



**UNIVERSIDADE FEDERAL DO ACRE**

**CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**MAGO SYSTEM – SISTEMA DE GERENCIAMENTO DE ESTOQUE DA LOJA  
MAGO SOM & ACESSÓRIOS**

**RIO BRANCO**

**2018**

**LUCAS CAMPELO VIEIRA**

**MAGO SYSTEM – SISTEMA DE GERENCIAMENTO DE ESTOQUE DA LOJA  
MAGO SOM & ACESSÓRIOS**

Relatório de estágio apresentado como exigência parcial para obtenção do grau de bacharel em Sistemas de Informação da Universidade Federal do Acre.

Prof. Orientador: Olacir Rodrigues Castro Junior, Dr.

**RIO BRANCO**

**2018**

## **TERMO DE APROVAÇÃO**

**LUCAS CAMPELO VIEIRA**

### **MAGO SYSTEM – SISTEMA DE GERENCIAMENTO DE ESTOQUE DA LOJA MAGO SOM & ACESSÓRIOS**

Este relatório de estágio supervisionado foi apresentado como requisito parcial do Curso de Bacharelado em Sistemas de Informação da Universidade Federal do Acre, sendo aprovado pela banca constituída pelo professor orientador e membros abaixo mencionados.

Compuseram a banca:

---

Prof. Olacir Rodrigues Castro Junior, Dr  
Curso de Bacharelado em Sistemas de Informação

---

Prof. Jean Gonzaga Souza de Oliveira, Me.  
Curso de Bacharelado em Sistemas de Informação

Rio Branco, 23 de março de 2018

*“O sucesso é um professor perverso.  
Ele seduz as pessoas inteligentes e as faz  
pensar que jamais vão cair.”(Bill Gates).*

## **AGRADECIMENTOS**

Agradeço primeiramente a minha família que sempre me apoia em todas as decisões e dificuldades apresentadas, especialmente minha mãe Andréa Campelo que me auxilia em todas as horas.

Agradeço a minha namorada Daniela Barivieri que me ajudou desde a elaboração do projeto até a finalização do mesmo.

Agradeço ao meu orientador Olacir Rodrigues que aceitou me auxiliar durante todo esse projeto, sua participação foi de grande importância para obter os resultados esperados.

Agradeço aos amigos que fiz no curso de Sistemas e que pude compartilhar conhecimentos e experiências com os mesmos.

## RESUMO

Dentro de uma empresa, gerenciar o estoque pode ser uma tarefa bastante complicada. O estoque é um dos itens mais importantes para a gestão de uma empresa, pois é necessário ter um absoluto controle sobre ele, mantendo suas informações atualizadas para não superlotar o almoxarifado ou deixar faltar itens por má gestão. Para impedir que erros aconteçam, uma alternativa viável é adquirir um sistema de gerenciamento de estoque que se adeque as necessidades da empresa. A empresa Mago Som & Acessórios especializada em peças automotivas com foco na instalação de som automotivo necessita de um sistema que tenha fácil usabilidade e que apenas pessoas autorizadas possam acessá-lo e que o sistema possibilite a emissão de relatórios de acordo com as necessidades da loja, facilitando assim o controle do estoque e das transações que ocorrem na empresa. Para alcançar o objetivo destacado foram aplicadas técnicas de engenharia de software para possibilitar o desenvolvimento de um sistema na linguagem Java. Deste modo o controle do estoque deixará de ser manual através de planilhas e passará a ser um controle automatizado com menos intervenção humana e mais segurança dos dados.

**Palavras-chave:** Engenharia de software. Java. Estoque.

## **ABSTRACT**

In a company, managing the inventory can be a rather complicated task. Inventory is one of the most important items for a company's management, it is a absolute control is a requirement, keeping your information up-to-date to avoid overloading the warehouse or missing itens dueto control reasons. To prevent errors from occurring, a viable alternative is to acquire a stock management system that fits the company's needs. The company Mago Som & Acessórios is specialized in training tools in automotive sound installation needs a system that is easy to use and that only allows authorised personnel to use and has more over , the system must allow the generation of reports according to the store needs, thus facilitating the control of the inventory and the transactions that occur in the company. In order to reach the highlighted objective, it is necessary to apply software engineering techniques to enable the development of the system in the Java language. In this way the inventory control will no longer be manual through spreadsheets and will become an automated task with less human intervention and more data security.

**Keywords:** Software Engineering. Java. Stock.

## LISTA DE FIGURAS

Figura 1: Exemplo de casos de uso .....	17
Figura 2: Exemplo de diagrama de atividade .....	17
Figura 3: Exemplo de diagrama de classes.....	18
Figura 4: Exemplo de protótipo de baixo nível .....	19
Figura 5: Empresas que usam Java.....	20
Figura 6: Empresas que dependem do MySQL.....	21
Figura 7: Logo da Mago Som & Acessórios .....	22
Figura 8: Fachada da Mago Som & Acessórios .....	22
Figura 9: Som instalado pela Mago Som & Acessórios .....	23
Figura 10: Som instalado pela Mago Som & Acessórios .....	23
Figura 11: Diagrama de caso de uso .....	26
Figura 12: Diagrama EER .....	27
Figura 13: Protótipo da tela principal .....	28
Figura 14: Trecho do código do Cálculo do valor de venda do produto .....	29
Figura 15: Tela de cadastro de produto.....	29
Figura 16: Relatório de todos os produtos.....	30
Figura 17: Adicionar item à venda.....	31
Figura 18: Tela de venda com itens adicionados .....	32
Figura 19: Comprovante da venda realizada.....	33



## **LISTA DE QUADROS**

Quadro 1: Requisitos funcionais do sistema .....	25
--	----

## SUMÁRIO

<b>1.</b>	<b>INTRODUÇÃO.....</b>	<b>11</b>
1.1	PROBLEMA DA PESQUISA .....	11
1.2	OBJETIVOS DA PESQUISA .....	12
1.2.1	OBJETIVO GERAL.....	12
1.2.2	OBJETIVOS ESPECÍFICOS.....	12
1.3	JUSTIFICATIVA DA PESQUISA .....	12
1.4	METODOLOGIA .....	12
1.5	ORGANIZAÇÃO DO TRABALHO .....	14
<b>2.</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>15</b>
2.1	ENGENHARIA DE SOFTWARE .....	15
2.1.1	LEVANTAMENTO DE REQUISITOS.....	15
2.1.2	CASOS DE USO.....	16
2.1.3	DIAGRAMA DE ATIVIDADE .....	17
2.1.4	DIAGRAMA DE CLASSES.....	18
2.1.5	PROTOTIPAÇÃO .....	18
2.2	PROGRAMAÇÃO .....	19
2.2.1	JAVA .....	19
2.2.2	MYSQL .....	20
<b>3.</b>	<b>ESTÁGIO SUPERVISIONADO.....</b>	<b>22</b>
3.1	EMPRESA .....	22
3.2	ESTRUTURA ORGANIZACIONAL.....	23
3.3	ATIVIDADES DESENVOLVIDAS .....	23
3.3.1	IDENTIFICAÇÃO DO PROBLEMA.....	24
3.3.2	LEVANTAMENTO DOS REQUISITOS.....	24
3.3.3	MODELAGEM DO SISTEMA .....	25
3.3.4	PROTOTIPAÇÃO .....	27
3.3.5	DESENVOLVIMENTO DO SISTEMA .....	28
3.4	RESULTADOS OBTIDOS .....	33
<b>4.</b>	<b>CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES .....</b>	<b>35</b>
4.1	CONSIDERAÇÕES FINAIS .....	35
4.2	RECOMENDAÇÕES.....	36
	REFERÊNCIAS BIBLIOGRÁFICAS .....	37

