DE USO PEDIDO DE COMPRA .....	37
QUADRO 4 – DETALHAMENTO DO CASO DE USO HISTÓRICO DE VENDAS .....	37
QUADRO 5 – DETALHAMENTO DO CASO DE USO CADASTRO DE FUNCIONÁRIO .....	38
QUADRO 6 – DETALHAMENTO DO CASO DE USO CADASTRO DE FORNECEDOR.....	39
QUADRO 7 – DETALHAMENTO DO CASO DE USO CONSULTA DO FORNECEDOR.....	40
QUADRO 8 – DETALHAMENTO DO CASO DE USO CONTROLE DE ESTOQUE .....	41
QUADRO 9 – DETALHAMENTO DO CASO DE USO CONSULTA DO CLIENTE.....	42
QUADRO 10 – DETALHAMENTO DO CASO DE USO CADASTRO DO CLIENTE.....	42
QUADRO 11 – DETALHAMENTO DO CASO DE USO AUTORIZA NOVOS USUÁRIOS ...	43

## LISTA DE SIGLAS E ABREVIATURAS

- API - *Application Programming Interfaces*
- CSS - *Cascating Style Sheets*
- DAO - *Data Access Object*
- ERP - *Enterprise Resource Planning*
- HTML - *HyperText Markup Language*
- JEE - *Java Enterprise Edition*
- JSE - *Java Standard Edition*
- UML - *Unified Markup Language*

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	PROBLEMA	13
1.2	HIPÓTESE	14
<b>2</b>	<b>OBJETIVOS</b>	<b>15</b>
2.1	OBJETIVO GERAL	15
2.2	OBJETIVOS ESPECÍFICOS	15
<b>3</b>	<b>REVISÃO DA LITERATURA</b>	<b>16</b>
3.1	TECNOLOGIAS UTILIZADAS	16
3.1.1	Ambiente de Desenvolvimento	16
3.1.2	IDE	16
3.1.3	ERP	17
3.1.4	Java	18
3.1.5	PostgreSQL	19
3.1.6	H2	19
3.1.7	Thymeleaf	20
3.1.7.1	Características do Thymeleaf	20
3.1.7.2	Como o Thymeleaf funciona?	21
3.1.7.3	Quem utiliza o Thymeleaf?	22
3.1.7.4	Conclusão	22
3.1.8	HTML	22
3.1.9	CSS	23
3.1.10	JavaScript	23
3.1.11	Frameworks	24
3.1.12	Bootstrap	24
3.1.13	Spring	25
3.1.13.1	Spring MVC	25
3.1.13.2	Spring Data	25
3.1.13.3	Spring Boot	25
3.1.13.4	Spring Security	26
3.1.14	Heroku	26
3.1.15	Git	26
3.1.16	GitHub	27
3.1.17	Astah Community	27

<b>4</b>	<b>DESENVOLVIMENTO .....</b>	<b>28</b>
4.1	METODOLOGIA .....	28
4.1.1	Estudo de Caso.....	29
4.1.1.1	História da Empresa.....	29
4.1.1.2	Gestão do Negócio .....	29
4.1.1.3	Funcionamento da Empresa .....	30
4.1.1.4	Problemas existentes na empresa .....	30
4.2	LEVANTAMENTO DE REQUISITOS.....	31
4.2.1	Requisitos Funcionais .....	31
4.2.2	Requisitos Não Funcionais.....	32
4.3	DIAGRAMAS DA UML .....	33
4.3.1	Diagrama de Casos de Uso .....	33
4.3.1.1	Caso de Uso Fazer Login.....	35
4.3.1.2	Caso de Uso Pedido de Venda .....	35
4.3.1.3	Caso de Uso Pedido de Compra.....	36
4.3.1.4	Caso de Uso Histórico de Vendas .....	37
4.3.1.5	Caso de Uso Cadastrar Funcionário .....	38
4.3.1.6	Caso de Uso Cadastrar Fornecedor .....	38
4.3.1.7	Caso de Uso Consulta Fornecedor .....	39
4.3.1.8	Caso de Uso Controle de Estoque.....	40
4.3.1.9	Caso de Uso Consulta Cliente .....	41
4.3.1.10	Caso de Uso Cadastrar Cliente .....	42
4.3.1.11	Caso de Uso Autoriza ou Revoga Novos Usuários .....	43
4.3.2	Diagrama de Classes .....	43
4.4	MODELAGEM DO SISTEMA.....	46
4.4.1	Diagrama Entidade Relacionamento .....	46
<b>5</b>	<b>RESULTADOS .....</b>	<b>48</b>
<b>6</b>	<b>CONCLUSÃO .....</b>	<b>53</b>
	<b>REFERÊNCIAS.....</b>	<b>54</b>
	<b>APÊNDICES .....</b>	<b>57</b>

## 1 INTRODUÇÃO

O presente trabalho abordou o desenvolvimento de um sistema de pedidos com o conceito de planejamento de recurso corporativo ou *Enterprise Resource Planning* (ERP). Proporcionando com isso que a empresa tivesse um crescimento tecnológico e um melhor controle dos pedidos realizados. A empresa estava passando por problemas na gestão do negócio, já que todos os cálculos de venda e compra de produtos eram feitos manualmente. Por esse motivo a gestão era dificultada e propensa a erros. Então com isso pensou-se: como aperfeiçoar a gestão da empresa, agilizando os processos existentes e reduzindo erros da gestão manual. Com o desenvolvimento do sistema para a gestão, esperou-se minimizar os problemas que a gestão manual apresentava, automatizando o processo de fazer pedidos e transformando os cálculos que até então eram manuais para que a aplicação executasse automaticamente.

A empresa não possuía qualquer tipo de sistema computacional para gestão de seus negócios. Com a falta de um sistema para controlar os pedidos e a empresa, os clientes faziam seus pedidos e às vezes acaba que não tem o produto solicitado devido à falta de um controle de estoque. Também convém citar o fato de não possuir sistema acarretava muitos documentos escritos ou impressos para o controle da empresa. Segundo Geremias e Papior (2006), o uso de um sistema ERP para microempresas é recomendado, pois apoia a gestão de todas as atividades proporcionando uma visão mais detalhada do negócio. A necessidade de implantação desse tipo de sistema foi identificada quando os antigos procedimentos já não estavam mais sendo tão eficientes. Com essa afirmação, de Geremias e de Papior (2006), não apenas uma empresa necessita de possuir um sistema ERP, mas como é recomendado para todas as microempresas possuir um sistema como esse, para controlar os seus negócios. Com o uso do sistema ERP que foi desenvolvido, a empresa pode ter seus processos melhorados e isso pode ajudar no seu crescimento.

### 1.1 PROBLEMA

A empresa não possuía qualquer tipo de sistema computacional para gestão de seus negócios. Com isso existia a falta do controle de seu estoque, falta de informações para seguir com o processo.

## 1.2 HIPÓTESE

Com o desenvolvimento do sistema para a gestão, espera-se minimizar os problemas que a gestão manual apresenta, automatizando o processo de fazer pedidos e transformando os cálculos que são manuais para que o sistema otimize o tempo.

## 2 OBJETIVOS

A seguir serão apresentados os objetivos gerais e específicos do trabalho proposto

### 2.1 OBJETIVO GERAL

Desenvolver uma aplicação, usando o conceito de ERP para gestão de negócio da empresa Cantinho da Costela, para um melhor controle dos pedidos feitos na empresa

### 2.2 OBJETIVOS ESPECÍFICOS

- Estudar o funcionamento da empresa;
- Modelar o software
- Desenvolver uma aplicação para realizar a gestão dos processos.

### 3 REVISÃO DA LITERATURA

Apresentaremos neste capítulo a fundamentação teórica sobre as tecnologias utilizadas para o desenvolvimento do sistema.

#### 3.1 TECNOLOGIAS UTILIZADAS

Neste capítulo são apresentados os conceitos e tecnologias que foram utilizadas, linguagem de programação, ambiente de desenvolvimento integrado e framework utilizados no desenvolvimento do projeto, baseando-se em pesquisas bibliográficas.

##### 3.1.1 Ambiente de Desenvolvimento

Ambiente de Desenvolvimento Integrado vem do inglês *Integrated Development Environment* (IDE) e são um conjunto de softwares que auxiliam no desenvolvimento de aplicações, sendo muito utilizado por desenvolvedores, com o objetivo de facilitar diversos processos (ligados ao desenvolvimento), que combinam ferramentas comuns em uma única Interface Gráfica do Usuário. Para o desenvolvedor, é uma forma de criar aplicações de maneira mais rápida, uma vez que estas IDEs auxiliam em todo o processo de desenvolvimento de uma aplicação, provendo diversos benefícios, como a análise de todo o código a ser escrito para identificar bugs causados por um erro de digitação, autocompletam trechos de códigos entre outros recursos.

##### 3.1.2 IDE

Para o desenvolvimento do sistema de pedidos, utilizou-se a *IDE IntelliJ Community Edition*, figura mostrada na (Figura 1). Existem diversas IDEs para se desenvolver aplicações com JavaEE e também para o desenvolvimento front-end, porém nem todas possuem uma integração tão grande com outros softwares e em conjunto serem fáceis de se utilizar, consumindo poucos recursos do computador que estiver executando.

Com isso o *IntelliJ IDEA Ultimate da JetBrains* vem atendendo aos requisitos, pois possui além da funcionalidade de IDE para desenvolvimento em Java EE também possui outras IDEs da empresa integradas, como *DataGrid* que é uma IDE para banco de dados, servindo para a geração de diagramas de diversos SGBD, *Webstorm* como IDE de front-end (JETBRAINS, 2018).

Além de todas estas funcionalidades citadas, conta também com ferramentas para auxiliar na integração com o Git e a geração automática de diagramas de classe, pacote, modelagem das tabelas de banco de dados; necessárias para a construção do código.

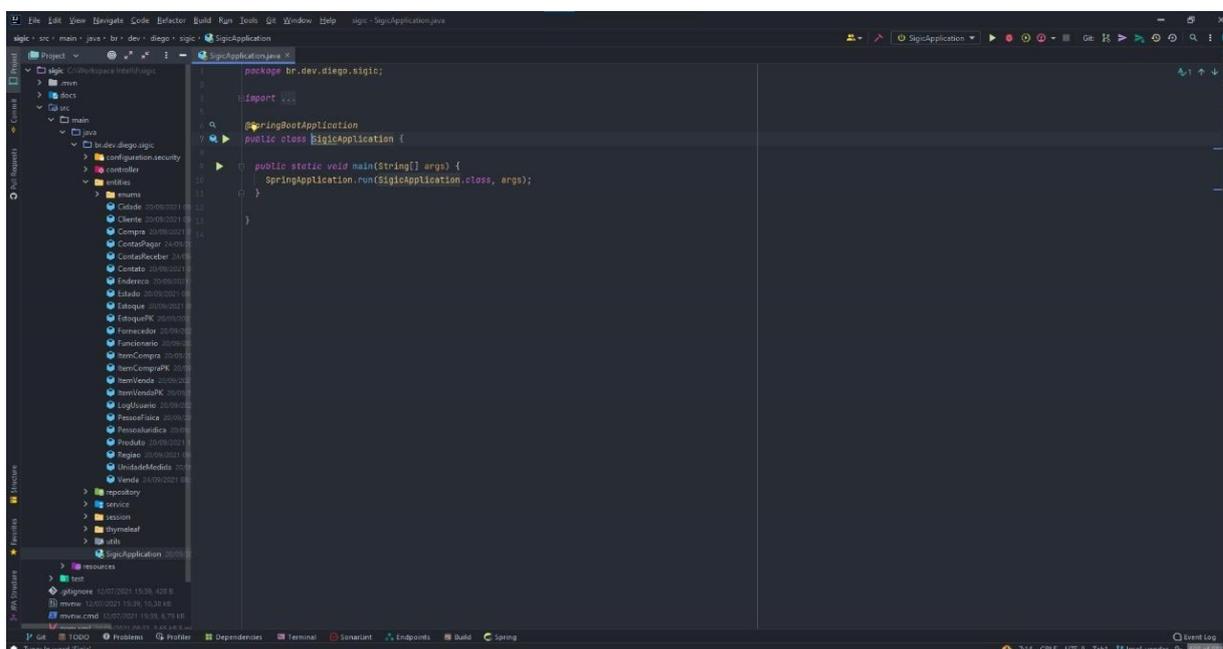


FIGURA 1 - IDE INTELLIJ COMMUNITY  
FONTE: INTELLIJ (2021)

### 3.1.3 ERP

O ERP surgiu da evolução de outras duas siglas que também representavam o programa de produção: *Material Requirement Planning* (MRP) e *Material Requirement Planning* (MRP II).

Em 1990 o MRP II incorporou aspectos de recursos humanos e financeiro. Com a grande mudança começou a ser chamado de ERP (TOSTES, 2009). Segundo Colangelo (Campos, 2006), MRP surgiu com o objetivo de dar apoio às funções de planejamento de produção e compras.

Seus procedimentos eram baseados em listas de matérias e um plano de produção. Após algum tempo o MRP englobou aspectos financeiros, orçamentos e custeio de produção, mas ainda não englobava todos os aspectos do sistema da empresa e o suporte completo e integrado da empresa era difícil. Em 1990 sofreu mais modificações e passou a ser chamado de ERP. O ERP trabalha em um ambiente cliente-servidor, com base de dados única e integrada, que possibilita à empresa ter uma visão mais ampla da utilização de recursos em diversas áreas.

Ainda segundo Colangelo (CAMPOS, 2006), à medida que o ERP ampliava suas áreas de atuação dentro das organizações acabou por determinar a classificação para uma determinada categoria de software, que é possível ver “ERP para comércio” ou “ERP para serviços”.

#### 3.1.4 Java

O Java é uma linguagem de programação usada em diversos projetos, sendo web ou desktop, foi lançado em 1995 pela empresa *Sun Microsystems*. Essa nova linguagem cresceu rapidamente entre os desenvolvedores, pois poderia ser usada em diversos sistemas operacionais. O Java é encontrado atualmente em inúmeros servidores web, bancos de dados relacionais, computadores de grande porte, telefones móveis, sistemas de cartão de crédito entre outros (GONÇALVES, 2007). Java é uma tecnologia usada para desenvolver aplicações que tornam a Web um lugar mais divertido e útil. Java não é igual a JavaScript, que só é executado em seu browser. Java permite executar jogos, fazer uploads de fotos e fazer bate papos online entre outras funções. Se você não possui o Java em seu computador muitas aplicações web simplesmente não iriam funcionar (JAVA, 2017). Programas Java consistem em partes conhecidas como classes. As classes possuem métodos que realizam tarefas e retornam informações quando são concluídas. A linguagem Java já possui várias bibliotecas para usar em seus programas, que também são conhecidas como *Java Application Programming Interfaces (APIs)*. Em geral, programas Java passam por 5 fases: edição, compilação, carregamento, verificação e execução (DEITEL, DEITEL, 2010). Segundo Deitel e Deitel (2010, p.11), “Java é uma poderosa linguagem de programação”. Por esse motivo que foi utilizado o Java na aplicação e por ela sempre ter crescido junto com os desenvolvedores existem muitos desenvolvedores Java e muita documentação para o desenvolvimento.

