GOVERNO DO ESTADO DO RIO DE JANEIROSECRETARIA DE ESTADO DE CIÊNCIA E TECNOLOGIAFUNDAÇÃO DE APOIO À ESCOLA TÉCNICAFACULDADE DE EDUCAÇÃO TECNOLÓGICA DO ESTADO DO RIO DEJANEIRO FAETERJ/PARACAMBI

CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE INFORMAÇÃOTRABALHO DE CONCLUSÃO DE CURSO

**RAFAEL SILVA PATRICIO**

**DESENVOLVIMENTO DO SOFTWARE VENDOR: UM SISTEMA COMPLETO DE VENDAS**

**PARACAMBI, RJ  
2019**

**RAFAEL SILVA PATRICIO**

**DESENVOLVIMENTO DE   
SOFTWARE VENDOR:   
UM SISTEMA COMPLETO   
DE VENDAS**

Trabalho de Conclusão de Curso (TCC)  
apresentado ao Curso Superior de Tecnologia em Sistemas de Informação da Faculdade de Educação  
Tecnológica do Estado do Rio de Janeiro, Campus  
Paracambi – FAETERJ/Paracambi, como requisito  
parcial para a obtenção do grau de Tecnólogo em  
Sistemas de Informação.

**Orientador**: Prof. Artur Sérgio Lopes

**PARACAMBI, RJ  
2019**

|  |
| --- |
|  |

**1. INTRODUÇÃO**

1.1. OBJETIVOS

1.1.1 GERAL

1.1.2 ESPECÍFICO

1.1.3 JUSTIFICATIVA

**2. METODOLOGIA**

2.1 FERRAMENTAS E TECNOLOGIAS UTILIZADAS

**3. RESULTADOS E DISCUSSÃO**

3.1 ANÁLISE DE REQUISITOS DO SISTEMA

3.2 IDENTIDADE VISUAL

3.3 DIAGRAMAS x

3.3.1 CASO DE USO

3.3.2 CLASSE

3.3.3 ENTIDADE-RELACIONAMENTO

**4. INTERFACES GRÁFICAS**

4.1 ACESSO

4.2 TELA INICIAL (DASHBOARD)

4.3 PERFIL DO USUÁRIO

4.4 AÇÕES DO ADMINISTRADOR

4.5 AÇÕES DO PROFESSOR

**6. CONCLUSÃO**

**7. REFERÊNCIAS BIBLIOGRÁFICAS**

**RESUMO**

Este trabalho apresenta o desenvolvimento de um software para gerenciamento de estoque e realização de vendas em um modelo de negócios entre empresas.   
A partir da observação de vendedores visitando clientes, especificamente aqueles que oferecem seus produtos para empresas, percebendo a manualidade do seu processo em muitos dos casos e aliados à vontade do desenvolvedor em criar aplicações no ramo corporativo, decidiu-se pela criação de uma solução robusta, que atendesse aos praticantes desse nicho de negócios.   
O software possibilitará a criação e controle de estoque, execução de vendas, envio do resumo da venda por e-mail, geração de relatórios, resumo financeiro com gráficos e separação de responsabilidades através de níveis de acesso de usuários.  
Dessa forma então, assistindo tanto os vendedores durante o processo de vendas, à empresa nos processos gerenciais e consequentemente os clientes, com mais agilidade e informação.

**ABSTRACT**

This work presents the software development for inventory management and sales in an intercompany business model.  
From the observation of vendors visiting customers, specifically those who offer their products to companies, realizing the manuality of their process in many cases and allied to the willingness of the developer to create corporate applications, it was decided to create a robust solution, that would attend practitioners of this business niche.  
The software will allow anyone to create and control inventory, execute sales, send sales summary by e-mail, generate reports, financial summary with charts and separation of responsibilities through user access levels.  
That way, by assisting both the sales people during the sales process, the company in the management processes and consequently the customers, with more agility and information.

**1. INTRODUÇÃO**  
 Ao longo de toda evolução da informática, ela sempre propôs a criação de novos produtos, serviços e ideias, mas mais do que isso, ela sempre se dispôs a aprimorar o que já existia.  
Renovar o que já funciona, agora com os recursos da tecnologia, é pensamento fundamental para qualquer estudante ou profissional da área.