<b>APÊNDICE .....</b>	<b>38</b>
<b>APÊNDICE A – DOCUMENTO DE REQUISITOS .....</b>	<b>39</b>
<b>APÊNDICE B – PROTÓTIPOS DO SISTEMA.....</b>	<b>43</b>

## 1. INTRODUÇÃO

Fazer um bom controle de estoque vai muito além de saber o que comprar, quais mercadorias estão estocadas e organizar a saída delas. Esse controle eficaz é de extrema importância, uma vez que todo o estoque representa dinheiro da empresa que foi empregado e que pode resultar em grandes prejuízos, caso ocorra um erro no controle, afetando consideravelmente a gestão financeira do seu negócio.

Nesse aspecto, um sistema de controle de estoque pode ajudar a realizar essa tarefa mais facilmente, pois reduz o risco de erros, permite maior controle e fornece informações importantes. (HIPER, 2018)

Nesse contexto, este trabalho tem como objetivo desenvolver um sistema para gerenciar o estoque da loja Mago Som & Acessórios de acordo com as necessidades específicas do cliente. Para isto, pretende-se utilizar algumas tecnologias que auxiliam o desenvolvimento do software, tais como: NetBeans IDE e SGBD MySQL.

### 1.1 PROBLEMA DA PESQUISA

Segundo Hiper (2018) “A adoção de um sistema de controle de estoque reduz a necessidade de controles manuais, o que faz com que o fluxo de informação seja contínuo, além de aumentar a produtividade de sua equipe”.

Além disso, o controle manual requer mais atenção, está mais suscetível a erros e é bem mais trabalhoso.

A automatização de estoques traz algumas vantagens como:

- Redução do tempo na contagem de mercadorias;
- Conhecimento do tempo real do que se tem no estoque;
- Maior controle de perdas e furtos;
- Aumento da velocidade no acompanhamento das metas;
- Monitoramento e controle da marcação de preço dos produtos.

Com esse controle é possível consultar de forma rápida todas as transações feitas na empresa, tais como: Entrada e saída de produtos, dados dos produtos e dos fornecedores, dados transacionais das vendas efetuadas, entre outros benefícios.

Com o exposto, como melhorar o controle do estoque da empresa de maneira rápida e segura?

## **1.2 OBJETIVOS DA PESQUISA**

A seguir, serão apresentados o objetivo geral e os objetivos específicos do trabalho.

### **1.2.1 OBJETIVO GERAL**

Desenvolver um sistema para gerenciamento de estoque com tecnologias livres para a empresa Mago Som & Acessórios.

### **1.2.2 OBJETIVOS ESPECÍFICOS**

Este trabalho tem os seguintes objetivos específicos:

- a) identificar o problema a ser resolvido;
- b) levantar os requisitos do sistema e realizar a modelagem do sistema com diagramas UML;
- c) protótipo o sistema;
- d) desenvolver o sistema utilizando a linguagem Java;
- e) testar as funcionalidades do sistema em ambiente de produção.

## **1.3 JUSTIFICATIVA DA PESQUISA**

O sistema de gerenciamento de estoque é uma necessidade para a empresa, pois todo o controle é feito de forma manual, dificultando a consulta de transações e de produtos, facilitando a perda e impossibilitando a emissão de relatórios que os gestores necessitam para uma melhor administração da empresa.

Com a implantação do sistema na empresa, o controle será feito de maneira simples, contendo todos os dados transacionais que relativos aos estoques da empresa, possibilitando a emissão de vários tipos de relatórios que os gestores necessitam para administrar a empresa.

## **1.4 METODOLOGIA**

A metodologia proposta consistiu em executar algumas atividades na seguinte ordem:

1. A primeira etapa diz respeito a identificação dos requisitos e elaboração do documento de requisitos de software. Os principais instrumentos para identificação dos requisitos foram a entrevista com o cliente e posteriormente

foi feita uma busca de softwares semelhantes afim de identificar requisitos obrigatórios nesse tipo de sistema e que pudessem facilitar mais ainda o uso por parte dos gestores.

2. Na segunda etapa foi feita a modelagem do banco de dados, onde foi necessário estudar diagramas de entidade e relacionamento em serviços online, afim de identificar o comportamento do banco de dados nos sistemas de gerenciamento de estoque, buscando desenvolver um sistema sólido de fácil uso e que armazene os dados no banco de dados possibilitando uma fácil busca.
3. Na terceira etapa foram feitos protótipos de baixa fidelidade afim de simular o armazenamento do banco de dados, observando se os requisitos estavam claros ou se havia a falta de algum deles para que não precisasse retornar a essa fase futuramente.
4. Na quarta fase foi desenvolvida a primeira versão do sistema, pois a empresa tinha bastante pressa para usar o sistema uma vez que não estava conseguindo mais controlar manualmente seus estoques, então foi desenvolvido uma versão que controlava o estoque de forma manual sem a opção de realizar compras e vendas de produtos, era necessário atribuir a quantidade dos produtos manualmente.
5. Na quinta fase foram realizados testes com o usuário, onde foram detectados algumas falhas que foram resolvidas imediatamente e posteriormente o cliente requisitou algumas outras funcionalidades que julgou importantes para tirar melhor proveito do sistema.
6. Na sexta fase foram desenvolvidos os demais requisitos, possibilitando realizar compras e vendas dos produtos, adicionar fornecedores aos respectivos produtos e emitir outros tipos de relatórios que o cliente solicitou conforme sentia necessidade para um melhor controle da empresa.
7. Na sétima e última fase foram realizados os testes finais do sistema já em produção, onde foram testadas as novas funcionalidades e os novos relatórios, ocorrendo tudo dentro do previsto, sendo assim, o sistema foi entregue e o projeto foi concluído.

## **1.5 ORGANIZAÇÃO DO TRABALHO**

O relatório de estágio, além desta introdução onde foram apresentados o tema, o problema, os objetivos, a justificativa, bem como a metodologia utilizada, é composto por mais 3 capítulos.

O segundo capítulo abordará a fundamentação utilizada para realização deste trabalho, mostrando os conceitos de engenharia de software utilizados na documentação do sistema criado.

O terceiro capítulo contém informações sobre a execução do estágio supervisionado, trazendo informações sobre a empresa de atuação do estágio e sua estrutura organizacional. Descreve também todas as atividades desenvolvidas e os resultados obtidos na execução deste projeto.

Por fim, no quarto capítulo são apresentadas as considerações finais e recomendações para trabalhos futuros.

O presente estágio ainda conta com apêndices contendo a documentação necessária do sistema.

## 2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados alguns conceitos de engenharia de software e de programação.

### 2.1 ENGENHARIA DE SOFTWARE

A engenharia de software é um rebento da engenharia de sistemas e hardware. Ela abrange um conjunto de três elementos fundamentais – métodos, ferramentas e procedimentos – que possibilita ao gerente o controle do processo de desenvolvimento do software e oferece ao profissional uma base para a construção de software de alta qualidade produtivamente (PRESSMAN, 1995).