### 3.1.5 PostgreSQL

O PostgreSQL começou a sua construção em meados de 1986 na Universidade Norte-Americana da Califórnia, a Universidade de Berkeley. Em 1995 obteve um interpretador SQL que recebeu o nome de Postgres 95, onde teve um aumento de 25% em seu código fonte. Então em 1996 teve uma abertura completa de seu código fonte. Nessa época já era conhecido como PostgreSQL (Pereira Neto, 2007).

Com a abertura do sistema, vários usuários começaram a colaborar com a melhoria do software. O PostgreSQL tem suporte para Windows, Linux e Solaris e MAC. Possui a maioria dos tipos de dados SQL como: INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL e TIMESTAMP. Também suporta armazenamento de imagens, sons ou vídeo. Possui interfaces de programação nativas para C / C ++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre outros (POSTGRESQL, 2017). Toda a aplicação precisa ter um banco de dados para a persistência dos dados. O PostgreSQL é suportado em diversos sistemas operacionais e como possui licença aberta foi a escolha cabível no desenvolvimento do projeto, para guardar os dados que serão gerados pela aplicação.

### 3.1.6 H2

Um banco de dados puramente implementado em Java. Em um único jar com pouco mais de 1Mb ele traz: um poderoso SGBD que pode rodar tanto em modo *embedded* quanto *Client-Server*, um driver JDBC tipo 4 e várias ferramentas como uma interface de administração web com autocomplete para SQL. É possível fazer o download do instalador para Windows, do zip multiplataforma, ou ainda apenas do jar do SGBD. Com o jar ou a versão zipada nenhuma instalação é necessária. Basta clicar duas vezes em h2-(versão) jar que o console do banco aparecerá no seu navegador padrão. Se o seu sistema suportar o conceito de “Área de notificação” (classe System Tray do Java) um ícone amarelo também deve aparecer; a partir deste você pode, a qualquer momento, acessar o console, conforme a Figura 2. (ACCIOLY, 2011).



Login

Configuração ativa: Generic H2 (Embedded) ▼

Nome da configuração: Generic H2 (Embedded) Gravar Remover

Classe com o driver: org.h2.Driver

JDBC URL: jdbc:h2:mem:testdb

Usuário: sa

Senha:

Conectar Testar conexão

FIGURA 2 - PAINEL DE ACESSO AO BANCO H2 EM NAVEGADOR WEB COM SPRING

FONTE: ACCIOLY (2011)

### 3.1.7 Thymeleaf

O Thymeleaf é uma *template engine* para projetos Java que facilita a criação de páginas HTML. Sendo assim, ele serve para gerar páginas HTML no lado servidor de forma dinâmica, permitindo a troca de informações entre o código Java e as páginas HTML, de tal maneira ele garante que o desenvolvedor consiga criar *templates* de forma mais fácil para suas aplicações.

#### 3.1.7.1 Características do Thymeleaf

Antes de mais nada a principal funcionalidade de um *template engine* é permitir que linguagens de programação possam ser incorporadas em páginas HTML. de tal forma, uma *template engine* permite que os programadores possam utilizar estruturas de condição, estruturas de repetição, herança e diversos outros recursos presentes apenas nas linguagens de programação em páginas HTML.

Analogamente o Thymeleaf não é diferente, ele permite que desenvolvedores incorporem código Java em páginas HTML e utilizem as principais características da linguagem em seus *templates*.

Dentre diversas características, podemos citar as principais:

- Permite o uso de estruturas de condição e repetição em páginas HTML;

- Com o Thymeleaf é possível utilizar herança de layouts, garantindo uma estrutura com um maior reaproveitamento de código;
- Permite exibir o conteúdo de variáveis Java em páginas HTML;
- Sistema de fragmentos de *templates*, dentre outros.

### 3.1.7.2 Como o Thymeleaf funciona?

Basicamente, quando criamos um *template* com Thymeleaf e incorporamos código Java nas páginas HTML, a própria ferramenta traduz o código Java e incorpora à página HTML, já que o Browser não consegue exibir código diferente do HTML.

Abaixo temos um exemplo de código escrito com o Thymeleaf:

```
<ul>
  <li th:each="user : ${users}" >
    <a
      th:href="/user/{username} (username=${user.username})"
      th:text="${user.firstname} + ' ' + ${user.lastname}"
    ></a>
  </li>
</ul>
```

Após o processamento do *template* realizado pelo Thymeleaf será gerado um código HTML como o do exemplo abaixo:

```
<ul>
  <li>
    <a href="/user/joao_silva">João da Silva</a>
  </li>
  <li>
    <a href="/user/maria_luiza">Maria Luiza</a>
  </li>
  <li>
    <a href="/user/jose_ramos">José Ramos</a>
  </li>
  <li>
    <a href="/user/neuza_cristina">Neuza Cristina</a>
  </li>
  <li>
    <a href="/user/gerald_santos">Geraldo Santos</a>
  </li>
</ul>
```

Certamente podemos notar a quão poderosa é essa ferramenta. Com ela podemos utilizar os principais recursos da linguagem de programação para a criação de páginas HTML.

#### 3.1.7.3 Quem utiliza o Thymeleaf?

O Thymeleaf é muito utilizado em todo ecossistema Java, sendo principalmente utilizado em aplicações Spring Web MVC, um dos frameworks mais famosos do ecossistema Java para desenvolvimento de aplicações Web, além disso o Thymeleaf possui um starter oficial para o Spring Boot.

#### 3.1.7.4 Conclusão

Por fim, como vimos neste artigo, o Thymeleaf é uma ótima *template engine* que permite a criação de páginas HTML utilizando os recursos do Java.

### 3.1.8 HTML

Segundo Silva (2007), é uma linguagem utilizada para produzir páginas web. Essas páginas são reproduzidas por navegadores instalados nos dispositivos de seus visitantes. Estes navegadores têm a capacidade de ler a linguagem HTML, e a partir dela renderizar na tela o conteúdo das páginas, como textos, tabelas, formulários de entrada de dados, entre outros. O HTML atualmente se encontra na sua 5ª versão (conhecida como HTML5), que é a evolução da versão 4. O principal objetivo desta versão é deixar o HTML mais semântico, ou seja, dar mais significado e objetividade na leitura do código (MAZZA, 2013). O HTML5 também oferece maiores facilidades para que as folhas de estilo em cascata (CSS) e o Javascript (que serão apresentados a seguir) possam realizar seu trabalho com maior eficiência (FERREIRA, 2011). Atualmente grandes empresas utilizam HTML5, pode-se citar entre elas a Adobe 2, Ebay3, Paypal4, Facebook5, Intel 6 e a Microsoft 7 (W3C, 2014d). Como vantagem do uso do HTML5, pode-se citar que por ele possuir uma maior semântica em relação às versões anteriores, possibilita aos buscadores mais precisão nas buscas, visto que o código é mais interpretável, então a busca de dados acaba tendo um maior sucesso (MAZZA, 2013).

### 3.1.9 CSS

É uma linguagem de estilização que complementa o HTML. Ela serve para definir cores, alinhamentos, backgrounds etc. Sua maior finalidade, é separar o estilo da página, de seu conteúdo, deixando assim, o código mais limpo e facilitando a sua manutenção e escrita (MURPHY et al., 2012). Silva (2011) cita que, na versão 3 (conhecida como CSS3), o CSS traz diversas novas funcionalidades de cor, alinhamento e espaçamento de elementos, estilização de imagens, entre outros. Porém, sua melhor conquista foi o design responsivo. Segundo Mazza (2013) design responsivo, é o poder que dá a uma aplicação, ser renderizada e visualizada amigavelmente em dispositivos de diferentes tipos e tamanhos. Como exemplo, pode-se citar uma página web que é criada sem que o desenvolvedor se preocupe com a sua correta visualização em diferentes dispositivos. Talvez esta página, dependendo dos componentes utilizados e da forma que eles são manipulados, possa ser visualizada corretamente em um monitor ou dispositivo com a mesma resolução da tela que foi utilizada pelo desenvolvedor enquanto criava a página. Já a mesma página pode ser visualizada por meio de um celular, e neste dispositivo fica totalmente desconfigurada. Isto se dá, pela não-validação dos estilos empregados na página. Quando se trabalha com responsividade, este problema não existe, já que a aplicação deve ser bem visualizada em qualquer tipo de dispositivo e em todas as suas variações de tamanho.

### 3.1.10 JavaScript

É uma linguagem que oferece interação em tempo real com usuários de uma página web. Ele continua crescendo cada vez mais em seu número de adeptos e é uma das principais linguagens da web de hoje em dia. O JavaScript permite que o usuário possa interagir com a página. É uma linguagem que trabalha com eventos, podendo ser considerado um evento, entre outros, clicks do mouse, movimentos do mouse, uma tecla ser pressionada etc. Também é orientada a objeto, o que entre outras coisas, traz mais organização, produtividade e desempenho para o programador (POWERS, 2008). Segundo Morrison (2008), o Javascript permite manipular tudo que é renderizado por um navegador.

Ele é capaz de capturar praticamente todas as ações do usuário em uma página web. Por ser considerado um “meio de comunicação” entre o usuário e a página, o JavaScript é lido, renderizado e executado dentro do navegador. Ou seja, ele trabalha do lado do cliente na maioria de suas utilizações. Isso pode ser considerada uma vantagem, do ponto de vista de que a comunicação se torna mais rápida por estar do lado do cliente, e uma desvantagem, se pensarmos que o programador não consegue garantir segurança, já que a aplicação roda em um ambiente que o desenvolvedor não manipula (MORRISON, 2008).

### 3.1.11 Frameworks

Para que fosse possível implementar o sistema foi necessário utilizar-se de alguns frameworks. Para maior agilidade no desenvolvimento da aplicação e para implementação de um design responsivo.

Várias definições sobre framework são descritas na literatura, mas segundo Gamma (2002) um framework é “um conjunto de classes que cooperam entre si provendo assim um projeto reutilizável para um domínio específico de classes de sistema” [GAM]. Um framework ou arcabouço é uma estrutura de suporte definida em que um outro projeto de software pode ser organizado e desenvolvido, quando se analisa o conceito no âmbito do desenvolvimento de software. Um framework pode incluir programas de suporte, bibliotecas de código, linguagens de script e outros softwares para ajudar a desenvolver e juntar diferentes componentes de um projeto de software. Os Frameworks são projetados com o propósito de facilitar o desenvolvimento de software, habilitando projetistas e programadores a gastarem mais tempo detalhando as exigências de negócio do software do que com detalhes tediosos de baixo nível do sistema.

### 3.1.12 Bootstrap

Um framework que disponibiliza configurações padrão para elementos de telas web. Sua intenção é que desenvolvedores que não possuem conhecimentos sólidos em técnicas de design consigam desenvolver páginas visualmente bonitas, sem grandes dificuldades (BOOTSTRAP, 2013). Cochran (2012) diz, que o Bootstrap utiliza em alguns de seus elementos, a biblioteca jQuery, já citada, para criar

animações, detectar ações do usuário, entre outras funções. Outra grande vantagem do Bootstrap, é seu sistema de grids (COCHRAN, 2012). Ele permite que o desenvolvedor programe suas aplicações para um padrão de tela, e automaticamente ou com poucas mudanças, elas sejam perfeitamente visualizadas em dispositivos de tamanhos diferentes, como celulares, tablets, monitores de tamanho pequeno, entre outros (BOOTSTRAP, 2013).

### 3.1.13 Spring

O Spring dispõe para o programador diversas tecnologias, que simplificam o desenvolvimento de código de infraestrutura. Neste projeto utilizamos algumas ferramentas que compõem este ecossistema de Frameworks os quais vamos listar resumidamente abaixo:

#### 3.1.13.1 Spring MVC

Algumas tarefas são comuns à maioria das aplicações web. Como exemplo podemos citar o mapeamento dos dados contidos na requisição para objetos da camada de modelo. O Spring MVC, um dos principais projetos dentro do Spring Framework, abstrai essa e outras rotinas de infraestrutura, permitindo ao programador se dedicar às regras de negócio da aplicação.

#### 3.1.13.2 Spring Data

Dentro da proposta de prover código de infraestrutura, o Spring Data unifica diferentes projetos com o objeto de facilitar o acesso a fontes de dados, como bancos de dados relacionais e não relacionais.

#### 3.1.13.3 Spring Boot

Gerenciar dependências pode ser uma dor de cabeça no início de qualquer projeto. Spring Boot permite iniciar rapidamente projetos Spring ao gerenciar as dependências do projeto a partir do princípio da Convenção sobre Configuração, que busca reduzir o número de decisões a serem tomadas pelo programador no momento de configurar as diferentes áreas de uma aplicação.

#### 3.1.13.4 Spring Security

Segurança é uma preocupação da maioria das aplicações web. Spring Security fornece um mecanismo customizável para autenticação e autorização, de forma a controlar o acesso aos dados gerenciados pelo sistema. Uma vez que ele é o mecanismo padrão de segurança em projetos Spring, seu conhecimento é recomendado aos desenvolvedores que utilizam seus projetos.

#### 3.1.14 Heroku

Para a hospedagem foi usado como apoio o Heroku, um serviço de plataforma (Platform as a Service - PaaS) de computação em nuvem conhecida no mercado, com a possibilidade de subir aplicações nas linguagens Ruby, Node.js, Java, Python, Clojure, Scala, Go e PHP. O Heroku foi escolhido pela facilidade em subir uma aplicação gratuitamente, o que possibilita testar e validar ideias básicas. O Heroku é um serviço que disponibiliza um ambiente computacional em nuvem e suporte para diversas linguagens de programação (DEGGES, 2012). Permite ao usuário criar e submeter aplicações desenvolvidas. Cada projeto é associado a um repositório remoto, sendo a principal forma de acesso através do Git. O código fonte, uma descrição das dependências utilizadas e um arquivo (Procfile) contendo as instruções que o Heroku deve executar para o software são passados como parâmetros (SOLÓRZANO; CHARÃO, 2017).

#### 3.1.15 Git

Segundo Santacroce (2015) o Git é um sistema de controle de versões (SCV) de arquivos distribuído, criado por Linus Torvalds, criador do Linux, que ajuda a controlar as modificações feitas em cada arquivo registrando quem modificou um arquivo, quando e o que foi modificado. Permitindo com essas informações ter um registro histórico de modificações, apontar (através do comando git blame) quem foi o responsável por uma modificação que possa ter sido errônea e fazer comparativos entre mudanças no arquivo, facilitando um processo de depuração. Assim utilizamos o git como controle de versionamento do código.

### 3.1.16 GitHub

O GitHub é uma empresa e site com o mesmo nome, oferecendo serviço de hospedagem de código grátis para projetos públicos (qualquer pessoa pode ver e copiar o código fonte) e projetos privados por um custo mensal. Nele, o Git está presente na base de todo o sistema pois é a maneira padrão e única disponível no serviço para hospedar o código, não à toa que está incluso no nome. O “hub” do nome vem da palavra da língua inglesa que reflete o centro efetivo de atividade de uma região ou rede, a ideia do nome reflete a união de pessoas e código em um local específico, o GitHub. Com características de colaboração aberta, o GitHub em 2018 é o maior site de repositórios de código online do mundo com 85 milhões de repositórios pelo mundo, segundo o próprio site, com diversos projetos administrados por contribuidores dos mais diversos lugares, apoio de empresas que incentivam o desenvolvimento de código aberto disponibilizando o próprio sistema no site, além de ter funcionários trabalhando ativamente nesses projetos. Porém, a grande maioria dos contribuidores são pessoas ao redor do mundo que voluntariamente interagem com os projetos e os ajudam a crescer ao contribuírem com código novo, faturamentos, comunicando problemas no código-fonte, ou criando issues. Utilizamos o GitHub como repositório remoto.