Segundo Richard Stallman,

“Pode-se definir tecnologia da Informação (TI) como o conjunto de todas as [atividades](https://pt.wikipedia.org/wiki/Atividade_(engenharia_de_software)) e soluções providas por recursos de computação que visam a produção, o armazenamento, a transmissão, o acesso, a [segurança](https://pt.wikipedia.org/wiki/Segurança_da_informação) e o uso das [informações](https://pt.wikipedia.org/wiki/Escrita). Na verdade, as aplicações para TI são tantas — e estão ligadas a tantas áreas — que há diversas definições para a expressão e nenhuma delas consegue determiná-la por completo. É a área da [informática](https://pt.wikipedia.org/wiki/Informática) que trata a informação, a [organização](https://pt.wikipedia.org/wiki/Organização) e a classificação de forma a permitir a tomada de decisão em prol de algum objetivo.”[1]

Desta forma, podemos acrescentar que TI é utilizada como um meio em qualquer área conhecida. É um instrumento que possibilita a revitalização de antigas práticas, a melhor compreensão do objeto aplicado, a criação de novos indicadores e de tendências potencializadoras, em qualquer que seja a área de aplicação.

Por isso que, observa-se a tecnologia tão impregnada em tudo.  
Ela é parte integrante dos serviços, produtos e informações que são geradas.

Um dos segmentos que mais usufrui da tecnologia como um todo são as empresas.  
O mundo corporativo, sempre buscando melhorar seus processos, serviços, produtos e alcançar destaque no cenário de competitividade, não costuma poupar esforços quando trata-se de investimento em sistemas informatizados.

Uma pesquisa anual do PortalERP realizada em janeiro de 2017 junto a mais de 4000 empresas constatou que, pelo menos 44% delas, pretendiam fazer investimentos na melhoria de seus processos e sistemas.

Segundo a 28ª edição da Pesquisa do Uso de Tecnologia nas Empresas, da FGV, o aumento do investimento de 1% em tecnologia, tem aumentado os lucros das companhias em 7% dentro de dois anos.

É fácil para as empresas perceberem a importância de sistemas informatizados, uma vez que é fácil apurar sua eficácia. O aprimoramento dos processos impulsiona o negócio e resulta em lucro, facilmente aferido em qualquer relatório de receitas.

O software em questão debruça-se sobre um nicho específico de vendas, conhecido como Business to Business, B2B, ou Negócio para Negócio, que nada mais é do que a venda para outras empresas, ao invés de direto ao cliente final.

Alguns exemplos seriam indústrias automobilísticas que vendem os carros para agências de automóveis (empresa), e essas sim vendem ao consumidor final, ou a empresa de sorvetes Kibon, que não possui lojas físicas, mas vendedores que se responsabilizam em distribuir os produtos para lanchonetes, supermercados, padarias e outros, não tendo contato direto com quem irá consumi-lo de fato.

A aplicação toma como exemplo um distribuidor de produtos eletrônicos, que possui equipe de vendedores responsáveis por fechar vendas com lojas especializadas.

**1.1. OBJETIVOS**

**1.1.1 Objetivo geral**

Desenvolver um sistema web, capaz de atender às demandas de uma empresa no modelo B2B, no que se refere à parte de vendas e produtos.  
A aplicação deve oferecer controle sobre o estoque de produtos, garantindo a fidedignidade dos dados e com isso, permitir operações de vendas pelos vendedores e ações de consultas de informações pertinentes aos usuários.

**1.1.2 Objetivos Específicos**

Criar uma interface amigável, de modo que qualquer usuário, independentemente de suas experiências prévias com sistemas informatizados, tenha facilidade e sinta-se à vontade para usufruir da aplicação;

permitir a definição de estratégias de venda através da análise de indicadores;

conseguir qualificar melhor o trabalho dos vendedores.

**1.1.3 Justificativa**

A ideia surgiu após a observação de negócios, no estilo descrito, ocorrendo sem a utilização de nenhuma tecnologia.   
Após pesquisa, encontrou-se dificuldades em identificar softwares que cumprissem tais quesitos. Os sistemas de vendas existentes são muito voltados para o tipo de venda mais comum, vendedor-cliente final, deixando carentes os que negociam com outras empresas.

Além disso, há um desejo forte e longínquo do autor em desenvolver soluções comerciais, sendo seu maior foco ao longo dos estudos e criação de projetos pessoais.

Portanto, percebeu-se a oportunidade em aliar o desenvolvimento do trabalho de conclusão de curso, ausência na variedade de soluções disponíveis e a possibilidade de enriquecimento do portfólio do desenvolvedor, buscando alocá-lo no mercado de trabalho após a graduação.