Uma das primeiras definições de Engenharia de Software foi conceituada por Fritz Bauer na primeira grande conferência [NAU69],

“O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais” (PRESSMAN, 1995).

#### 2.1.1 LEVANTAMENTO DE REQUISITOS

A etapa de análise de requisitos serve para estruturar e detalhar os requisitos de forma que eles possam ser abordados na fase de elaboração para o desenvolvimento de outros elementos como casos de uso, classes e interface (WAZLAWICK, 2011).

O levantamento de requisitos é o processo de encontrar quais são as funções que o sistema deve alcançar e quais são as restrições que existem sobre essas funções. Temos duas classificações de requisito, sendo eles:

- Requisito funcional: especifica uma função que o sistema ou componente deve ser capaz de realizar.
- Requisito não funcional: descreve não o que o sistema fará, mas como ele fará.

O documento de requisitos registra todos os objetos relativos ao que o sistema necessita fazer e sob quais condições. Esse documento não precisa ser totalmente estruturado, podendo ser organizado de forma textual ou na forma de um diagrama.

O princípio de toda a atividade de desenvolvimento de software é o levantamento de requisitos. Este processo é iterativo, onde existem diversas técnicas



para o levantamento de requisitos, apresentaremos as que foram utilizadas para este trabalho conceituados por (KENDALL,2010) a seguir:

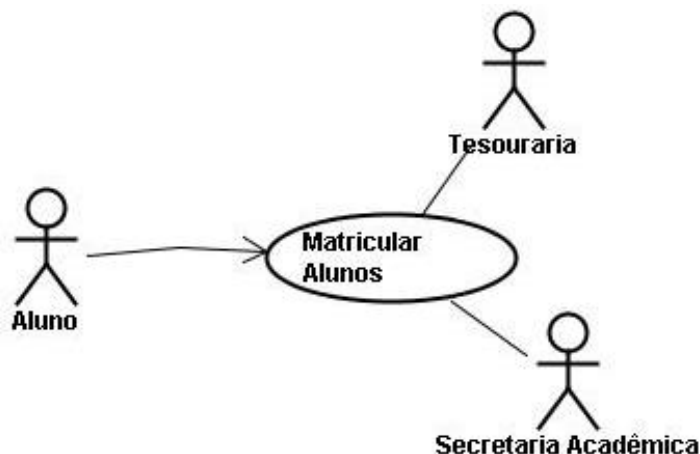
- Entrevista: é uma das técnicas tradicionais mais simples de utilizar e que produz bons resultados na fase inicial de obtenção de dados. Convém que o entrevistador dê margem ao entrevistado para expor as suas ideias.
- Questionário: o uso de questionário é indicado, por exemplo, quando há diversos grupos de usuários que podem estar em diversos locais diferentes. Neste caso, elaboram-se pesquisas específicas de acompanhamento com usuários selecionados, que a contribuição em potencial pareça mais importante, pois não seria prático entrevistar todas as pessoas em todos os locais.
- Observação: observar o comportamento e o ambiente do indivíduo que toma decisões pode ser uma forma bastante eficaz de levantar informações que, tipicamente, passam despercebidas usando outras técnicas.

### **2.1.2 CASOS DE USO**

Segundo Wazlawick (2011), os casos de uso necessitam cobrir as principais atividades de negócio ligadas ao sistema que será desenvolvido. Um caso de uso corresponde a somente um nome, possivelmente associado a um ou mais atores. A finalidade de listar os casos de uso é apanhar informações sobre como o sistema deve interagir com prováveis usuários e quais consultas e transformações da informação são indispensáveis para que processo completo de interação seja executado. É, deste modo, uma maneira de sistematizar e organizar os requisitos.

Na Figura 1 abaixo podemos observar um cenário com 3 atores que trocam mensagens exemplificando um caso de uso:

Figura 1: Exemplo de casos de uso



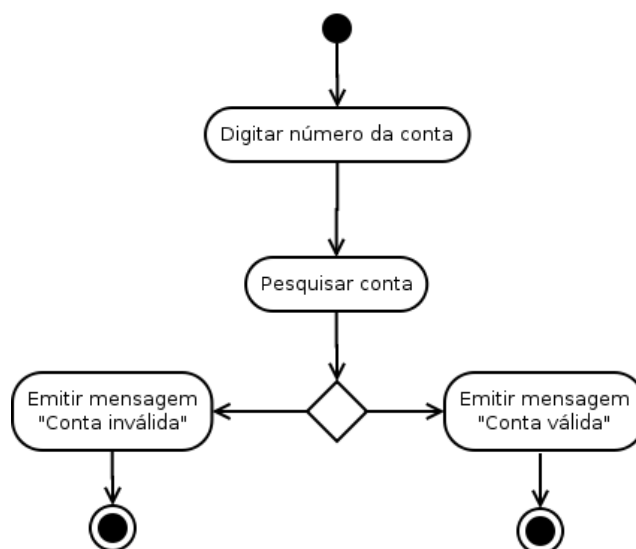
Fonte: <https://engenhariasoftware.wordpress.com/2010/12/14/aplicacao-diferenciada-da-uml-na-modelagem-de-negocio/>

### 2.1.3 DIAGRAMA DE ATIVIDADE

Os diagramas de atividades podem ser empregados para representar processos em nível organizacionais, sendo assim, uma forma mais ampla do que a mera visão de requisitos de um sistema informatizado. O diagrama pode ser dividido em raias (swimlanes), de maneira que cada raia represente um ator que participa de um conjunto de atividades. (WAZLAWICK, 2011).

Na Figura 2 abaixo podemos observar um exemplo de diagrama de atividade mostrando os passos a serem percorridos para a conclusão de um algoritmo:

Figura 2: Exemplo de diagrama de atividade



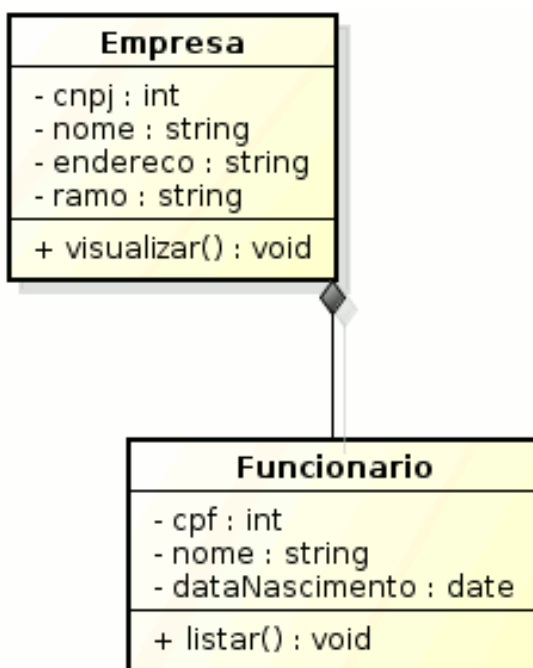
Fonte: <http://diariodainformatica.blogspot.com.br/2012/03/artigo-engenharia-de-software-diagrama.html>

### 2.1.4 DIAGRAMA DE CLASSES

Os diagramas de classe apresentam um conjunto de classes, interface e colaborações e seus relacionamentos. Graficamente, um diagrama de classes é uma coleção de vértices e arcos. Os diagramas de classes são empregados para fazer a modelagem da visão estática de um sistema. Essa visão oferece principalmente suporte para os requisitos funcionais de um sistema (BOOCH, 2005).