### 3.1.17 Astah Community

Astah Community foi utilizado para elaborar os diagramas do modelo UML. É um software para modelagem UML (Unified Modeling Language – Linguagem de Modelagem Unificada) com suporte a UML 2.

Astah Community disponibiliza para desenvolvimento, os diagramas de Classes, Casos de Uso, Sequência, Comunicação, Máquina de Estados, Atividade, Componentes, Implantação e Diagrama de Estrutura Composta (TECHTUDO, 2021).

## 4 DESENVOLVIMENTO

Neste capítulo será apresentada a Metodologia, Modelagem e Requisitos para o desenvolvimento do sistema.

### 4.1 METODOLOGIA

Para a resolução dos problemas encontrados primeiramente foram levantados alguns requisitos para a criação de um novo sistema e como o mesmo poderia ser utilizado no dia a dia, em sequência foi projetado uma ideia de solução, e que por final fora executado e aplicado na organização. O presente trabalho, pode ser classificado como uma pesquisa qualitativa com entrevista semiestruturada, sendo uma técnica importante para fazer a coleta de dados.

Segundo TRIVIÑOS (1987, p. 124):

A pesquisa qualitativa é conhecida também como "estudo de campo", "estudo qualitativo", "interacionismo simbólico", "perspectiva interna", "interpretativa", "etnometodologia", "ecológica", "descritiva", "observação participante", "entrevista qualitativa", "abordagem de estudo de caso", "pesquisa participante", "pesquisa fenomenológica", "pesquisa-ação", "pesquisa naturalista", "entrevista em profundidade", "pesquisa qualitativa e fenomenológica", e outras [...]. Sob esses nomes, em geral, não obstante, devemos estar alertas em relação, pelo menos, a dois aspectos. Alguns desses enfoques rejeitam total ou parcialmente o ponto de vista quantitativo na pesquisa educacional; e outros denunciam, claramente, os suportes teóricos sobre os quais elaboraram seus postulados interpretativos da realidade

As metodologias utilizadas consistem em coletar através de entrevistas informações sobre a necessidade que um determinado comércio apresenta com a falta de um sistema para auxiliar nas suas vendas, a fim de estudar aspectos relacionados ao tema de gerenciamento de ferramentas de software para comércio. Neste método foi buscada a aplicação prática dos conhecimentos para uma solução proposta que resultou no desenvolvimento de um sistema para gerenciamento de um comércio.

Para elaboração deste trabalho de conclusão analisou-se a história, funcionamento da empresa Cantinho da Costela e requisitos do sistema pedido. A partir dos dados levantados foi modelado o sistema para seu desenvolvimento, fazendo uso da linguagem de modelagem UML.

Foram escolhidas diversas tecnologias baseadas nas pesquisas bibliográficas como: Banco de dados PostgreSQL para gerenciamento dos dados, Java com *Spring* e *thymeleaf* para persistência dos dados.

#### 4.1.1 Estudo de Caso

Neste capítulo será apresentado o estudo de caso efetuado, e será descrito o método utilizado atualmente pela empresa para gestão do negócio e as falhas encontradas para execução da gestão.

##### 4.1.1.1 História da Empresa

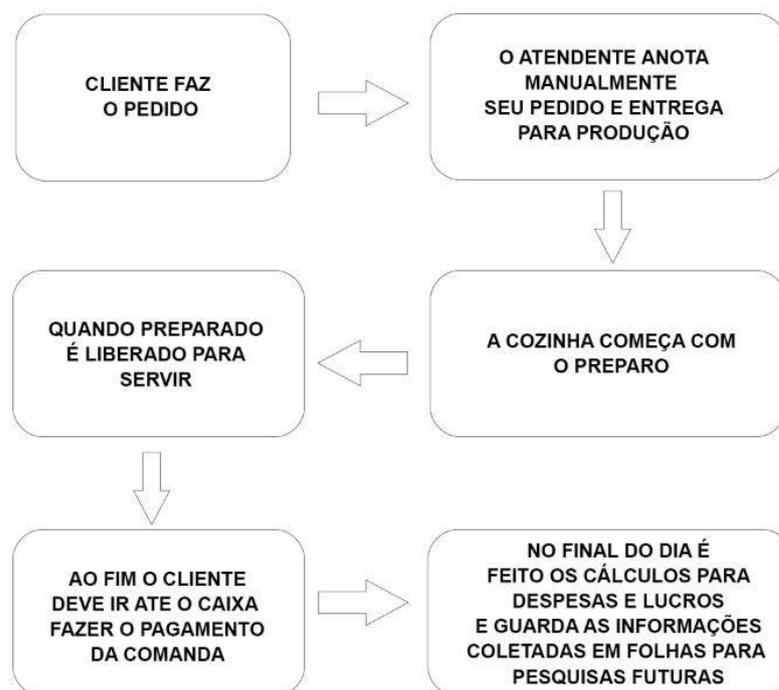
A empresa Cantinho da Costela surgiu no ano de 2014, localizada em Colombo PR. Ela é do setor de alimentos, sendo um restaurante que trabalha com delivery e com almoço no local. A empresa é de pequeno porte com apenas 18 funcionários. O dono costuma fazer as compras das mercadorias em vários locais e fornecedores.

##### 4.1.1.2 Gestão do Negócio

A empresa até o momento não possuía um sistema computacional, e toda a gestão do negócio se dava por processo manual. Com isso gerava inúmeros documentos impressos e escritos à mão. As contas a receber eram geradas pelo proprietário conforme ele escrevia ou memorizava seu valor, informando a cada cliente. Já as contas a pagar eram geradas feitas verificando com o fornecedor o total a pagar.

Todos os cálculos feitos a punho são sujeitos a falhas, sendo muito comum o fornecedor dos produtos dizer o total de um valor e a empresa ter como o valor a pagar sendo um valor divergente dependendo dos erros de cálculos.

A Figura 3 exibe o Fluxograma para realizar pedido, desde o atendimento até o pagamento.



**FIGURA 3 - FLUXOGRAMA PARA FAZER PEDIDO**

Fonte: Os Autores (2021)

#### 4.1.1.3 Funcionamento da Empresa

Até então a empresa funcionava da seguinte forma: os clientes realizam seus pedidos, o atendente anota em uma comanda, encaminha para a cozinha fazer os preparos. Quando pronto é chamado o atendente para liberar o pedido. Quando o cliente finaliza sua refeição ele se dirige ao caixa para pagamento da comanda, sendo que a mesma fica na mesa com cada cliente. Para pagamento da comanda é somado tudo na calculadora o que foi consumido, no final informa o valor e faz a cobrança. A comanda cobrada é descartada e assim se faz com os outros clientes. Após o fechamento do restaurante é feito a conta de quanto se tem em despesas e quanto deu de lucro e quais foram os gastos extras. Com isso em mãos ele paga o fornecedor e outras despesas, e fica com o lucro em sua conta.

#### 4.1.1.4 Problemas existentes na empresa

A empresa apresenta como principal problema a falta de gerenciamento. O proprietário da empresa não possui controle dos lucros da sua empresa.

Ele é responsável por tudo manualmente: vender, faturar, controlar estoques, entrada e saída de caixa. Devido à falta de formação do proprietário, seus métodos de controle são antigos, não fazendo uso de novos métodos e das tecnologias que facilitam a vida dos empresários. Outro problema enfrentado é a má gestão da empresa em controlar suas contas. Não se tem uma separação das contas Jurídica e Física, o proprietário acaba misturando todas as despesas.

## 4.2 LEVANTAMENTO DE REQUISITOS

A engenharia de requisitos se preocupa com a definição de objetivos de um sistema proposto, suas características, atributos, capacidades e fatores de qualidade para oferecer valor agregado ao cliente (YOUNG, 2003). Nesse contexto, o levantamento de requisitos deve ser visto como um precursor necessário de qualquer trabalho de desenvolvimento, mesmo que seja feito de forma rápida, informal ou em paralelo a outras atividades, nunca deve ser ignorado (ROBERTSON e ROBERTSON, 2012)

De acordo com Falbo e Barcellos (2011, pág 5 apud SOMMERVILLE, 2007): As descrições das funções que um sistema deve incorporar e das restrições que devem ser satisfeitas são os requisitos para o sistema.

Os requisitos de um sistema definem o que o sistema deve fazer e as circunstâncias sob as quais deve operar. Em outras palavras, os requisitos definem os serviços que o sistema deve fornecer e dispõem sobre as restrições à operação do mesmo. Os requisitos podem ser divididos em Funcionais e Não Funcionais. Os Requisitos Funcionais podem indicar o que o sistema deve fazer e como o sistema deve agir em determinadas situações. Já os Requisitos Não Funcionais descrevem as restrições do sistema, com tempo, uso de recursos entre outros (Falbo e Barcellos 2011, pág 5, 6 apud SOMMERVILLE, 2007).

### 4.2.1 Requisitos Funcionais

Os requisitos de determinado sistema são, basicamente, as descrições do que ele deve fazer, que tipos de serviços este oferece, bem como as restrições a seu funcionamento.

Tais requisitos não são simplesmente impostos ou presumidos, e sim, são artefatos que refletem as necessidades e expectativas dos clientes e demais interessados com relação ao sistema ao qual eles pertencem (SOMMERVILLE, 2011).

Quando se fala do ato de descobrir, analisar, documentar e verificar os requisitos de um software, está-se tratando da subárea da engenharia de software denominada engenharia de requisitos (*Requirements Engineering* - RE) (SOMMERVILLE, 2011)

Seguem abaixo os principais Requisitos Funcionais desenvolvidos pelo Sistema de Gestão Integrado Comercial (SIGIC):

- **RF1.** O sistema deve manter um cadastro de cliente.
- **RF2.** O sistema deve manter um cadastro de funcionários
- **RF3.** O sistema deve manter um cadastro de fornecedores
- **RF4.** O sistema deve possuir um controle de estoque dos produtos
- **RF5.** O sistema deve manter um cadastro dos produtos vendidos pela empresa.
- **RF6.** O sistema permitirá visualizar as contas a receber das vendas.
- **RF7.** O sistema deve possuir uma página para fazer pedidos.
- **RF8.** O sistema deverá dispor de mecanismos de buscas para os produtos.
- **RF9.** O sistema deve manter um cadastro dos produtos comprados pela empresa
- **RF10.** O sistema deve possuir um relatório de fechamento de caixa

#### 4.2.2 Requisitos Não Funcionais

Os requisitos não funcionais descrevem o sistema em termos de desempenho, confiança, usabilidade e tecnologia. São características que não são visíveis aos usuários, mas são importantes para o desenvolvedor e podem ser mais críticos que requisitos funcionais, pois se não forem satisfeitos podem tornar o sistema inútil. De acordo, com a natureza dos requisitos não funcionais, iremos destacar os que foram utilizados na aplicação, conforme listados abaixo:

- **RNF 1.** O sistema utilizará design responsivo, para os dispositivos compatíveis.
- **RNF 2.** O sistema deverá utilizar tecnologias compatíveis aos padrões atuais da computação.
- **RNF 3.** Compatibilidade com sistemas operacionais Windows e Linux

### 4.3 DIAGRAMAS DA UML

O *Unified Modeling Language* (UML) é uma linguagem visual para modelagem de softwares baseados em orientação a objetos. É uma linguagem de propósito geral e pode ser usada em todos os estágios do desenvolvimento da aplicação. Atualmente é utilizada em todo o mundo pelas empresas de desenvolvimento de softwares. A UML não é uma linguagem de programação e sim uma linguagem de modelagem, cujo propósito é auxiliar os desenvolvedores na construção da aplicação (GUEDES, 2011).

A UML surgiu da união de 3 métodos: o método de Booch, o método *Object Modeling Technique* (OMT) e o método *Object-Oriented Software Engineering* (OOSE). A união desses três métodos começou em 1990 e só foi concluída em 1996. A versão 2.0 da UML foi lançada em 2005 e atualmente encontra-se na 2.3 Beta (GUEDES, 2011). Segundo Guedes (2011), a UML possui diversos diagramas para uma visão ampla de todo o sistema. Cada diagrama demonstra uma parte do sistema, como se o sistema fosse modelado por camadas. Dentre eles temos: O diagrama de classes; O diagrama de caso de uso; O diagrama de objeto; O diagrama de pacote; O diagrama de sequência; O diagrama de comunicação; O diagrama de máquina de estado; O diagrama de atividade; O diagrama de visão geral de interação; O diagrama de componentes; O diagrama de implantação; O diagrama de estrutura composta; O diagrama de tempo. A UML é importante para este projeto de conclusão, pois é necessário em qualquer projeto pelo mundo, arquitetar o que será desenvolvido e para isso será usado a UML na criação dos diagramas de classes, diagramas de casos de uso e diagramas de sequência. Também podemos citar que será feita a descrição dos casos de uso.

#### 4.3.1 Diagrama de Casos de Uso

O diagrama de casos de uso tem como objetivo especificar como o sistema funcionará do ponto de vista do usuário e é fundamental para a visualização simplificada das funções dos atores no programa. Como um caso de uso define uma visão do sistema de acordo com a interação com agentes internos, é este fato que o diagrama de casos de uso da UML tenta expressar através de notação gráfica (PRESSMAN, 2011).

Ele não representa de maneira completa o funcionamento do caso de uso, que só pode ser trabalhado sob uma perspectiva textual. Um diagrama de caso de uso é uma visão geral de todos os casos de uso de determinado sistema e como estes se relacionam, fornecendo uma visão geral das funcionalidades do sistema (PRESSMAN, 2011).

Linhas são utilizadas para “ligar” os atores aos casos de uso, representando assim um relacionamento entre eles (LIMA, 2007), conforme Figura 4.