**2. METODOLOGIA**

**2.1 Tecnologias e Ferramentas utilizadas**

**2.1.1 UML (Unified Modeling Language)**

A UML, ou Linguagem Unificada de Modelagem,   
é uma linguagem gráfica para visualização ,   
especificação, construção e documentação de   
artefatos de sistemas complexos de software.

**2.1.2 StarUML**

O StarUML é um software que possibilita a criação de modelagens e diagramas seguindo a padronização da UML 2.0, a última padronização da linguagem.

Todos os diagramas de software apresentados neste trabalho foram criados nesta aplicação.

**2.1.1 Java**

Java é uma linguagem de programação orientada a objetos criada no início dos anos 90. Desde então é extremamente usada e difundida, sendo precursora de preceitos que seriam e são replicados em diversas outras linguagens.  
É hoje uma das linguagens mais utilizadas no mundo, sendo base para a criação de aplicativos para aparelhos móveis com sistema Android.

Segundo Schildt[2]:

“Java é a principal linguagem da Internet,   
mas é mais do que isso. Ela revolucionou a programação,   
mudando a maneira de pensarmos tanto sobre a forma   
quanto sobre a função de um programa.   
Atualmente, ser um programador profissional exige   
a habilidade de programar em Java, tal é sua importância.”

A linguagem foi escolhida por ser bastante robusta e com vasta documentação e suporte da comunidade.

**2.1.2 Spring Framework**

O Spring Framework é uma coleção de soluções que ajudam na criação de aplicações Java, principalmente web.

Dentro dela, estão sendo utilizadas neste projeto os seguintes módulos:

Spring MVC, que ajuda na construção de aplicações web dentro do padrão Model, View e Controller.

Spring Data, que auxilia na parte de persistência com o banco de dados, principalmente nas operações de CRUD da aplicação.

Spring Security, que provê facilidades para a parte de segurança do sistema.

O Spring Framework foi escolhido pois reúne o que existe de melhor, no que toca as soluções para Java na arquitetura web.   
Tendo bastante documentação e uma comunidade muito ativa, oferece soluções já sólidas no mercado.

**2.1.4 MySQL**

É um Sistema de Gerenciamento de Banco de Dados (SGBD) dos mais populares no mundo. Utiliza a linguagem SQL e é extremamente difundida no mundo do desenvolvimento.

Segundo a empresa Oracle:

“O MySQL é o banco de dados de código aberto mais conhecido no mundo. Com comprovado desempenho, confiabilidade e facilidade de uso, tornou-se a principal opção de banco de dados para aplicativos baseados na Web, usado por propriedades de alto perfil, incluindo Facebook, Twitter, YouTube.”.

Foi escolhido pois apresenta uma solução simples, consolidada, com boa documentação de fácil implementação.

**2.1.5 Wildfly**

O Wildfly foi utilizado para a incrementação gradual do banco de dados e controle versionado dessas evoluções.

**2.1.6 Hibernate**

O Hibernate é o framework baseado no conceito ORM (Modelo Entidade-Relacionamento) mais utilizado atualmente. Ele abstrai as operações com banco de dados e disponibiliza uma enorme gama de propriedades e opções para controle do fluxo de informação da aplicação.

**2.1.3 HTML 5**

HTML é a sigla em inglês para HyperText Markup Language, que em português significa linguagem para marcação de hipertexto [3] .

Nada mais é que uma linguagem de marcação que estrutura documentos seguindo os Padrões Web.

O número 5 refere-se à última versão adotada pela internet.

**2.1.7 Javascript**

O Javascript faz parte da tríade de desenvolvimento web, além do HTML e do CSS.  
É ele que define o comportamento da página, no sentido de interatividade.

Apesar de a linguagem ter crescido bastante e ter se tornado bem mais do que só definição de como a página irá se comportar, é essa funcionalidade que será incorporada neste projeto.

**2.1.8 CSS**

O CSS (Cascading Style Sheets) é utilizado para definir o estilo e posicionamento dos elementos dentro das páginas HTML. As fontes, cores, margem, altura, largura e inúmeros outros aspectos são controlados através de códigos escritos com esta tecnologia.  
  
**2.1.9 Bootstrap**

O Bootstrap é uma biblioteca para ajudar na criação de interfaces.