Na Figura 3 abaixo podemos observar um exemplo de diagrama de classes mostrando duas classes denominadas 'empresa' e 'funcionário', onde de acordo com o losango cheio na imagem o funcionário é a classe filha e empresa é a classe pai pois não faz sentido ter funcionários, se não existir uma empresa:

Figura 3: Exemplo de diagrama de classes



Fonte: <http://www.cleibsonalmeida.blog.br/site/uml-composicao-vs-agregacao/>

### 2.1.5 PROTOTIPAÇÃO

Conforme Pressman (1995), a prototipação é um processo que capacita o desenvolvedor a criar um modelo do software que será implementado. O modelo pode assumir uma das três formas:

- Um protótipo em papel ou modelo baseado em PC que retrata a interação homem-máquina de uma forma que capacita o usuário a entender quanta interação ocorrerá;

- Um protótipo de trabalho que implementa algum subconjunto da função exigida do software desejado;
- Um programa existente que executa parte ou toda a função, mas que tem outras características que serão melhoradas em um novo esforço de desenvolvimento.

A seguir podemos observar um exemplo de protótipo de baixa fidelidade:

Figura 4: Exemplo de protótipo de baixa fidelidade



Fonte: <https://sites.google.com/site/interfacecomousuario/webquest/prototipo-de-baixa-fidelidade>

## 2.2 PROGRAMAÇÃO

Nesta seção serão abordados conceitos da linguagem de programação usada para desenvolver o trabalho.

### 2.2.1 JAVA

Conforme (JAVA, 2017), Java é a base para praticamente todos os tipos de aplicações em rede e é o padrão global para o desenvolvimento e distribuição de aplicações móveis e incorporadas, jogos, conteúdo baseado na Web e softwares corporativos. Com mais de 9 milhões de desenvolvedores em todo o mundo, de forma eficiente, o Java permite que você desenvolva, implante e use aplicações e serviços estimulantes.

De laptops a datacenters, consoles de games a supercomputadores científicos, telefones celulares à Internet, o Java está em todos os lugares!

Figura 5: Empresas que usam Java



Fonte: <https://image.slidesharecdn.com/apresentao1-150630014012-lva1-app6892/95/java-por-onde-comear-codeshouse-3-638.jpg?cb=1435628520>

### 2.2.2 MYSQL

Conforme (SANTOS, 2017), O MySQL foi concebido por uma empresa chamada TcX em 1996. Foi criado porque a empresa necessitava de um banco de dados relacional estável, seguro, rápido e ainda por cima pouco custoso, pois muitos outros SGBD's (Sistemas de Gerenciamento de Banco de Dados) já possuem estas qualidades, porém com custos elevados demais. Além de tudo, o novo SGBD também precisava ser executado em hardwares relativamente baratos.

Santos (2017) ainda explica que o MySQL surgiu, então, da ideia de se criar um banco de dados que tivesse todas estas características, primordiais tanto no mercado on-line (leia-se internet e outros meios) quanto no mercado corporativo (sistemas para uso em desktop).

Segundo (SUEHRING, 2002) suas principais vantagens são:

- Desempenho: como dito acima, O MySQL é considerado o banco de dados mais rápido. Agilidade tem sido um ponto bastante relevante em matéria

de sistemas, principalmente no quesito Internet. E para que os sonhos de muitos programadores se realizem, a equipe de desenvolvimento do MySQL tem se mostrado bastante calcada no que diz respeito à performance. Isso significa que cada dia que passa o SGBD mais rápido do mercado tende a ficar mais ágil e eficaz, dando conta do recado ao qual foi designado.

- **Segurança:** outro ponto forte e decisivo em um banco de dados é a sua segurança. Com seus diversos tipos de tabelas, característica exclusiva do MySQL, é possível ter um banco de dados bastante seguro e estável contando com integridade referencial, backup e restore, controle de usuários e acessos, e ainda, se necessário, verificação e correção de corrompimento de tabelas.
- **Aplicabilidade:** muitos me perguntam se o MySQL é usado somente para aplicações internet. A resposta é não. O MySQL pode ser utilizado como banco de dados desktop, ou seja, em aplicações corporativas; e o melhor de tudo: dá suporte a diversas linguagens de programação incluindo o Delphi da empresa Borland e o Java da Sun. Ambos podem acessar o MySQL através dos drivers MySQL ODBC e MySQL JDBC, respectivamente.
- **Gratuito:** o mesmo está inscrito no tipo de licenciamento GNU General Public.

Na Figura 6 a seguir podemos observar algumas empresas que dependem do MySQL:

Figura 6: Empresas que dependem do MySQL



Fonte: <https://image.slidesharecdn.com/mysqlunaintroducciontecnica-admin-131205030839-phpapp02/95/mysql-una-introduccion-tecnica-6-638.jpg?cb=1386213067>

### 3. ESTÁGIO SUPERVISIONADO

O estágio supervisionado foi realizado na Mago Som & Acessórios durante um período de 3 meses sob a supervisão do Claudionor Alencar do Nascimento, nesse período foram desempenhadas diversas atividades: levantamento dos requisitos, prototipação de baixa fidelidade, modelagem do banco de dados, implementação do sistema na linguagem Java e testes com o cliente. A solução está pronta e já está em uso na empresa.

#### 3.1 EMPRESA

A Mago Som & Acessórios começou a funcionar em dezembro de 2003 na cidade de Rolim de Moura – RO. Em maio de 2011 a loja se transferiu para Rio Branco – AC, onde está funcionando até o momento.

Na Figura 7 a seguir podemos observar a logo da loja Mago Som & Acessórios:

Figura 7: Logo da Mago Som & Acessórios



Fonte: [www.magosom.com](http://www.magosom.com)

A seguir na Figura 8 atual fachada da loja Mago Som & Acessórios:

Figura 8: Fachada da Mago Som & Acessórios



Fonte: [www.magosom.com](http://www.magosom.com)



A loja trabalha com a venda e a instalação de acessórios automotivos, porém seu foco é na instalação de som automotivo, como podemos ver na Figura 9 abaixo:

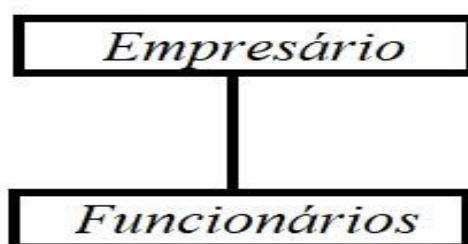
Figura 9: Som instalado pela Mago Som & Acessórios



Fonte: [www.magosom.com](http://www.magosom.com)

### 3.2 ESTRUTURA ORGANIZACIONAL

Figura 10: Estrutura organizacional da empresa Mago Som & Acessórios



Fonte: Elaboração própria

### 3.3 ATIVIDADES DESENVOLVIDAS

As atividades desenvolvidas durante o estágio foram realizadas no intuito de alcançar os objetivos estabelecidos no estágio, tais atividades foram:



- a) Identificação do problema
- b) Levantamento dos requisitos
- c) Modelagem do sistema
- d) Prototipação da interface
- e) Desenvolvimento do sistema
- f) Testes e validação

O detalhamento de cada atividade será abordado nas próximas seções.