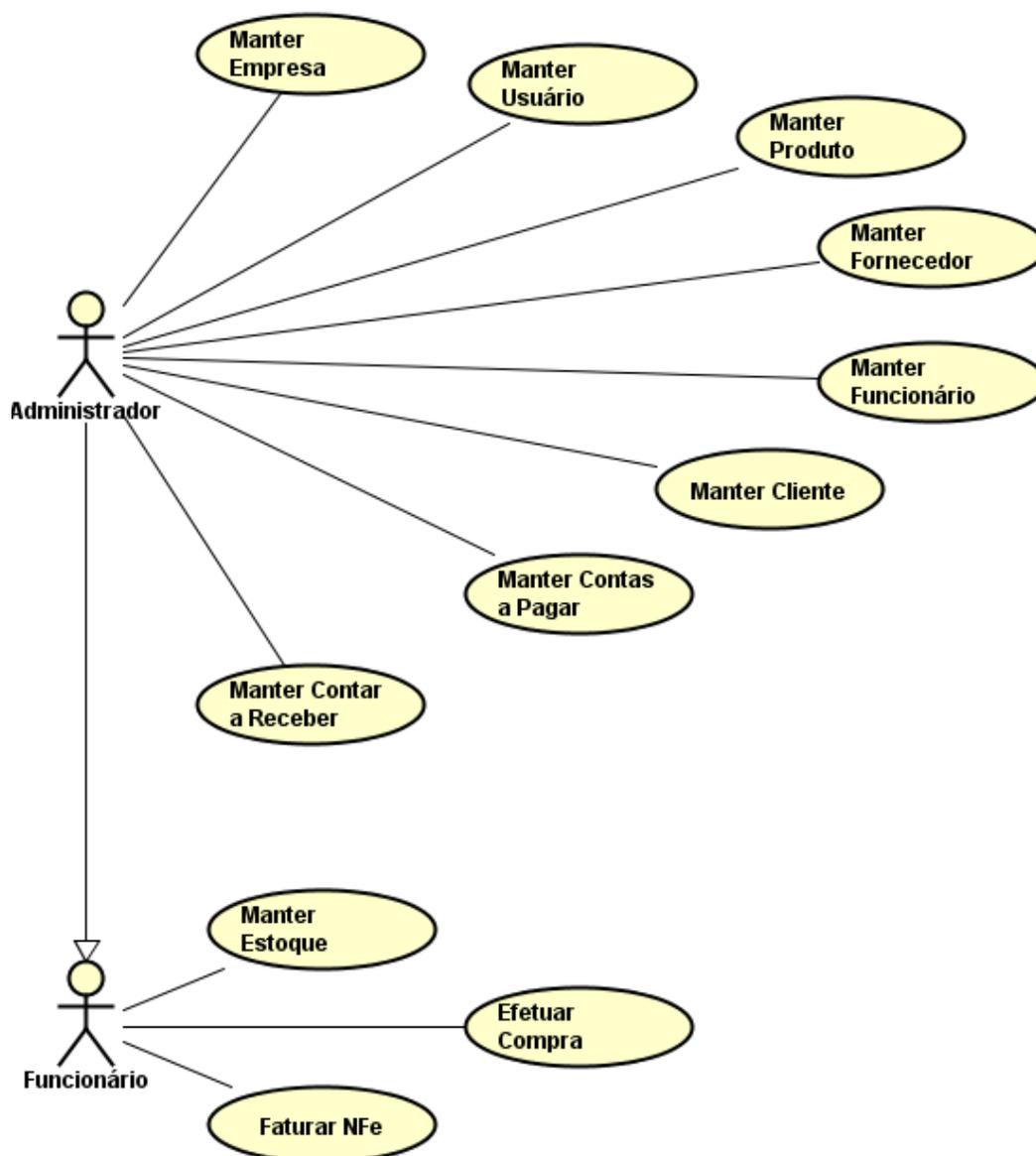


FIGURA 4 - CASO DE USO  
FONTE: Os Autores (2021)

Um software para trabalhar com diagramas e esquemas. O software permite a criação de projetos de complexidade variados, tais como: banco de dados de diagramas, esquemas de árvores, a estrutura, redes e diagramas de fluxo etc.

Abaixo serão descritos os casos de uso através dos quadros apresentados. A descrição dos casos de uso tem como objetivo detalhar as funções do autor com os casos de uso.

#### 4.3.1.1 Caso de Uso Fazer Login

O usuário efetua o login, define uma senha, quando validado o login, irá atualizar permissões para acesso ao sistema, após a validação o mesmo terá acesso ao sistema, conforme Quadro 1.

QUADRO 1 – DETALHAMENTO DO CASO DE USO FAZER LOGIN

Documentação de Caso de Uso	
<b>Caso de Uso</b>	Login
<b>Ator que o Invoca</b>	Usuario
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Case: efetuar Login	2. Validar Login
3. Definir uma senha	4. Segurança do sistema
5. Autoriza ou revoga permissões	6. Atualiza Permissões para acesso ao sistema
	7. Informa permissões liberadas
Cenários Alternativos	
Passo	Descrição
1	Após validação o mesmo terá acesso ao sistema.

FONTE: Os Autores (2021)

#### 4.3.1.2 Caso de Uso Pedido de Venda

Realizado o login ao sistema, usuário escolhe a opção efetuar pedido de venda, sistema exibe a tela para inserir os itens, usuário insere os dados solicitados, sistema valida dados inseridos, clica em salvar, sistema retorna a mensagem de venda realizada, conforme Quadro 2.

QUADRO 2 – DETALHAMENTO DO CASO DE USO PEDIDO DE VENDA DO PRODUTO

Documentação de Caso de Uso	
<b>Caso de Uso</b>	Pedido de venda
<b>Ator que o Invoca</b>	Usuario
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Realiza login no sistema	2. Usuario escolhe opção efetuar pedido de venda
3. Sistema exibe a tela para inserir os itens	4. Usuario insere dados solicitados para realizar a venda
5. Sistema valida dados inseridos	6. Usuario clica em salvar
7. Sistema exibe mensagem de "Venda realizada "	
Cenários Alternativos	
Passo	Descrição
1	Usuario comum não pode excluir o pedido

FONTE: Os Autores (2021)

#### 4.3.1.3 Caso de Uso Pedido de Compra

Sistema disponibiliza ações disponíveis, funcionário escolhe a opção efetuar pedido de compra, sistema exibe a tela para inserir os itens, inserir os dados solicitados para realizar a compra, sistema valida os dados inseridos, clicar em salvar. Sistema retorna a mensagem compra realizada, conforme Quadro 3.

QUADRO 3 – DETALHAMENTO DO CASO DE USO PEDIDO DE COMPRA

Documentação de Caso de Uso	
<b>Caso de Uso</b>	Pedido de compra
<b>Ator que o Invoca</b>	Comprador
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Realiza login no sistema	2. Sistema disponibiliza ações disponíveis
3. Funcionario escolhe opção efetuar pedido de compra	4. Sistema exibe a tela para inserir os itens
5. Funcionario insere dados solicitados para realizar a compra	6. Sistema valida dados inseridos
7. Funcionario clica em salvar	8. Sistema exibe mensagem de "Compra realizada"
Cenários Alternativos	
Passo	Descrição

FONTE: Os Autores (2021)

#### 4.3.1.4 Caso de Uso Histórico de Vendas

Logar com usuário, confirmar o login, consultar por data, pesquisar, sistema retornará com produtos encontrados, conforme Quadro 4.

QUADRO 4 – DETALHAMENTO DO CASO DE USO HISTÓRICO DE VENDAS

Documentação de Caso de Uso	
<b>Caso de Uso</b>	Histórico de vendas
<b>Ator que o Invoca</b>	Administrador
Cenário Principal	
Ações do Ator	Ações do Sistema
1. login usuario	2. confirma login
3. Consulta por data	4. Pesquisar
5. Produto encontrados	
Cenários Alternativos	
Passo	Descrição
1	Caso não informado o periodo da venda , retornar não localizado venda para este produto.

FONTE: Os Autores (2021)

#### 4.3.1.5 Caso de Uso Cadastrar Funcionário

Cadastrar funcionário, alterar e remover funcionário, autorizado funcionário, consulta funcionário, funcionário ativo, conforme Quadro 5.

QUADRO 5 – DETALHAMENTO DO CASO DE USO CADASTRO DE FUNCIONÁRIO

Documentação de Caso de Uso	
<b>Caso de Uso</b>	Cadastro funcionario
<b>Ator que o Invoca</b>	Funcionario
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Cadastrar Funcionario	2. Alterar e remover Funcionario
3. Autorizado Funcionario	4. Consulta Funcionario
5. Funcionario Ativo	
Cenários Alternativos	
Passo	Descrição

FONTE: Os Autores (2021)

#### 4.3.1.6 Caso de Uso Cadastrar Fornecedor

Cadastrar fornecedor, alterar e remover fornecedor, autorizar para pedidos, controlar compras e vendas, autorizar ou bloquear fornecedor e cliente liberado, conforme Quadro 6.

**QUADRO 6 – DETALHAMENTO DO CASO DE USO CADASTRO DE FORNECEDOR**

Documentação de Caso de Uso	
<b>Caso de Uso</b>	Cadastrar
<b>Ator que o Invoca</b>	Fornecedor
Cenário Principal	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Case: Cadastrar Fornecedor	2. Alterar e remover fornecedor
3. Autorizado para pedidos	4. Controlar compras e vendas
5. Autoriza ou bloquea fornecedor	06/03/20
	7. Cliente liberado
Cenários Alternativos	
<b>Passo</b>	<b>Descrição</b>
1	Após validação o cliente estará cadastrado em seu sistema, informando sua situação financeira.

FONTE: Os Autores (2021)

#### 4.3.1.7 Caso de Uso Consulta Fornecedor

Efetuar login no sistema, selecionar consulta fornecedor, sistema exibe a tela dos fornecedores cadastrados, usuário pode utilizar a opção de busca para filtrar resultados, clicar em pesquisar, sistema valida dados inseridos, sistema exibe o conteúdo da busca, conforme Quadro 7.

**QUADRO 7 – DETALHAMENTO DO CASO DE USO CONSULTA DO FORNECEDOR**

Documentação de Caso de Uso	
<b>Caso de Uso</b>	Fornecedor
<b>Ator que o Invoca</b>	Usuario
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Usuario Realiza login no sistema	2. Usuario seleciona consulta Fornecedor
3. Sistema exibe a tela dos Fornecedores cadastrados	4. Usuario pode utilizar a opção de busca para filtrar o resultado
5. Usuario clica em pesquisar	6. Sistema valida dados inseridos
7. Sistema exibe o conteúdo da busca	
Cenários Alternativos	
Passo	Descrição
1	Caso o fornecedor não seja localizado, exibe a mensagem de funcionario não cadastrado

FONTE: Os Autores (2021)

#### 4.3.1.8 Caso de Uso Controle de Estoque

Realizar login, selecionar consulta produtos, sistema exibe a tela dos produtos cadastrados, usuário pode utilizar a opção de busca para filtrar o resultado dos produtos, clicar em pesquisar, sistema valida dados inseridos, sistema exibe o conteúdo da busca, sistema exibe estoque atual, conforme Quadro 8.

## QUADRO 8 – DETALHAMENTO DO CASO DE USO CONTROLE DE ESTOQUE

Documentação de Caso de Uso	
<b>Caso de Uso</b>	Controle de Estoque
<b>Ator que o Invoca</b>	Usuario
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Usuario Realiza login no sistema	2. Usuario seleciona consulta produtos
3. Sistema exibe a tela dos produtos cadastrados	4. Usuario pode utilizar a opção de busca para filtrar o resultado dos produtos
5. Usuario clica em pesquisar	6. Sistema valida dados inseridos
7. Sistema exibe o conteúdo da busca	8. Sistema exibe estoque atual
Cenários Alternativos	
Passo	Descrição
1	Caso o item não seja cadastro, exibe a mensagem para cadastrar item.

FONTE: Os Autores (2021)

### 4.3.1.9 Caso de Uso Consulta Cliente

Realizar login, usuário seleciona consulta cliente, sistema retorna com a tela dos clientes cadastrados, usuário pode utilizar busca por filtro, clicar em pesquisar, sistema valida dados inseridos, sistema exibe o conteúdo da busca.

Caso não seja localizado exibe a mensagem de cliente não cadastrado, conforme Quadro 9.

QUADRO 9 – DETALHAMENTO DO CASO DE USO CONSULTA DO CLIENTE

Documentação de Caso de Uso	
<b>Caso de Uso</b>	Consulta cliente
<b>Ator que o Invoca</b>	Usuário
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Realiza login no sistema	2. Usuário seleciona consulta cliente
3. Sistema exibe a tela os clientes cadastrados	4. Usuário pode utilizar a opção de busca para filtrar o resultado
5. Usuário clica em pesquisar	6. Sistema valida dados inseridos
7. Sistema exibe o conteúdo da busca	
Cenários Alternativos	
Passo	Descrição
1	Caso o cliente não seja localizado, exibe a mensagem de cliente não cadastrado

FONTE: Os Autores (2021)

#### 4.3.1.10 Caso de Uso Cadastrar Cliente

Cadastrar cliente, alterar e remover clientes, acesso a lista, mostrar as pendências, autoriza ou revoga permissões, atualiza quando pendências forem resolvidas, cliente liberado, conforme Quadro 10.

QUADRO 10 – DETALHAMENTO DO CASO DE USO CADASTRO DO CLIENTE

Documentação de Caso de Uso	
<b>Caso de Uso</b>	Cadastrar
<b>Ator que o Invoca</b>	Clientes
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Case: Cadastrar Cliente	2. Alterar e remover cliente
3. Acesso a lista	4. Mostrar pendencias financeiras
5. Autoriza ou revoga permissoes	6. Atualiza quando pendencias forem resolvidas
	7. Cliente liberado
Cenários Alternativos	
Passo	Descrição
1	Após validação o cliente estará cadastrado em seu sistema, informando sua situação financeira.

FONTE: Os Autores (2021)

#### 4.3.1.11 Caso de Uso Autoriza ou Revoga Novos Usuários

Efetuar login, validar login e mostrar as opções de administração, verificar pedidos de usuários, mostrar pedidos de usuários, autorizar ou revogar permissões, atualizar permissões, informar quais permissões foram alteradas, conforme Quadro 11.

QUADRO 11 – DETALHAMENTO DO CASO DE USO AUTORIZA NOVOS USUÁRIOS

Documentação de Caso de Uso	
<b>Caso de Uso</b>	Autoriza ou revoga novos usuarios
<b>Ator que o Invoca</b>	Administrador
Cenário Principal	
Ações do Ator	Ações do Sistema
1. Case: efetuar Login	2. Validar Login e mostrar opções de administração
3. Verificar pedidos de usuarios	4. Mostra pedidos de usuarios
5. Autoriza ou revoga permissoes	6. Atualiza Permissoes
	7. Informa quais permissoes foram alteradas
Cenários Alternativos	
Passo	Descrição
1	Caso seja o usuario alvo seja outro administrador, notifique-o e não altere permissoes

FONTE: Os Autores (2021)

#### 4.3.2 Diagrama de Classes

O Diagrama de Classes é fundamental na modelagem de um Software, como demonstra Melo (2010), afirmando que ele é a estrela principal de um sistema orientado a objetos.

Tal diagrama tem como principal finalidade detalhar as classes pertencentes ao modelo e identificar seus relacionamentos. Segundo Booch, Rumbaugh e Jacobson (2005, p. 51) “classe é uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.”

Cada classe possui seus atributos e operações.

Os atributos representam alguma propriedade da classe na qual ele está inserido, uma classe não necessariamente precisa ter 31 atributos.

Já uma operação é uma atividade que pode ser realizada por um objeto da classe na qual esta operação está inserida.

Como afirmam os autores Booch, Rumbaugh e Jacobson (2005) e Melo (2010), a maioria das classes não trabalham sozinhas, elas precisam se relacionar, colaborando uma com a outra. Os primeiros autores defendem a ideia de que existem três tipos de relacionamentos na modelagem orientada a objetos: dependências, generalizações e associações.

A seguir será apresentado o Diagrama de Classes, que será usado para o desenvolvimento do sistema, demonstrado na Figura 5. Todos os métodos relativos as Classes foram suprimidos e fazem parte da *Create Read Update Delete* (CRUD) adicionados do método Print (Imprimir) para cada uma das Classes criadas para o sistema.