O framework, criado pelos desenvolvedores do Twitter, contém uma vasta quantidade de elementos e componentes estilizados e já adequados para adaptação em vários tamanhos de telas.

Foi escolhido pois retira um pouco da responsabilidade da estilização e comportamento das páginas, sobrando mais tempo para que o desenvolvedor foque na solução do problema, e também fornece componentes já prontos para serem bem renderizados em diferentes formatos de telas, a chamada responsividade.  
  
**2.1.10 Thymeleaf**

O Thymeleaf é uma template engine que ajuda a dinamizar nossas páginas HTML, tornando possível que elas se comuniquem com o código back-end, nesse caso, o Java.

Foi escolhido pois aparentou ser uma solução melhor do que o “concorrente” JSP para a criação deste tipo de aplicação monolítica. E ao longo do projeto realmente mostrou-se mais dinâmica e intuitiva.

**2.1.11 Eclipse IDE**Como Ambiente de Desenvolvimento Integrado (IDE), foi utilizado o Eclipse em sua versão denominada de Neon.2.   
O Software é grátis e de código-aberto.

**2.1.12 Servidor container**

Como servidor web, foi escolhido o Tomcat, mais precisamente sua versão 8.  
Além de simples e leve, é ideal para este tipo de aplicação mais simples.  
É mantido pela Apache e tem seu código livre e aberto, sendo gratuita a sua utilização.

**2.1.13 Git**

O GIT é uma linguagem de versionamento de código gratuita e open-source.  
Ela facilita a organização e o rastreamento do projeto.

**2.1.14 GitHub**

O GitHub é uma plataforma gratuita que permite a hospedagem de código-fonte de softwares através da linguagem de controle de versionamento Git.

Todas as tecnologias citadas foram escolhidas após análise dos diagramas e da projeção da aplicação.   
Com a necessidade da utilização do software, tanto no computador quanto em dispositivos móveis, foi definido então que uma aplicação web seria o ideal.

**3. RESULTADOS E DISCUSSÃO**

**3.1 Análise de Requisitos do Sistema**

|  |
| --- |
| 1. O sistema deverá permitir que o administrador cadastre e remova vendedores. |
| 2. O administrador poderá atualizar seus dados cadastrais e dos vendedores. |
| 4. O vendedor poderá atualizar seus dados cadastrais. |
| 5. O sistema deverá permitir que o vendedor cadastre produtos, marcas e subcategorias. |
| 6. O sistema deverá possibilitar que um administrador retire relatórios. |
| 7. O vendedor poderá, através de vários filtros, pesquisar pelos produtos no banco de dados. |
| 8. O sistema deverá permitir que o vendedor realize uma venda. |
| 9. O sistema deverá permitir que o vendedor envie a lista de compras para o e-mail do cliente. |
| 10. O vendedor deverá ter acesso ao histórico de vendas dele. |
| 10. O administrador poderá cancelar uma venda. |
| 11. O software deverá funcionar em qualquer browser. |

Auxiliar o vendedor no processo de venda, agilizando e tendo maior controle sobre a tarefa;

controle de estoque, com cadastro e consulta de itens simples e eficiente;

consultar informações sobre vendas e receitas, resumidamente através de gráficos e detalhadamente através de relatórios.

possibilitar a alteração de dados cadastrais, incluindo senha, pelo próprio vendedor;

possibilitar que um usuário de nível administrador inclua ou remova os vendedores do acesso ao sistema.

**3.1.3 Identidade Visual**

Apesar do software ser aplicável a qualquer ramo de negócios, dentro da proposta B2B, a aplicação foi pensada para um cliente fictício especialista em produtos eletrônicos. Portanto, foi desenvolvida, em conjunto com a solução sob demanda, uma identidade visual para a empresa.

Definiu-se o cliente como Mago Digital.  
Além de aproximar a experiência de desenvolvimento deste projeto o mais próximo possível de uma criação de solução no mundo real, as imagens e marca são utilizadas no software para representar a empresa e ligá-lo ainda mais à companhia.

  
Figura 1: Identidade visual da empresa fictícia Mago Virtual

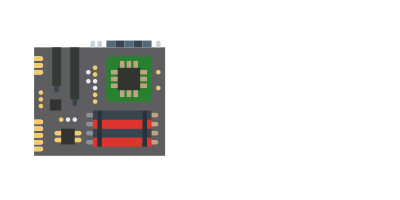
  
Figura 2: Elemento da imagem separado

  
Figura 3: Elemento textual separado