### **3.3.1 IDENTIFICAÇÃO DO PROBLEMA**

A motivação para trabalhar o tema do estágio surgiu ao perceber que a empresa tinha dificuldade em controlar o estoque, pois todos os dados de produtos, compras e vendas da empresa eram armazenados em planilhas físicas, dificultando o controle dos mesmos. Dentre as dificuldades estão a impossibilidade de emissão de relatórios com as informações do que foi movimentado durante o dia, mês ou ano.

Desta forma, com o intuito de coletar todas as informações necessárias para a realização da modelagem do sistema foram realizadas duas reuniões com o cliente para o mesmo informar as necessidades do sistema, informando quais recursos o sistema deveria ter para atender sua demanda.

### **3.3.2 LEVANTAMENTO DOS REQUISITOS**

Após a identificação do problema, foi realizada a identificação dos requisitos e necessidades do sistema de gerenciamento de estoque, de acordo com o processo de engenharia do sistema.

Foi especificado as necessidades do cliente na forma de requisitos com o objetivo de especificar informações referentes aos requisitos funcionais e não funcionais.

Posteriormente foi necessário especificar informações relacionadas às: necessidades do cliente que precisam ser atendidas; estudo de viabilidade para analisar se possui todos os recursos necessários para o desenvolvimento e uso do sistema; limites e restrições que o sistema possui; e as ferramentas que foram necessárias para o desenvolvimento do projeto.

No entanto, no decorrer do desenvolvimento do projeto, com novas reuniões, o documento de requisitos foi sofrendo alterações para atender aos novos requisitos

solicitados pelo cliente e pela construção dos protótipos de baixa fidelidade mostrando a necessidade de mudanças no documento de requisitos.

A versão final dos requisitos funcionais obtidos é apresentada no Quadro 1 a seguir:

ID	FUNCIONALIDADE	NECESSIDADES	PRIORIDADE
RF1	CRUD Usuário	O sistema realiza 4 operações CRUD com a classe usuário: <i>Create, Read, Update e Delete</i> .	Alta
RF2	CRUD Fornecedor	O sistema realiza 4 operações CRUD com a classe fornecedor: <i>Create, Read, Update e Delete</i> .	Alta
RF3	CRUD Produto	O sistema realiza 4 operações CRUD com a classe produto: <i>Create, Read, Update e Delete</i> .	Alta
RF4	Gerenciamento de Compras	O usuário pode gerenciar as compras realizadas podendo buscar e realizar compras.	Alta
RF4	Gerenciamento de Vendas	O usuário pode gerenciar as vendas realizadas podendo buscar e realizar vendas.	Alta
RF5	Gerar relatórios	SuperUsuário poderá gerar relatórios (Todos os produtos; Produtos disponíveis; Produtos indisponíveis; Produtos por nome; Detalhamento do preço; Controle Financeiro do Estoque).	Média

Quadro 1: Requisitos funcionais do sistema

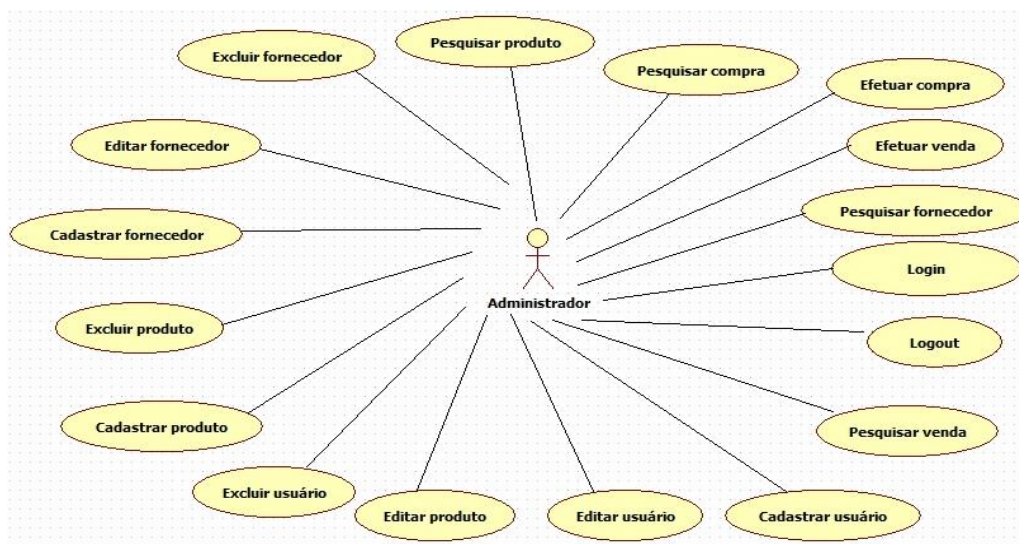
O documento de requisitos completo está localizado no APÊNDICE A deste trabalho.

### 3.3.3 MODELAGEM DO SISTEMA

Ainda durante a atividade de levantamento de requisitos, foram elaborados diagramas com o objetivo de auxiliar no desenvolvimento, validando as funcionalidades que foram estabelecidas com base nos requisitos obtidos.

Nos diagramas elaborados foi usado apenas um ator, o administrador, uma vez que o sistema será utilizado apenas por uma pessoa, possuindo apenas um responsável por gerenciá-lo. A Figura 11 mostra um diagrama com o administrador e os casos de uso com quais ele interage no sistema.

Figura 11: Diagrama de caso de uso



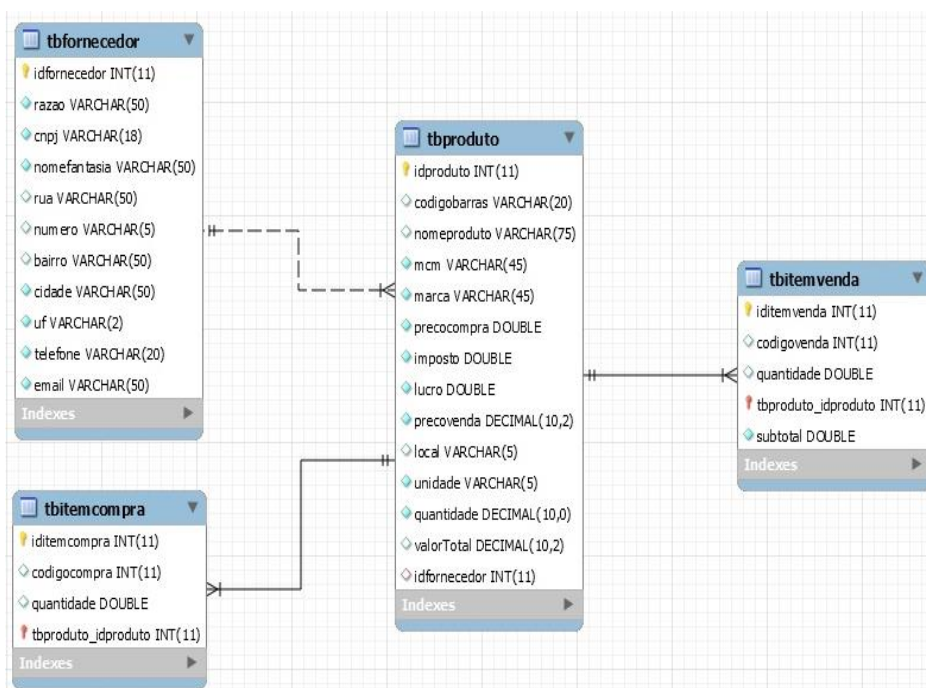
Fonte: Elaboração própria

Além do diagrama de casos de uso, foi elaborado um diagrama de entidade e relacionamento para a criação do banco de dados especificando as tabelas que são necessárias para fazer comunicação com o sistema. Na Figura 12 é possível observar o diagrama de entidade e relacionamento, onde o produto se relaciona com algumas tabelas.