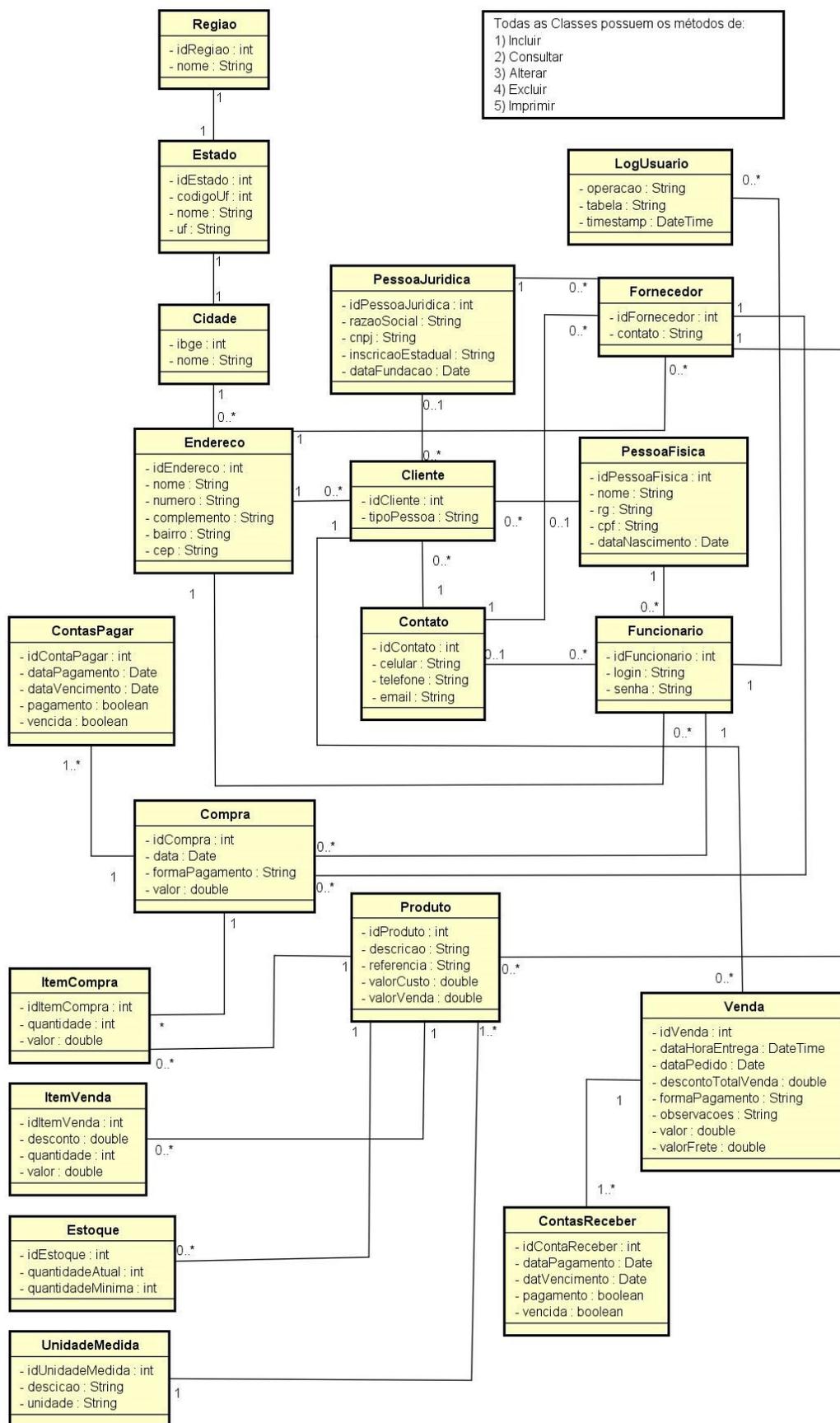


FIGURA 5 - DIAGRAMA DE CLASSES  
 FONTE: Os Autores (2021)

## 4.4 MODELAGEM DO SISTEMA

Primeiramente é importante salientar a importância da modelagem. Segundo os autores Booch, Rumbaugh e Jacobson (2005, p. 3), “uma empresa de Software bem-sucedida é aquela que fornece Software de qualidade e capaz de atender às necessidades dos respectivos usuários.” Estes autores defendem que a modelagem tem papel fundamental nas atividades que levam a implantação de um bom Software.

Para a modelagem do sistema foi marcado uma reunião com o representante da empresa e foram vistos pontos relevantes para colocar na aplicação. Nesta seção apresenta-se o diagrama que ilustra o funcionamento e estrutura do software.

### 4.4.1 Diagrama Entidade Relacionamento

O modelo entidade-relacionamento (modelo ER) tornou-se um padrão para a modelagem conceitual, segundo Heuser (2009).

Uma das principais vantagens dessa proposta é a técnica de diagramação, que permite representar de forma simples os principais aspectos do projeto de um BD. Basicamente, o modelo baseia-se em entidades e relações.

Uma entidade representa um fato do mundo real e seus dados são valores de propriedades associadas a esse fato.

Já as relações, são representações da forma como uma entidade se associa a outras entidades. Heuser (2009) aponta que os principais conceitos da abordagem ER são: entidade, relacionamento, atributo, generalização/especialização e entidade associativa.

A Figura 6 exibe o Diagrama Entidade Relacionamento do Sistema de Gestão Integrado Comercial (SIGIC) com as ligações entre as tabelas e seus atributos: chaves primárias e estrangeiras.

No APÊNDICE C foi inserido o script de criação dos Banco de Dados SIGIC e a criação das tabelas e seus relacionamentos.

No APÊNDICE D foi criado o Diagrama Entidade Relacionamento para uma melhor visualização das tabelas e seus relacionamentos.

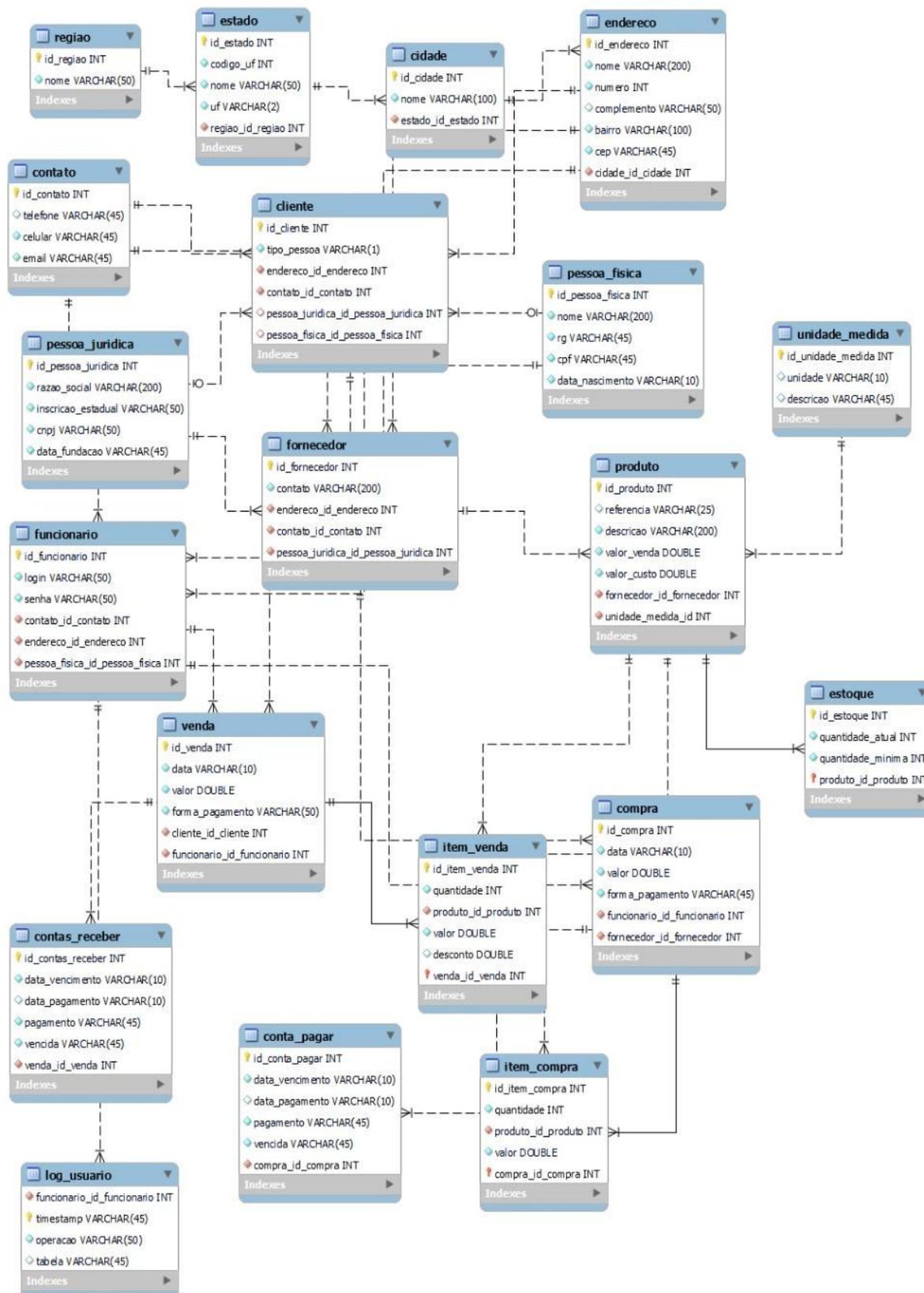


FIGURA 6 - DIAGRAMA DE ENTIDADE E RELACIONAMENTO  
 FONTE: Os Autores (2021)

## 5 RESULTADOS

O sistema foi implementado com sucesso utilizando as tecnologias apresentadas. A seguir serão apresentadas as principais telas desenvolvidas para o Sistema de Gestão Integrado Comercial: SIGIC.

Ao digitar o endereço URL, [www.sigic.com.br](http://www.sigic.com.br), o cliente será direcionado para a primeira tela do sistema para informar o Login e Senha, que serão validados no Banco de Dados, conforme apresentado na Figura 7.



A tela de login do SIGIC apresenta o logo da empresa no topo, seguido pelo texto "Seja Bem Vindo(a)!". Abaixo, há dois campos de entrada para "Login" e "Senha", e um botão azul arredondado com o texto "Login".

FIGURA 7 - TELA LOGIN  
FONTE: Os Autores (2021)

A segunda página do sistema tem como padrão o *Dashboard* com as seguintes opções: Cadastros, Produtos, Vendas, Compras, Logística, Faturamento, Financeiro e Sistema, conforme apresentado na Figura 8.

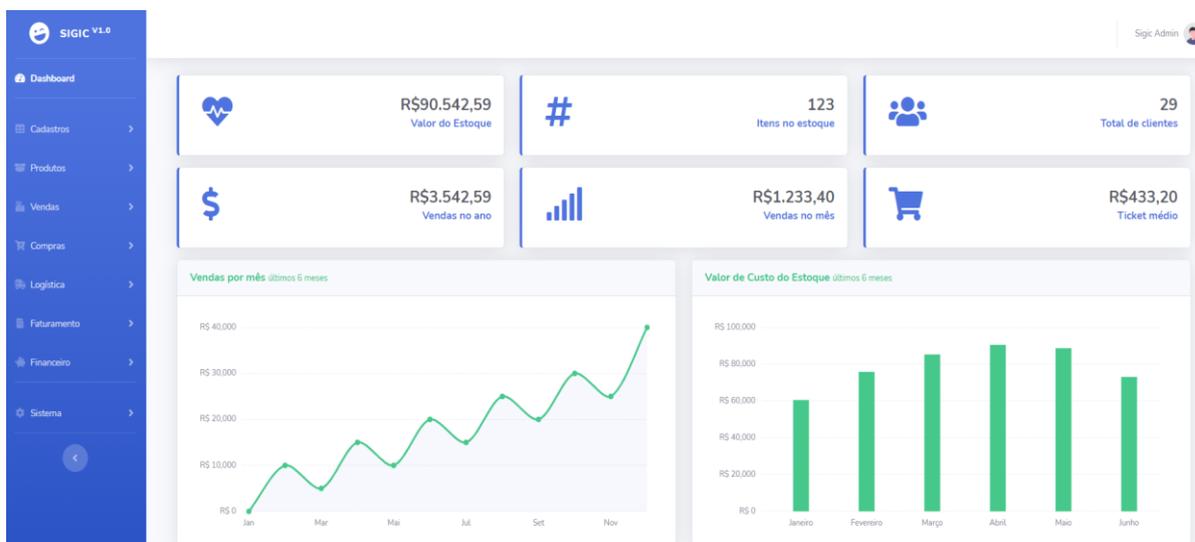


FIGURA 8 - TELA DASHBOARD  
FONTE: Os Autores (2021)

Na tela de Cadastros são realizados o cadastro e a manutenção de clientes, funcionários, fornecedores e produtos, conforme apresentado na Figura 9.

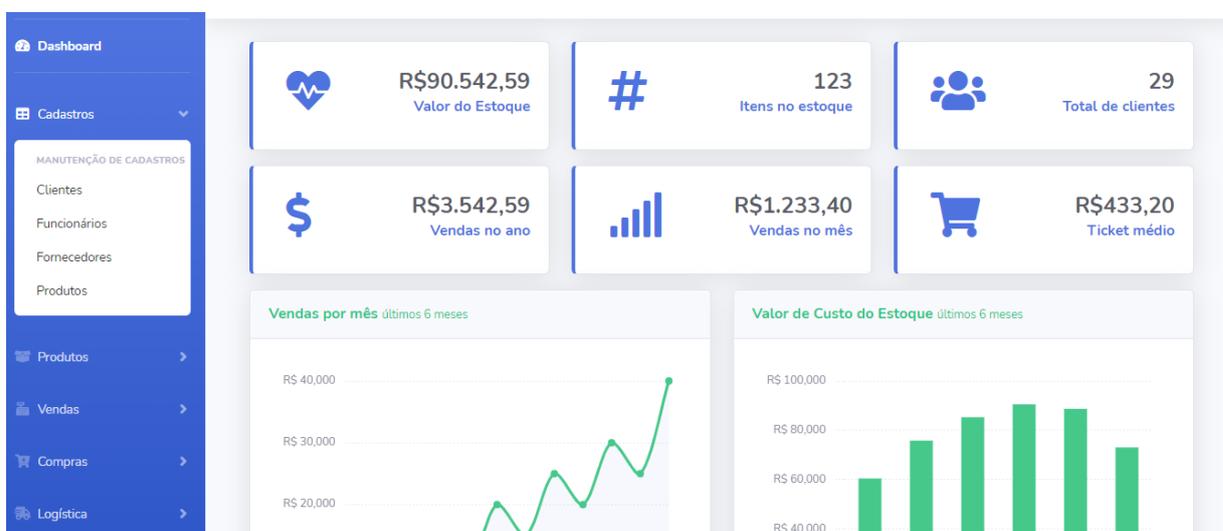


FIGURA 9 - TELA CADASTRO  
FONTE: Os Autores (2021)

Na tela de cadastro, navegar até o fornecedor é direcionado a tela do novo fornecedor, aqui você consegue listar os fornecedores cadastrados, conforme apresentado na Figura 10.

FIGURA 10 - TELA CADASTRO FORNECEDOR  
FONTE: Os Autores (2021)

Na tela de cadastro, navegar até o cliente é direcionado a tela do novo cliente, aqui você consegue listar os clientes cadastrados (Figura 11).