A relação da tabela produto com a tabela compra(tbitemcompra) é com o objetivo de uma comprar ter 1 ou mais produtos relacionados, assim como na tabela de venda(tbitemvenda). As tabelas compras vendas possuem uma chave estrangeira que ao realizar uma compra ou venda, será inserido o código(id) do produto que esta sendo comprado ou vendido, então assim, automaticamente a quantidade do produto armazenado é alterado no banco de dados e com isso é salvo o banco de dados a transação que foi feita para uso futuro de acordo com a necessidade do gestor.

Na Figura 12 abaixo podemos observar o centro do diagrama de entidade e relacionamento:

Figura 12: Diagrama EER



Fonte: Elaboração própria

### 3.3.4 PROTOTIPAÇÃO

A prototipação foi utilizada para auxiliar tanto na descoberta quanto na validação de requisitos, pois através do protótipo foi possível descobrir elementos necessários nas tabelas do banco de dados.

Os protótipos foram desenvolvidos com o auxílio da ferramenta *Moqups*, um site que permite a elaboração de conceitos de interface do usuário e protótipos, onde possui modelos pré-formatados para a utilização, contendo menus, botões, caixas de texto, etc. A Figura 13 traz o protótipo da tela principal do sistema Mago System criado a partir dessa ferramenta.

Figura 13: Protótipo da tela principal



Fonte: Elaboração própria

Os demais protótipos de baixa fidelidade representando as outras telas do sistema estão disponíveis no APÊNDICE B.

### 3.3.5 DESENVOLVIMENTO DO SISTEMA

Primeiramente foi desenvolvida a classe responsável pelo processamento do sistema, responsável pela comunicação com o banco de dados criado após a modelagem.

Após criação da camada de controle e seguindo os protótipos criados, foi dado início ao desenvolvimento do sistema, para isso foi utilizada a IDE Netbeans. Essa atividade foi dividida em duas etapas, onde foram desenvolvidas duas versões do sistema. Foi necessária a criação de uma versão incompleta pois havia urgência no cadastro de produtos do estoque, uma vez que o cliente estava com dificuldade em gerenciar sua mercadoria, então foi proposto ao cliente que seria desenvolvida uma versão em que apenas cadastrava-se os produtos e alterava-se manualmente a quantidade de produtos do estoque, pois ainda não havia como efetuar compras e vendas de produtos.

Na Figura 14 é possível observar a solução para um dos requisitos do cliente, onde foi solicitado que o sistema deveria calcular automaticamente o valor de venda

do produto, o cliente precisa apenas informar o preço de custo, imposto pago na nota do produto e o lucro que se deseja obter na venda do respectivo produto.

Figura 14: Trecho do código do Cálculo do valor de venda do produto

```
// INICIO DO CALCULO DO VALOR DE VENDA DO PRODUTO
double qtd, vt, vc, imp, lucro, vv, valorTotal;
vc = Double.parseDouble(campoVCCad.getText());
imp = Double.parseDouble(campoImpCad.getText());
lucro = Double.parseDouble(campoLucroCad.getText());
qtd = Double.parseDouble(campoQuantCad.getText());

vt = vc + ((vc * (imp/100))); // VALOR TOTAL RECEBE O VALOR DE COMPRA VEZES O IMPOSTO DIVIDIDO POR 100
vv = vt + (vt*lucro/100);
String valor = String.valueOf(vv);
valorTotal = qtd * vv; // QUANTIDADE TOTAL DO ESTOQUE DESSE PRODUTO ARMAZENADA
String total = String.valueOf(valorTotal);

// FIM DO CALCULO DO VALOR DE VENDA DO PRODUTO
```

Fonte: Elaboração própria

A Figura 15 é a tela final de cadastro de produto, já contendo o campo para informar o fornecedor que foi solicitado pelo cliente após o término da primeira versão do sistema. A tela foi desenvolvida a partir do protótipo de interface do sistema, onde algumas informações são obrigatórias, seguindo os requisitos do cliente, caso deixe em branco alguma informação obrigatória, o sistema não efetua o cadastro do produto, informando Ao usuário que é necessário preencher todos os dados obrigatórios especificados com o asterisco na interface.

Figura 15: Tela de cadastro de produto

Fonte: Elaboração própria

Ainda na primeira versão foi solicitado pelo cliente que o sistema permitisse gerar três tipos de relatórios: Todos os produtos do estoque; produtos disponíveis no estoque; e produtos indisponíveis no estoque.

Na Figura 16 é possível observar um dos relatórios gerados usando uma base de dados de teste, representado todos os produtos cadastrados no sistema, independentemente da disponibilidade no estoque.

Figura 16: Relatório de todos os produtos

## Estoque:



Nome	Marca	Quantidade	Preço	Total
Alarme	NK	88	R\$ 225,00	R\$ 11.250,00
Amortecedor porta traseira Clio	Stabilus	23	R\$ 45,50	R\$ 1.046,50
Auto Falante	Selenium	0	R\$ 252,00	R\$ 0,00
Bateria Solar	Freedom	0	R\$ 273,00	R\$ 0,00
Camera de re estacionamento	Power XL	10	R\$ 27,30	R\$ 273,00
Camera de segurança veicular	Auto Mais	15	R\$ 67,20	R\$ 1.008,00
Carburador Gol Parati Voyage Saveiro G1	Mecar	4	R\$ 638,40	R\$ 2.553,60
Conjunto coroa e pinhao 43x9 S10	Maxgear	7	R\$ 1.344,00	R\$ 9.408,00
Cristalizador de Vidros	Car Clio	0	R\$ 25,20	R\$ 0,00
Disco de freio dianteiro	Fremax	18	R\$ 163,80	R\$ 2.948,40

Fonte: Elaboração própria

Após entregar a versão 1.0 do sistema apresentada anteriormente, foi iniciado o desenvolvimento da versão 2.0, onde foi necessário complementar o banco de dados, inserindo novas tabelas relacionadas a compras e vendas de produtos.

A solução encontrada para realizar vendas no sistema foi criar um método para adicionar um produto dentro da venda que está sendo efetuada, este item possui o código da venda, a quantidade que está sendo vendida, e o subtotal da venda, ou seja, o valor do produto multiplicado pela quantidade que está sendo vendido.

Na Figura 17 é possível ver o código responsável por adicionar um item na venda, esse item recebeu o nome de “item de venda”, utilizando o código da venda que o sistema busca automaticamente no banco de dados através de um comando SQL que incrementa o último código de venda registrado no sistema, ou seja, caso a última venda registrada no sistema seja a 103, o item adicionado no momento terá o código de venda 104, enquanto não fechar a venda e enviar para o banco de dados,



todos os produtos adicionados na venda, serão adicionados na venda 104, pois até o presente momento, a loja só possui um computador direcionado a esse serviço.

Figura 17: Adicionar item à venda

```

86 private void adicionarItem(){
87     String sql = "insert into tbitemvenda(codigovenda,quantidade,tbproduto_idproduto,subtotal) values (?, ?, ?, ?)";
88
89
90     try {
91         pst = conexao.prepareStatement(sql);
92         pst.setString(1, campoCodigoVenda.getText());
93         pst.setString(2, campoQtd.getText());
94         pst.setString(3, campoId.getText());
95         pst.setString(4, campoSubtotal.getText());
96
97         if ((campoQtd.getText().isEmpty())) {
98             JOptionPane.showMessageDialog(null, "Informe a quantidade!", "Error", JOptionPane.ERROR_MESSAGE);
99         } else if (campoId.getText().isEmpty()) {
100             JOptionPane.showMessageDialog(null, "Selecione o produto!", "Error", JOptionPane.ERROR_MESSAGE);
101         } else {
102             int adicionado = pst.executeUpdate();
103             if (adicionado > 0) {
104                 total = total + Double.parseDouble(campoSubtotal.getText());
105                 campoTotal.setText(String.valueOf(total));
106                 reduzir();
107                 pesquisarItens();
108                 pesquisar();
109                 campoId.setText(null);
110                 campoQtd.setText(null);
111             } else {
112                 JOptionPane.showMessageDialog(null, "Não foi possível incluir esse produto!", "Error",
113                 JOptionPane.ERROR_MESSAGE);
114             }
115         }
116     }

```

Fonte: Elaboração própria

Após adicionar todos os itens da venda, é necessário finalizar a venda, onde a venda vai ser armazenada com os dados relacionados aos produtos que foram adicionados, como o valor total da venda, no qual serão somados todos os subtotais, o desconto que é opcional no sistema, a forma de pagamento e a data da venda que o sistema pega automaticamente através do atributo 'date' no banco de dados.

Na Figura 18 é possível observar os itens já adicionados na venda, faltando apenas finalizar a venda.



Figura 18: Tela de venda com itens adicionados

Mago Som - Sistema de Gerenciamento de Estoque

Gerenciamento Cadastro Consulta Ajuda Opções

**Nova Venda:** 19 15/12/17

Produto:

Código	Nome	Preço	Estoque
1	Pastilhas de Freio Dianteira Saveiro	67.20	25
2	Disco de freio dianteiro	163.80	18
3	Escapamento esportivo yamaha R1	2016.00	8

Quantidade: 1 Valor Unit: 163.80 Subtotal: 163.8

Código	Produto	Quantidade	Valor	SubTotal
...	2 Trava Elétrica	1.0	11.00	11.0
...	21 Pastilhas de Freio Dianteira Saveiro	1.0	67.20	67.2
...	20 Amortecedor porta traseira Clio	1.0	45.50	45.5
...	19 Sensor nível combustível	2.0	50.40	100.8
...	17 Camera de segurança veicular	1.0	67.20	67.2
...	16 Camera de re estacionamento	1.0	27.30	27.3
...	4 Fio	2.0	50.00	100.0
...	3 Alarme	1.0	225.00	225.0
...	25 Kit de molas esportivas Montana	1.0	504.00	504.0
...	26 Carburador Gol Parati Voyage Saveiro G1	1.0	638.40	638.4
...	27 Inversor de força 600w	1.0	655.20	655.2

Forma de Pagamento: Dinheiro Desconto:  Total: 2508.8

**Finalizar Venda**

Fonte: Elaboração própria

Após finalizar a venda é gerado um comprovante não fiscal informando os produtos vendidos na respectiva venda conforme na Figura 19.

Figura 19: Comprovante da venda realizada



**Venda nº: 19**

Produto	Qtd	SubTotal
Trava Eletrica	1	R\$ 11,00
Pastilhas de Freio Dianteira Saveiro	1	R\$ 67,20
Amortecedor porta traseira Clio	1	R\$ 45,50
Sensor nivel combustivel	2	R\$ 100,80
Camera de segurança veicular	1	R\$ 67,20
Camera de re estacionamento	1	R\$ 27,30
Fio	2	R\$ 100,00
Alarme	1	R\$ 225,00
Kit de molas esportivas Montana	1	R\$ 504,00
Carburador Gol Parati Voyage Saveiro G1	1	R\$ 638,40
Inversor de força 600w	1	R\$ 655,20
Pastilhas de Freio Dianteira Saveiro	1	R\$ 67,20
<b>Valor Total:</b>		<b>R\$ 2.508,80</b>

Fonte: Elaboração própria

### 3.4 RESULTADOS OBTIDOS

Este trabalho atingiu com sucesso todos os objetivos esperados, para isto foi necessário obter êxito nos seguintes objetivos: a identificação do problema a ser resolvido, levantamento de requisitos do sistema, modelagem do sistema utilizando UML e prototipação da interface do sistema foi obtido o documento de requisitos, que pode ser consultado no APÊNDICE A.

Quanto ao desenvolvimento do sistema em Java e testes do sistema com o usuário obteve-se o Sistema de Gerenciamento de estoque da loja Mago som denominado Mago System, possibilitando o controle do estoque automatizado ao realizar compras e vendas de produtos sem a possibilidade de alterar de forma manual a quantidade do estoque, evitando assim erros e perdas de dados dos produtos.

No que se refere a relatórios, foi atendido todos os pedidos do cliente, onde foi solicitado 9 tipos de relatórios que o mesmo achou necessário e já está usando diariamente o sistema que foi entregue.

Então é possível afirmar que o sistema obteve 100% dos requisitos atendidos, obtendo total satisfação do cliente.

## **4. CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES**

### **4.1 CONSIDERAÇÕES FINAIS**

Este relatório descreveu as atividades desenvolvidas durante o período de estágio supervisionado na Mago Som & Acessórios, detalhando o desenvolvimento de uma solução para gerenciar o estoque e as transações da empresa.

O estágio proporcionou um grande aprendizado, pois permitiu vivenciar novos desafios que servirão na resolução de problemas futuros durante a carreira de trabalho.

Durante a realização do estágio foi possível aplicar conhecimentos adquiridos durante a graduação, principalmente no que diz respeito a desenvolvimento de software, incluindo modelagem, prototipação, desenvolvimento na linguagem Java e testes de interface com o usuário. Além desses conhecimentos adquiridos, foi necessário se aprofundar por exemplo, em outros assuntos que foram necessários para atender as expectativas do cliente.

Dentre as dificuldades encontradas durante o período de estágio, alguns exemplos são: compreender como funcionava um controle de estoque de uma empresa, como seria feito as compras e vendas dos produtos, como seriam feitas essas transações dentro do banco de dados. Então foi preciso buscar na internet alguns softwares de controle de estoque genéricos, para observar como eram suas interfaces e quais dados eram necessários armazenar e posteriormente buscar no banco de dados.