Novo Cliente

Pessoa Física  Pessoa Jurídica

Nome

CPF RG Data de Nascimento

Endereço

Cep

Endereço Complemento Número

Bairro Seleccione o Estado... Seleccione a Cidade...

Contato

E-mail

Telefone Celular Telefone Fixo

Salvar

FIGURA 11 - TELA CADASTRO CLIENTES  
FONTE: Os Autores (2021)

Na tela de cadastro, navegar até o produto é direcionado a tela de novo produto, aqui você consegue cadastrar e listar os produtos (Figura 12).

Sigic Admin

Novo Produto

Listar Produtos

Referência Descrição

Seleccione o Fornecedor...

Unidade de Medida... Valor de Venda Valor de Custo

Salvar

Cancelar

FIGURA 12 - TELA CADASTRO PRODUTOS  
FONTE: Os Autores (2021)

No menu produtos, navegar até ajuste de estoque é direcionado a tela para ajustar o estoque, aqui você consegue alterar seu estoque. (Figura 13).

FIGURA 13 - TELA AJUSTAR ESTOQUE

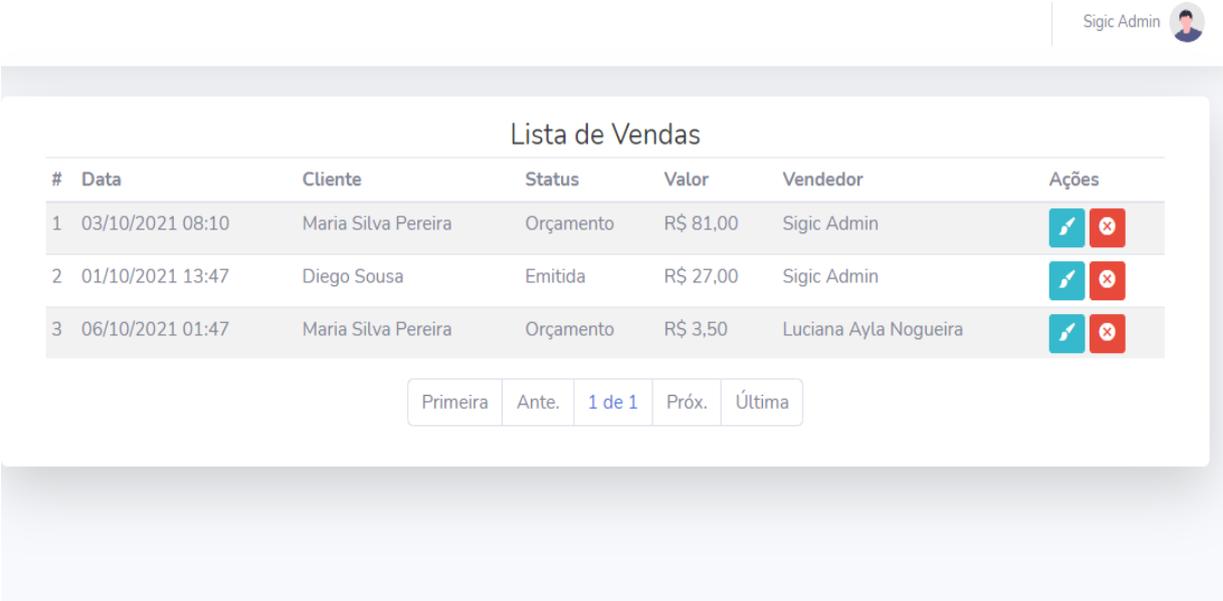
FONTE: Os Autores (2021)

No menu vendas, navegar até efetuar vendas é direcionado a tela para realizar a venda, aqui você consegue incluir a venda do cliente, endereço para entrega, incluir frete e obter desconto caso necessário (Figura 14).

FIGURA 14 - TELA EFETUAR VENDAS

FONTE: Os Autores (2021)

No menu vendas, navegar até listar vendas é direcionado a tela da lista de vendas, aqui você consegue verificar as vendas realizadas, conforme apresentado na Figura 15.



Lista de Vendas

#	Data	Cliente	Status	Valor	Vendedor	Ações
1	03/10/2021 08:10	Maria Silva Pereira	Orçamento	R\$ 81,00	Sigic Admin	 
2	01/10/2021 13:47	Diego Sousa	Emitida	R\$ 27,00	Sigic Admin	 
3	06/10/2021 01:47	Maria Silva Pereira	Orçamento	R\$ 3,50	Luciana Ayla Nogueira	 

Primeira Ante. 1 de 1 Próx. Última

FIGURA 15 - TELA LISTA DE VENDAS  
FONTE: Os Autores (2021)

O projeto foi submetido para apresentação na VI Mostra de Ensino, Extensão, Pesquisa e Inovação, Campus Colombo, sendo aceito. No APÊNDICE E foi inserido o certificado do trabalho intitulado SIGIC: SISTEMA DE GESTÃO INTEGRADO COMERCIAL.

## 6 CONCLUSÃO

O presente estudo teve como objetivo o desenvolvimento de um ERP para pedidos da empresa Cantinho da Costela. A aplicação apresentada buscou reduzir a processo manual e otimizar o controle de pedidos. Após a implementação da aplicação buscamos um resultado positivo, pois com a aplicação a empresa terá controle mais efetivo de suas compras e venda, controle dos clientes e de suas compras a pagar ou a receber.

O desenvolvimento da aplicação deu-se pela integração de variadas tecnologias, tais como: a plataforma Java EE, *Spring* com *Thymeleaf* e o banco de dados objeto relacional PostgreSQL

Destaca-se que as atividades de levantamento de requisitos e coleta de dados, foram realizadas juntamente com o dono da empresa para entender melhor o funcionamento e métodos utilizados na empresa. Seguidamente realizou-se a modelagem de acordo com o estudo de caso realizado anteriormente. Por fim, realizou-se o desenvolvimento contendo métodos específicos para realizar as operações necessárias com o objetivo de facilitar os processos da geração de pedidos da empresa.

O produto tem capacidade comercial, desde que sejam adaptados outros módulos de acordo com a necessidade do cliente e sejam desenvolvidos para agregar maior valor e facilitar a negociação com potenciais compradores. Portanto, o desenvolvimento do sistema não cessará com o término deste trabalho, seguindo o objetivo de construir um sistema completo para a comercialização em pequenas e médias empresas.

## REFERÊNCIAS

ACCIOLY, Anthony. Satisfação imediata com o Banco de Dados H2. Disponível em: <http://a.accioly.7rtc.com/>. 2011.

BOOTSTRAP, T. Twitter bootstrap. 2013. Disponível em: <<https://getbootstrap.com/>>

BOOCH, Grady; James RUMBAUGH; e Ivar, JACOBSON. UML: guia do usuário. 6ª Reimpressão. Rio de Janeiro: Elsevier, 2005.

COCHRAN, D. Twitter Bootstrap Web Development How-To. Packt Publishing, Limited, 2012. ISBN 9781849518833. Disponível em: <<https://books.google.com.br/books?id=7LjFN1wXH8QC>>

DEGGES, R. The Heroku Hacker's Guide. [S.l.]: Release, 2012.

DEITEL, Paul; DEITEL, Harvey. Java: como programar. 8. ed. São Paulo, SP: Pearson, 2010. 1144 p.

FALBO, Ricardo de Almeida; BARCELLOS, Monalessa Perini. Engenharia de Software. Espírito Santo: Ufes - Universidade Federal do Espírito Santo, 2011. Disponível em: . Acesso em: 25 jan. 2019

FERREIRA, S. Guia Prático de HTML5:. Universo dos Livros, 2011. ISBN 9788579303944. Disponível em: <<https://books.google.com.br/books?id=gV0icqoRsgC>>

GAMMA, Eric, HELM, Richard, JOHNSON, Ralph, VLISSIDES, John. Padrões de Projeto: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2002.

GEREMIAS, Felipe; PAPIOR, Luiz Henrique. Resumo de TCC "Desenvolvimento de um sistema ERP com foco nas tecnologias de software livre / código aberto". UNISUL. 2006.

GONCALVES, Antonio. Introdução à plataforma Java (TM) EE 6 com o glassFish (TM) 3. 2. ed. Rio de Janeiro: Ciência Moderna, 2011. 563 p.

GUEDES, Gilleanes T. A. UML 2: uma abordagem prática. 2. ed. São Paulo, SP: Novatec, 2011. 484 p.

JACOBSON, Ivar. Use Cases and Aspects - working Seamlessly together. Rational Software Corporation. In: Jornal of Object technology - Chair of Software Engineering

JETBRAINS. IntelliJ Idea Ultimate. 2021. Disponível em: <<https://www.jetbrains.com/idea/?fromMenu>>

LIMA, A. D. S. UML 2.0: do requisito à solução. 2. ed. São Paulo: Érica, 2007.

MAZZA, L. HTML5 e CSS3: Domine a web do futuro. Casa do código, 2013. ISBN 9788566250053. Disponível em: <<https://www.casadocodigo.com.br/products/livro-html-css>>.

MELO, Ana Cristina. Desenvolvendo aplicações com UML 2.2 do conceito à implementação. 3º edição. Rio de Janeiro: Brasport, 2010.

MORRISON, M. Use a cabeça: JavaScript. Alta Books, 2008. (Use a cabeça!). ISBN 9788576082132. Disponível em:<<https://books.google.com.br/books?id=oKZtPgAACAAJ>>

MURPHY, C. et al. Beginning HTML5 and CSS3: The Web Evolved. Apress, 2012. (Apresspod Series). ISBN 9781430228745. Disponível em:<<https://books.google.com.br/books?id=IP29IjBc6ooC>>

PEREIRA NETO, Álvaro. PostgreSQL técnicas avançadas: versões Open Source 7.x e 8.x: soluções para desenvolvedores e administradores de bancos de dados. 4. ed. São Paulo, SP: Érica, 2007. 284 p.

PRESSMAN, R. S. Engenharia de Software: uma abordagem profissional. 7. ed. Porto Alegre: McGraw-Hill, 2011.

POSTGRESQL. About. Disponível em < <https://www.postgresql.org/about/>> Acesso em 12 de julho de 2021.

POWERS, S. Learning JavaScript: Add Sparkle and Life to Your Web Pages. O'Reilly Media, 2008. ISBN 9780596554378. Disponível em:<<https://books.google.com.br/books?id=3On-moJDuO0C>>

ROBERTSON, S.; ROBERTSON, J. Mastering the Requirements Process. 3ª. ed. Westford: Pearson, 2012.

SANTACROCE, Ferdinando. Git Essentials. 1. ed.: O'Reilly, 2015. 160 p.

SILVA, M. JQUERY - A BIBLIOTECA DO PROGRAMADOR JAVASCRIPT. NOVATEC, 2008. ISBN 9788575221785. Disponível em: <<https://books.google.com.br/books?id=VafU5WLalwwC&hl=pt-BR>>

SOLÓRZANO, A. L. V.; CHARÃO, A. S. Explorando a plataforma de computação em nuvem heroku para execução de programas paralelos com o penmp. In: SBC. Anais da XVII Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul. [S.l.], 2017.

SOMMERVILLE, I. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

TECHTUDO. Modele softwares com Astah Community. Disponível em: <<https://www.techtudo.com.br/tudo-sobre/astah-commmunity.html>>. Acesso em: 14 jul. 2021.

TOSTES, Luís Eduardo Fernandes Rogério. Análise da implementação de sistema ERP em pequenas empresas auxiliada por software livre. São Paulo, São Carlos, 2009.

TREINAWEB. About. Disponível em <<https://www.treinaweb.com.br/blog/o-que-e-uma-ide-ambiente-de-desenvolvimento-integrado>> Acesso em 14 de julho de 2021.

TRIVIÑOS, Augusto Nivaldo Silva. Três enfoques na pesquisa em ciências sociais: o positivismo, a fenomenologia e o marxismo. In: Introdução à pesquisa em ciências sociais. São Paulo: Atlas, 1987. p. 31-79.

YOUNG, R. R. Requirements Engineering Handbook. Norwood: Artech House, 2003

**APÊNDICES**

<b>APÊNDICE A – PLANO DE TRABALHO .....</b>	<b>58</b>
<b>APÊNDICE B – PLANO DE MARKETING .....</b>	<b>61</b>
<b>APÊNDICE C – SCRIPT DO BANCO DE DADOS.....</b>	<b>68</b>
<b>APÊNDICE D – DIAGRAMA ENTIDADE RELACIONAMENTO.....</b>	<b>71</b>
<b>APÊNDICE E – APRESENTAÇÃO DO TRABALHO NO VI MEEPI.....</b>	<b>72</b>

## APÊNDICE A – PLANO DE TRABALHO

### SIGIC: SISTEMA INTEGRADO DE GESTÃO COMERCIAL

#### TURMA 2019

**Discentes** : Cristiane Motin – [crismotin2019@gmail.com](mailto:crismotin2019@gmail.com)

Eder Diego Souza – [diegosous@gmail.com](mailto:diegosous@gmail.com)

**Orientador** : Ademir Luiz do Prado – [ademir.prado@ifpr.edu.br](mailto:ademir.prado@ifpr.edu.br)

**Resumo:** Será desenvolvido um sistema para gerenciamento de pequenos comércios. Nossa plataforma será com pedidos *Delivery*, controle de pedidos em mesas, comandas e balcão. Tendo como funções, PDV, controle de estoque e financeiro, cadastro de fornecedores, funcionários e clientes. O sistema será desenvolvido utilizando HTML, JavaScript, CSS, Java. Para o Sistema Gerenciador Banco de Dados será utilizado o MySQL. No primeiro momento vamos implantar em um restaurante familiar, fazendo as alterações necessárias e melhorando. Nosso objetivo é implantar em outros comércios de pequeno porte.

#### 1. Introdução e Justificativa

Para gerenciar uma empresa é desejável ter um sistema informatizado. Quando não se tem o sistema, manualmente é realizada as somas do que se vendem. Assim tornam mais complicadas de fazer, o sistema vai armazenar em um banco de dados e não será preciso usar o armazenamento em pastas. O sistema terá o custo de implantação. Existem no mercado outros sistemas de gerenciamento, porém seus custos são diferenciados dependendo da necessidade de cada. O sistema será desenvolvido buscando ao máximo atender as necessidades do pequeno comércio.

#### 2. Objetivos

##### Objetivo Geral:

Gerenciar pequenos comércios

##### Objetivos Específicos:

- Gerenciar pedidos *Delivery*
- Gerenciar o cadastro de funcionários, clientes e fornecedores
- Gerenciar controle de pedidos em mesas
- Gerenciar comandas e balcão
- Gerenciar estoques (entrada e saída)

### 3. Metodologia

Como metodologia para desenvolvimento do sistema será utilizado a pesquisa junto ao cliente para saber quais funções serão necessárias para o desenvolvimento. Após as entrevistas será feito a documentação. Serão realizadas diversas entrevistas junto ao cliente para determinar a sua necessidade.

### 4. Cronograma Proposto

ETAPAS	PERÍODO										
	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
1	X	X									
2			X								
3				X							
4				X							
5					X	X					
6							X	X	X		
7										X	X

Etapas:

1. Definição do sistema e entrevistas com o cliente
2. Levantamento dos requisitos funcionais e não funcionais
3. Escolha da linguagem de programação e BD
4. Escolha da plataforma de hospedagem do sistema
5. Diagramas do sistema
6. Codificação e testes
7. Disponibilidade e entrega do sistema para o cliente

## 5. Forma de Acompanhamento/Orientação

Acompanhamento será realizado no horário de atendimento ao aluno com a frequência quinzenal entre aluno e orientador, via Google Meet, nas quartas-feiras das 17h00 às 19h00.

## 6. Referências Bibliográficas

DEITEL, Paul; DEITEL, Harvey. Java: como programar. 8. ed. São Paulo, SP: Pearson, 2010. 1144 p.

PRESSMAN, R. S. Engenharia de Software: uma abordagem profissional. 7. ed. Porto Alegre: McGraw-Hill, 2011.

LARMAN, C. **Utilizando UML e Padrões: Uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento**. 3. Ed. Editora Bookman, 2007. 695p.

POSTGRESQL. About. Disponível em < <https://www.postgresql.org/about/>> Acesso em 12 de julho de 2021

SILBERSCHATZ, Abraham; KORTH, Henry F; Sudarshan, S. **Sistema de Banco de Dados**. São Paulo: Makron Books, 1999.

Colombo, 14 de maio de 2021.



**Ademir Luiz do Prado**  
Orientador



**Cristiane Motin**  
Aluna



**Eder Diego de Sousa**  
Aluno

## APÊNDICE B – PLANO DE MARKETING

### 1. SUMÁRIO EXECUTIVO

De acordo com Guindani (2013), o mix de marketing é composto de produto, preço, praça e promoção chamados de 4P 's. Abaixo listamos a descrição

**PRODUTO:** O produto se refere àquilo que a empresa vende. Em outras palavras, o produto é tudo aquilo que é disponibilizado ao consumidor, pensando em se destacar entre os concorrentes e conquistar os clientes.

**PREÇO:** O preço se refere ao custo que o cliente terá ao adquirir seu produto e/ou serviço.

**PRAÇA:** A praça se refere dentro de uma estratégia de marketing refere-se ao lugar ou onde o produto será oferecido.

**PROMOÇÃO:** A promoção se refere em colocar um preço em seu produto que seja adequado ao mercado.

Assim sendo, o mix de marketing ajuda a definir os conceitos básicos para compor qualquer estratégia para alcançar um público.

O presente trabalho abordou o desenvolvimento de um sistema de pedidos com o conceito de planejamento de recurso corporativo ou Enterprise Resource Planning (ERP). Proporcionando com isso que a empresa tivesse um crescimento tecnológico e um melhor controle dos pedidos realizados. A empresa estava passando por problemas na gestão do negócio, já que todos os cálculos de venda e compra de produtos eram feitos manualmente. Por esse motivo a gestão era dificultada e propensa a erros. Então com isso pensou-se: como aperfeiçoar a gestão da empresa, agilizando os processos existentes e reduzindo erros da gestão manual. Com o desenvolvimento do sistema para a gestão, esperou-se minimizar os problemas que a gestão manual apresentava, automatizando o processo de fazer pedidos e transformando os cálculos que até então eram manuais para que a aplicação executasse automaticamente. Pretendemos fazer a divulgação do nosso sistema por meio de (redes sociais, Youtube, sites e panfletos).

**Missão:** Garantir uma experiência de satisfação para nosso cliente.

**Visão:** Desenvolver um sistema de alta qualidade e de uso fácil para aqueles que não são experts em comunicação.

**Valores:** Buscamos o comprometimento com nossos clientes, valorizamos as pessoas.

## 1.1 ANÁLISE SISTEMA DE GESTÃO INTEGRADO COMERCIAL

Figura 1 – Análise SISTEMA DE GESTÃO INTEGRADO COMERCIAL

	FORÇAS	FRAQUEZAS
FATORES INTERNOS	<p>Equipe capacitada e disponível para dar todo suporte aos usuários do sistema;</p> <p>Ofertas de treinamentos e cursos, abertos para opiniões dos clientes;</p> <p>Podemos mostrar que nosso software pode ser adaptado conforme necessidade do cliente.</p> <p>Superar as expectativas dos clientes.</p>	<p>Indisponibilidade neste momento de dedicação 100% do tempo para desenvolvimento do sistema e atendimento aos clientes.</p>
FATORES EXTERNOS	<p>Necessidade de agilidade, flexibilidade nas comunicação entre o prestador de serviço ou vendedor de produto e o seu cliente.</p> <p>Rápido avanço tecnológico que proporciona atualizações no sistema para atender os clientes.</p>	<p>Concorrentes;</p> <p>Resistência do cliente em adotar um sistema em substituições manuais que vem sendo feitas a anos ou décadas.</p>
	OPORTUNIDADES	AMEAÇAS

Fonte: Elaboração própria (2021)

Utilizamos nossas forças como uma forma de nos preparar contra as ameaças possíveis, buscamos mostrar para nosso cliente que ele terá o software da forma que seu comércio necessita. Queremos superar as expectativas. Nosso sistema poderá ser adaptado conforme sua necessidade. Assim, através de resultados vamos comprovar na prática como um software é capaz de aumentar a lucratividade do comércio.

Nesse momento nossa fraqueza tem sido a falta de tempo para dedicação ao desenvolvimento do nosso software. Assim queremos começar a dedicar maior parte para a criação e poder fazer a implantação e os ajustes finais.

As fraquezas nesse momento têm sido grandes pelo fato da nossa concorrência já ser implantada em comércios, assim nosso objetivo é correr atrás para conseguirmos fazer os ajustes finais e implantar.

Nossa maior força vem sendo que nosso sistema poderá ser adaptado conforme a necessidade do cliente, assim poderemos atender conforme sua necessidade.

## 2. APRESENTAÇÃO DO PRODUTO/SERVIÇO

Os pequenos negócios vêm se tornando cada vez mais parte fundamental da economia que move um país. É uma atividade econômica com essa relevância que não pode parar ou se limitar devido à falta de acompanhamento dos processos internos. Para isso, ferramentas tecnológicas estão dando suporte ao gestor de um negócio. Um software pode ser desenvolvido para auxiliar a gestão de uma empresa a controlar as entradas e saídas de insumos, o estoque do produto acabado, as entradas e saídas do caixa, as vendas, entre outras atividades.

O presente trabalho abordou o desenvolvimento de um sistema de pedidos com o conceito de planejamento de recurso corporativo ou Enterprise Resource Planning (ERP). Proporcionando com isso que a empresa tivesse um crescimento tecnológico e um melhor controle dos pedidos realizados. A empresa estava passando por problemas na gestão do negócio, já que todos os cálculos de venda e compra de produtos eram feitos manualmente. Por esse motivo a gestão era dificultada e propensa a erros. Então com isso pensou-se: como aperfeiçoar a gestão da empresa, agilizando os processos existentes e reduzindo erros da gestão manual. Com o desenvolvimento do sistema para a gestão, esperou-se minimizar os problemas que a gestão manual apresentava, automatizando o processo de fazer pedidos e transformando os cálculos que até então eram manuais para que a aplicação executasse automaticamente.

É para atender a esta demanda da gestão das organizações que criamos o software Sistema de gestão integrado comercial.

Este software vai transformar a gestão da sua empresa. Esta ferramenta é customizada atendendo à necessidade do negócio. Configuramos sob medida e personalizado para cada ramo, com base em análises profundas no processo de cada estabelecimento.

### 2.1 Pesquisa de mercado

A coleta de dados primários foi feita por meio de uma entrevista realizada no dia 12/05/2021, em um restaurante de pequeno porte, localizado no Município de Colombo diretamente com o proprietário da empresa. Ele possui o MEI. Conseguimos por meio da pesquisa observar as dificuldades enfrentadas no dia a dia em seu comércio.

Com os resultados obtidos conseguimos apontar grandes dificuldades que esse proprietário tem em conseguir melhorar o aumento de seus lucros.

Ele não tem controle dos gastos e lucros obtidos no dia, a empresa não exibe um relatório final para saber como anda a situação financeira. Deixando claro que muitas vezes não separa as compras físicas das jurídicas.

Portanto o nosso sistema servirá para ajudar na melhoria de seu estabelecimento e dos demais que se encaixam.

## 2. ESTUDO DO PREÇO E FORMA DE DISTRIBUIÇÃO

### 3.1 Preço e forma de distribuição praticados pela concorrência para oferta do produto e/ou a prestação do serviço

Abaixo apresentamos um resumo dos concorrentes do produto que está sendo desenvolvido e proposto no trabalho. O que pode diferenciar nosso software em relação a concorrência, é que prestaremos serviços remoto e presencial. Sendo que alguns deles prestam serviço somente on-line. Os preços variam muito, alguns cobram pela implantação já outros cobram somente o valor mensal.

Nosso objetivo é prestar suporte com visitas técnicas até o local, podendo também haver serviços remotos.

Quadro 1 – Levantamento dos principais concorrentes

Concorrente	Preço praticado	Forma de pagamento	Forma de distribuição
Funcionário para fazer o que o sistema vai oferecer	4500	mensal	Funcionário que pode trabalhar a distância (custo menor) ou presencialmente na empresa (custo das instalações e equipamentos para ele)
GrandChef	149.9	mensal	Atendimento on line.
Sischef	900,00 + 169,90	Implantação mensal	Atendimento On line Atendimento On line
Controle na mão	129.9	Mensal	Atendimento On line/presencial

Fonte: Elaboração dos autores (2021)

Com base em relação aos preços praticados, o nosso produto é acessível para as pequenas empresas em relação aos concorrentes, a grande parte presta atendimento somente remoto. Já no nosso software queremos fazer a implantação no local realizando o treinamento para que qualquer proprietário ou funcionário consiga fazer a utilização do sistema que será desenvolvido.

### 3.2 Formação do preço do produto

#### Custos do sistema SIGIC

Desenvolvemos toda parte do trabalho sozinhos, desde os estudos para programar, estudos sobre o que desenvolver, buscamos aprimorar nossos conhecimentos cada vez mais para o desenvolvimento desse software.

#### **Custo Variável**

Desenvolvedor  $20,00 * 6$  horas por dia,  $* 24$  dias por mês = R\$ 2880,00  $* 2 =$  R\$ 5760,00

Sendo que calculamos 1000 horas para estudos, entrevistas, desenvolvimento do software, implantação e ajustes finais.

Computadores  $6500 * 2 =$  R\$ 13000.00

Servidor = R\$ 180,00/mês

#### **Custo e despesa fixa**

Internet = R\$ 120,00/mês

Luz = R\$ 150,00/mês

Divulgação redes sociais = R\$ 300,00/mês

Divulgação Panfletos = R\$ 400,00/mês

Divulgação Outdoor = R\$ 500,00/mês

#### **Valor sistema**

Tendo em vista os custos que teremos, decidimos cobrar para implantação do sistema R \$3000,00 com 12 horas de treinamento incluído e uma mensalidade de R\$250,00/mês para manutenção do sistema e suporte presencial e remoto.

Temos a segunda opção que seria um pagamento vitalício, paga pelo sistema uma única vez no valor de R \$20.000,00 sem manutenção apenas com suporte online. Para alguma manutenção será cobrado valores à parte.

#### 4. PRAÇA: CANAIS DE DISTRIBUIÇÃO

O público-alvo são pessoas que lidam com comércio. Sabemos a importância de existir um controle maior em nossos negócios, assim esse sistema é para ter um controle nas entradas e saídas de insumos, o estoque do produto acabado, as entradas e saídas do caixa, as vendas entre outras atividades. Nosso público-alvo são as pequenas e médias empresas, que precisam de um controle para não ter perdas financeiras e furos no estoque. Temos que caminhar junto com a tecnologia, disponibilizando serviços com acesso para facilitar o dia a dia de cada um.

#### 5. PROMOÇÃO

Nosso sistema será anunciado no site. Apostando também na divulgação em redes sociais, utilizando impulsionamento de publicações no Instagram, Facebook e por meio de anúncios do YouTube. Podendo também ser divulgado em outdoors com uma chamada criativa e panfletos.

##### 5.1 Orçamento da promoção

Quadro 2 – Preços para promoção do produto nos meios de comunicação

Meio de comunicação	Forma de promoção	Preço	Observações
Facebook/Instagram	Impulsionamento	300,00	
Outdoor	área visual	500,00	
Panfletos	Divulgação	400,00	

Fonte: Pesquisa de campo (2021)

## REFERÊNCIAS

BOOCH, Grady; James Rumbaugh; e Ivar, Jacobson. UML: guia do usuário. 6ª Reimpressão. Rio de Janeiro: Elsevier, 2005.

DEITEL, Paul; DEITEL, Harvey. Java: como programar. 8. ed. São Paulo, SP: Pearson, 2010. 1144 p.

GEREMIAS, Felipe; PAPIOR, Luiz Henrique. Resumo de TCC “Desenvolvimento de um sistema ERP com foco nas tecnologias de software livre / código aberto”. UNISUL. 2006.

GEREMIAS, Felipe; PAPIOR, Luiz Henrique. Resumo de TCC “Desenvolvimento de um sistema ERP com foco nas tecnologias de software livre / código aberto”. UNISUL. 2006.

GONCALVES, Antônio. Introdução à plataforma Java (TM) EE 6 com o glassFish (TM) 3. 2. ed. Rio de Janeiro: Ciência Moderna, 2011. 563 p.

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

PEREIRA NETO, Álvaro. PostgreSQL técnicas avançadas: versões Open Source 7.x e 8.x: soluções para desenvolvedores e administradores de bancos de dados. 4. ed. São Paulo, SP: Érica, 2007. 284 p.

POSTGRESQL. About. Disponível em < <https://www.postgresql.org/about/>>. Acesso em 17 de maio de 2017.

TOSTES, Luís Eduardo Fernandes Rogério. Análise da implementação de sistema ERP em pequenas empresas auxiliada por software livre. São Paulo, São Carlos, 2009.

DEVMEDIA. Conhecendo o Spring Framework. Disponível em <<https://www.devmedia.com.br/guia/guia-de-spring/37806>>. Acesso em 12 de Junho de 2021

TREINAWEB. O que é Thymeleaf. Disponível em <<https://www.treinaweb.com.br/blog/o-que-e-o-thymeleaf>>. Acesso em 12 de Junho de 2021

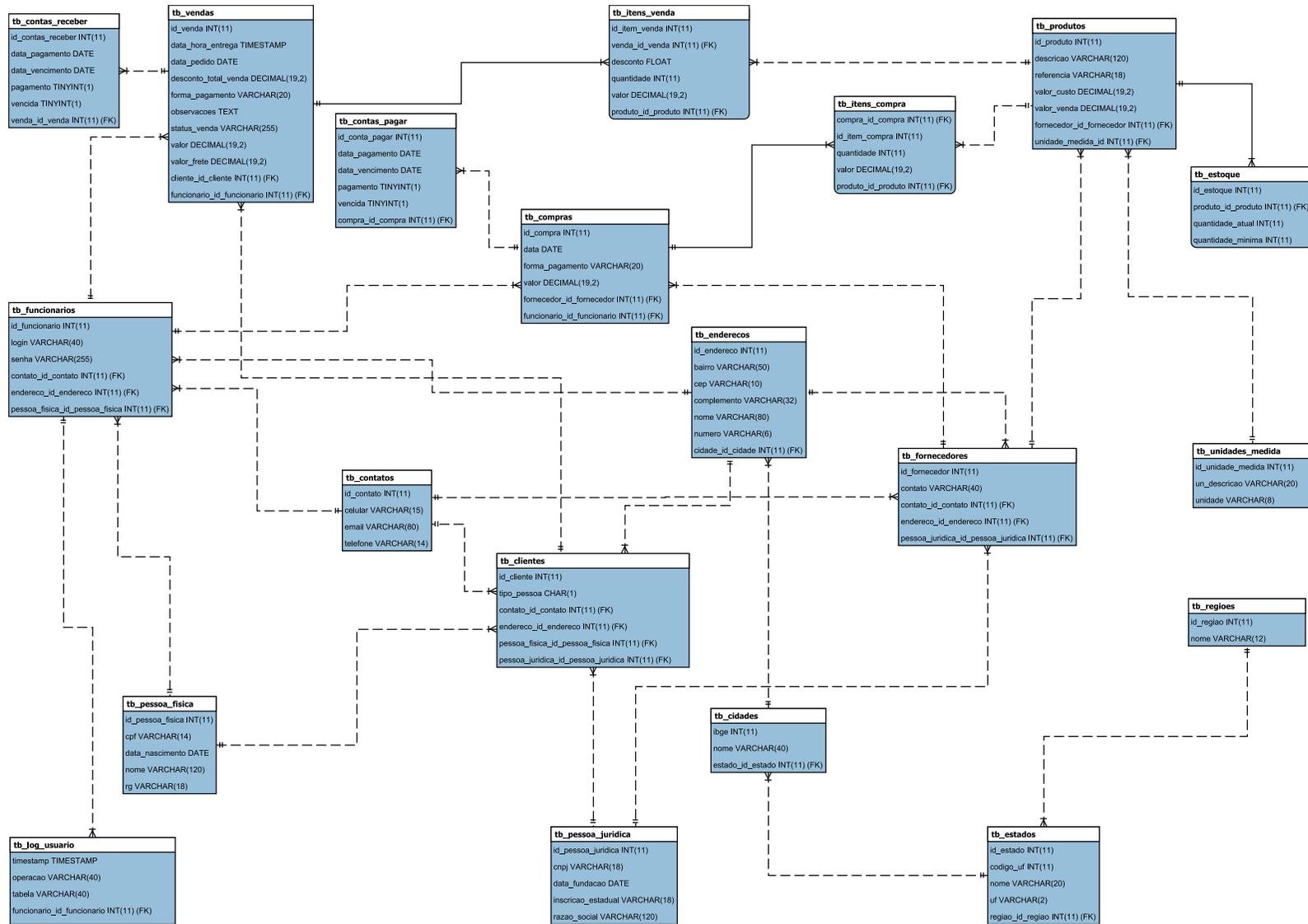
## APÊNDICE C – SCRIPT DO BANCO DE DADOS

- CREATE DATABASE SIGIC;
- create table tb\_cidades (ibge int not null, nome VARCHAR(40) not null, estado\_id\_estado int not null, primary key (ibge));
- create table tb\_clientes (id\_cliente int , tipo\_pessoa CHAR(1) not null, contato\_id\_contato int not null, endereco\_id\_endereco int not null, pessoa\_fisica\_id\_pessoa\_fisica int, pessoa\_juridica\_id\_pessoa\_juridica int, primary key (id\_cliente));
- create table tb\_compras (id\_compra int , data date not null, forma\_pagamento VARCHAR(20) not null, valor numeric(19, 2) not null, fornecedor\_id\_fornecedor int not null, funcionario\_id\_funcionario int not null, primary key (id\_compra));
- create table tb\_contas\_pagar (id\_conta\_pagar int , data\_pagamento date, data\_vencimento date not null, pagamento boolean not null, vencida boolean not null, compra\_id\_compra int not null, primary key (id\_conta\_pagar));
- create table tb\_contas\_receber (id\_contas\_receber int , data\_pagamento date, data\_vencimento date not null, pagamento boolean not null, vencida boolean not null, venda\_id\_venda int not null, primary key (id\_contas\_receber));
- create table tb\_contatos (id\_contato int , celular VARCHAR(15), email VARCHAR(80) not null, telefone VARCHAR(14), primary key (id\_contato));
- create table tb\_enderecos (id\_endereco int , bairro VARCHAR(50) not null, cep VARCHAR(10) not null, complemento VARCHAR(32), nome VARCHAR(80) not null, numero VARCHAR(6) not null, cidade\_id\_cidade int not null, primary key (id\_endereco));
- create table tb\_estados (id\_estado int , codigo\_uf int not null, nome VARCHAR(20) not null, uf VARCHAR(2) not null, regioao\_id\_regiao int not null, primary key (id\_estado));
- create table tb\_estoque (id\_estoque int not null, produto\_id\_produto int not null, quantidade\_atual int not null, quantidade\_minima int not null, primary key (id\_estoque, produto\_id\_produto));
- create table tb\_fornecedores (id\_fornecedor int , contato VARCHAR(40) not null, contato\_id\_contato int not null, endereco\_id\_endereco int not null, pessoa\_juridica\_id\_pessoa\_juridica int not null, primary key (id\_fornecedor));
- create table tb\_funcionarios (id\_funcionario int , login VARCHAR(40) not null, senha varchar(255) not null, contato\_id\_contato int not null, endereco\_id\_endereco int not null, pessoa\_fisica\_id\_pessoa\_fisica int not null, primary key (id\_funcionario));
- create table tb\_itens\_compra (compra\_id\_compra int not null, id\_item\_compra int not null, quantidade int not null, valor numeric(19, 2) not null, produto\_id\_produto int not null, primary key (compra\_id\_compra, id\_item\_compra));

- create table tb\_itens\_venda (id\_item\_venda int not null, venda\_id\_venda int not null, desconto float8, quantidade int not null, valor numeric(19, 2) not null, produto\_id\_produto int not null, primary key (id\_item\_venda, venda\_id\_venda));
- create table tb\_log\_usuario (timestamp timestamp not null, operacao VARCHAR(40) not null, tabela VARCHAR(40) not null, funcionario\_id\_funcionario int not null, primary key (timestamp));
- create table tb\_pessoa\_fisica (id\_pessoa\_fisica int , cpf VARCHAR(14) not null, data\_nascimento date not null, nome VARCHAR(120) not null, rg VARCHAR(18) not null, primary key (id\_pessoa\_fisica));
- create table tb\_pessoa\_juridica (id\_pessoa\_juridica int , cnpj VARCHAR(18) not null, data\_fundacao date not null, inscricao\_estadual VARCHAR(18) not null, razao\_social VARCHAR(120) not null, primary key (id\_pessoa\_juridica));
- create table tb\_produtos (id\_produto int , descricao VARCHAR(120) not null, referencia VARCHAR(18), valor\_custo numeric(19, 2) not null, valor\_venda numeric(19, 2) not null, fornecedor\_id\_fornecedor int not null, unidade\_medida\_id int not null, primary key (id\_produto));
- create table tb\_regioes (id\_regiao int , nome VARCHAR(12) not null, primary key (id\_regiao));
- create table tb\_unidades\_medida (id\_unidade\_medida int , un\_descricao VARCHAR(20) not null, unidade VARCHAR(8) not null, primary key (id\_unidade\_medida));
- create table tb\_vendas (id\_venda int , data\_hora\_entrega timestamp, data\_pedido timestamp not null, desconto\_total\_venda numeric(19, 2), forma\_pagamento VARCHAR(20), observacoes TEXT, status\_venda varchar(255), valor numeric(19, 2), valor\_frete numeric(19, 2), cliente\_id\_cliente int not null, funcionario\_id\_funcionario int not null, primary key (id\_venda));
- alter table if exists tb\_cidades add constraint FK4tshmkftglp2xnx3ehw2n92kg foreign key (estado\_id\_estado) references tb\_estados;
- alter table if exists tb\_clientes add constraint FKKey1v27j1qnh4cg1reutq6j1h1 foreign key (contato\_id\_contato) references tb\_contatos;
- alter table if exists tb\_clientes add constraint FKa84sw386p91cgg26vuhocr4w foreign key (endereço\_id\_endereço) references tb\_enderecos;
- alter table if exists tb\_clientes add constraint FKlXg4ch3vt120hrsrlf8e6van5 foreign key (pessoa\_fisica\_id\_pessoa\_fisica) references tb\_pessoa\_fisica;
- alter table if exists tb\_clientes add constraint FKk3hhdaghpK9nvshxyp5q0se2k foreign key (pessoa\_juridica\_id\_pessoa\_juridica) references tb\_pessoa\_juridica;
- alter table if exists tb\_compras add constraint FKao54r6lf97vo3jjag9wkbvufx foreign key (fornecedor\_id\_fornecedor) references tb\_fornecedores;
- alter table if exists tb\_compras add constraint FK77gdt68cubkv1d9y27iafb414 foreign key (funcionario\_id\_funcionario) references tb\_funcionarios;
- alter table if exists tb\_contas\_pagar add constraint FKhlslpam864vtcpix4k74q2hk foreign key (compra\_id\_compra) references tb\_compras;
- alter table if exists tb\_contas\_receber add constraint FKbn7tcepl2ts8ot63vmrr812ab foreign key (venda\_id\_venda) references tb\_vendas;
- alter table if exists tb\_enderecos add constraint FKooslw5nbhdw8dij2w15r41ymt foreign key (cidade\_id\_cidade) references tb\_cidades;

- alter table if exists tb\_estados add constraint FKtkxjkkuhqdcj41vh144tpmd1 foreign key (regiao\_id\_regiao) references tb\_regioes;
- alter table if exists tb\_estoque add constraint FKjoxa8o1j7fq7nl2d7ywmvjfq foreign key (produto\_id\_produto) references tb\_produtos;
- alter table if exists tb\_fornecedores add constraint FK63ekrw2gx8lj543shrysiabpe foreign key (contato\_id\_contato) references tb\_contatos;
- alter table if exists tb\_fornecedores add constraint FKd3el2byxdg0ssmsrjlcyoofd foreign key (endereco\_id\_endereco) references tb\_enderecos;
- alter table if exists tb\_fornecedores add constraint FK3e9tqsalnpogbjss8pdffl7n foreign key (pessoa\_juridica\_id\_pessoa\_juridica) references tb\_pessoa\_juridica;
- alter table if exists tb\_funcionarios add constraint FK9mwl4ilnpw7wl3ybi5ugsmmb6 foreign key (contato\_id\_contato) references tb\_contatos;
- alter table if exists tb\_funcionarios add constraint FKiw8a755mac13ef7dw6ce1st0n foreign key (endereco\_id\_endereco) references tb\_enderecos;
- alter table if exists tb\_funcionarios add constraint FKr6t117uv7mo3bqxgvlsmnr8s8 foreign key (pessoa\_fisica\_id\_pessoa\_fisica) references tb\_pessoa\_fisica;
- alter table if exists tb\_itens\_compra add constraint FKjriahyfa1ub8ntixpn54svjxf foreign key (compra\_id\_compra) references tb\_compras;
- alter table if exists tb\_itens\_compra add constraint FKexosj1u934tofw1r9906ibyg foreign key (produto\_id\_produto) references tb\_produtos;
- alter table if exists tb\_itens\_venda add constraint FKb3jrpcc5oisks4axrlupOvq69 foreign key (produto\_id\_produto) references tb\_produtos;
- alter table if exists tb\_itens\_venda add constraint FKexOfhm8t08as816jvh4gkgtb3 foreign key (venda\_id\_venda) references tb\_vendas;
- alter table if exists tb\_log\_usuario add constraint FKh6fe52v858sr1a6r2e2msoefm foreign key (funcionario\_id\_funcionario) references tb\_funcionarios;
- alter table if exists tb\_produtos add constraint FKmgfh6ted3h2mmfjo2g195r9g1 foreign key (fornecedor\_id\_fornecedor) references tb\_fornecedores;
- alter table if exists tb\_produtos add constraint FKsuldk8yseqimo3n3lmxdfnl5o foreign key (unidade\_medida\_id) references tb\_unidades\_medida;
- alter table if exists tb\_vendas add constraint FKj93vtlOnj9andetos5736pk3 foreign key (cliente\_id\_cliente) references tb\_clientes;
- alter table if exists tb\_vendas add constraint FKhxavbsqqwyOdbllhshhiinhxa foreign key (funcionario\_id\_funcionario) references tb\_funcionarios;

## APÊNDICE D – DIAGRAMA ENTIDADE RELACIONAMENTO



## APÊNDICE E – APRESENTAÇÃO DO TRABALHO NA VI MOSTRA DE ENSINO, EXTENSÃO, PESQUISA E INOVAÇÃO



### VI ME<sup>2</sup>PI

MOSTRA DE ENSINO, EXTENSÃO, PESQUISA E INOVAÇÃO  
Campus Colombo



#### **“ENSINO, EXTENSÃO, PESQUISA E INOVAÇÃO EM TEMPOS DE ADVERSIDADE”**

Certificamos que o trabalho intitulado, **SIGIC: SISTEMA DE GESTÃO INTEGRADO COMERCIAL**, de autoria de **Cristiane Motin, Éder Diego de Souza e Ademir Luiz do Prado**, foi apresentado na modalidade de Comunicação Oral *Online*, durante a VI Mostra de Ensino, Extensão, Pesquisa e Inovação (ME<sup>2</sup>PI) do IFPR – Campus Colombo, com o tema central *Ensino, Extensão, Pesquisa e Inovação em Tempos de Adversidade*, realizada no dia 24 de novembro de 2021.

Colombo, 24 de novembro de 2021.

**Eduard Henry Lui**  
Diretor de Ensino, Pesquisa, Extensão e Inovação

**Amanda Tavares Naves**  
Coordenadora de Ensino

PROGRAMAÇÃO da VI ME<sup>2</sup>PI

24/11/2021	
<b>MANHÃ</b>	08h30 às 09h – Abertura da IV - ME <sup>2</sup> PI do IFPR, Campus Colombo 9h às 9h30 – Programação Cultural 9h10 às 10h – Conversa com o Escritor Paulo Sandrini: a criação literária e as expectativas para a reabertura dos espaços culturais 10h às 12h - Sala 1 - Tecnologia e Informática 10h às 12h - Sala 2 - Sociedade e Cultura
<b>TARDE</b>	14h às 15h40 - Sala 3 - Sociedade e Cultura 14h às 15h40 - Sala 4 - Tecnologia em Alimentos
<b>NOITE</b>	18h10 às 18h50 – Mostra de Produções Artísticas e Culturais 19h às 20h30 – Palestra: O papel da ciência e da tecnologia em tempos de negacionismo científico e Fake News" 20h30 às 22h - Palestra: Finanças e Projetos para outra Economia 20h40 às 22h - Sala 5 - Tecnologia e Informática

 <p><b>INSTITUTO FEDERAL DO PARANÁ</b> CAMPUS COLOMBO Direção de Ensino, Pesquisa, Extensão e Inovação</p> <p><b>REGISTRO DO CERTIFICADO</b></p> <p>Registrado no COPE/COLOMBO sob nº: 002/2021</p> <p>Nº de Ordem: 1424 / IFPR Campus Colombo</p> <p>Colombo, 24 de novembro de 2021.</p> <p></p>
---