Também foi preciso buscar alguns modelos de diagramas de relacionamento de sistemas de controle de estoque na internet para entender como funcionava um sistema de estoque por dentro, assim possibilitando entender para posteriormente criar um banco de dados que seguisse as normas já estabelecidas, mas com informações que o cliente solicitou e não estavam disponíveis nos modelos encontrados.

Conclui-se que o objetivo do estágio foi alcançado com total satisfação do cliente, pois agora é possível armazenar os dados de forma rápida e segura podendo ser encontradas facilmente dentro do banco de dados além da possibilidade de emitir os relatórios que o cliente solicitou, que não podiam ser emitidos anteriormente devido aos dados serem armazenados fisicamente,.

## **4.2 RECOMENDAÇÕES**

Apesar da solução atender os objetivos que foram apresentados pelo cliente inicialmente, é possível identificar algumas funcionalidades que podem ser integradas afim de oferecer um sistema mais completo. Neste sentido recomenda-se que a solução seja estendida adicionando o recurso de comunicação com uma máquina de nota fiscal eletrônica, possibilitando a emissão de notas fiscais que é obrigatória para as empresas.

## REFERÊNCIAS BIBLIOGRÁFICAS

HIPER. **Sistema de controle de estoque: entenda por que você precisa de um.** Disponível em: <<https://gohiper.com.br/blog/sistema-controle-de-estoque/>>. Acesso em: 18 fev. 2018.

BOOCH, Grady. **UML: guia do usuário.** Rio de Janeiro: Elsevier, 2005 - 11ª Reimpressão.

KENDALL, K.E., Kendall, J.E., Systems Analysis and Design, 8th edition, Prentice Hall, 2010.

PRESSMAN, Roger S. **Engenharia de software.** São Paulo: Pearson Makron Books, 1995.

WAZLAWICK, Raul S. **Análise e projeto de sistemas de informação orientada a objetos.** 2 ed. – Rio de Janeiro: Elsevier, 2011.

SUEHRING, Steave - **MySQL a Bíblia** – Ed. Campus

JAVA. **Obtenha Informações sobre a Tecnologia Java.** Disponível em: <[https://www.java.com/pt\\_BR/about/](https://www.java.com/pt_BR/about/)>. Acesso em: 05 out. 2017.

SANTOS, Adriano. **MySQL: Quem é você?** Disponível em: <<https://www.devmedia.com.br/mysql-quem-e-voce/1752>>. Acesso em: 09 nov. 2017

## APÊNDICE

**APÊNDICE A – DOCUMENTO DE REQUISITOS**  
**DOCUMENTO DE REQUISITOS DE SOFTWARE**

**Mago Som System**  
Versão 1.0

**Desenvolvedores/Analistas**

Lucas Campelo Vieira

**Rio Branco – AC**  
**2018**



## **1. ANÁLISE DO PROBLEMA**

Foi verificado que a loja Mago Som & Acessórios possuía dificuldade em gerenciar seu estoque devido o controle ser todo manuscrito em grandes planilhas sem a possibilidade de gerar relatórios e de controlar as compras e vendas efetuadas no estabelecimento.

## **2. NECESSIDADES BÁSICAS DO CLIENTE**

O proprietário apresentou uma proposta solicitando um software que:

- Elimine totalmente as planilhas manuscritas devido ao difícil armazenamento e controle dos dados;
- Tenha fácil inserção e controle dos dados pois o usuário não possui muitas habilidades computacionais;
- Gere diversos tipos de relatórios de acordo com suas necessidades atuais: Compras realizadas, vendas realizadas, produtos disponíveis, produtos indisponíveis, todos os produtos, fornecedores, finanças da loja e produtos com preço de custo.

## **3. ESTUDO DE VIABILIDADE**

### **3.1 VIABILIDADE TÉCNICA**

Existem todos os recursos técnicos necessários para a execução do projeto atendendo suas especificações

### **3.2 VIABILIDADE ECONÔMICA**

O projeto não possui investimento econômico, tudo será feito gratuitamente.

### **3.3 VIABILIDADE LEGAL**

O projeto não infringe nenhuma lei, portanto todos os recursos utilizados são permitidos pela legislação.

## **4. MISSÃO DO SOFTWARE**

Ser um Sistema Desktop para gerenciar o estoque da loja Mago Som & Acessórios de maneira fácil e rápida, atendendo todas as necessidades do cliente.

## 5. LIMITES DO SISTEMA

ID	FUNCIONALIDADE	JUSTIFICATIVA
L1	Pequena capacidade	Gerenciamento pequeno, usando um banco de dados local.
L2	Rede LAN	O sistema terá um funcionamento através da Rede Local, sem uso de internet.

## 6. BENEFÍCIOS GERAIS

ID	BENEFÍCIOS
B1	Rápido acesso aos dados
B2	Controle de entrada e saída de mercadorias
B3	Geração de relatórios
B4	Registros de compras e vendas armazenadas no banco de dados

## 7. RESTRIÇÕES

ID	RESTRIÇÃO	DESCRIÇÃO
R1	Rede Local	Sistema Desktop funcionando em Rede sem necessidade de internet.

## 8. ATORES

ID	ATORES	DESCRIÇÃO
A1	Cliente	Usuário comum
A2	Usuário	Usuário máster

## 9. REQUISITOS FUNCIONAIS

ID	FUNCIONALIDADE	NECESSIDADES	PRIORIDADE
RF1	CRUD Usuário	Classe usuário com as 4 operações CRUD ( <i>Create, Read, Update e Delete</i> ).	Alta
RF2	CRUD Fornecedor	Classe Fornecedor com as 4 operações CRUD ( <i>Create, Read, Update e Delete</i> ).	Alta
RF3	CRUD Produto	Classe Produto com as 4 operações CRUD ( <i>Create, Read, Update e Delete</i> ).	Alta
RF4	CRUD Vendas	Classe Vendas com as 4 operações CRUD ( <i>Create, Read, Update e Delete</i> ).	Alta
RF5	Gerar relatórios	SuperUsuário poderá gerar relatórios (Todos os produtos; Produtos disponíveis; Produtos indisponíveis; Produtos por nome; Detalhamento do preço; Controle Financeiro do Estoque).	Média

## 10. REQUISITOS NÃO-FUNCIONAIS

ID	REQUISITOS	DESCRIÇÃO	CATEGORIA
NRF1	Login e Senha	O Sistema solicitará Login e Senha fazendo com que somente pessoas autorizadas utilizem o sistema.	Segurança
NRF2	Múltiplos usuários	O Sistema possibilita o acesso de múltiplos usuários através de login e senha.	Usabilidade
NRF3	Campo com as instruções de uso	A interface do sistema vai auxiliar o uso de pessoas com dificuldade, quando o usuário colocar a seta do mouse em um campo de texto, aparecerá o que deve ser feito no mesmo.	Usabilidade

## 11. REQUISITOS DE HARDWARE

### 11.1 CONFIGURAÇÃO MÍNIMA

- Sistema operacional: Windows XP e posterior
- CPU: Pentium 3 de 500 Mhz
- Memória de sistema (RAM): 512 MB
- Disco rígido: 500 MB de espaço livre
- Velocidade de rede: 128 Kbits/s
- Placa gráfica: DirectX 9 e 3D, com 64 MB de VRAM
- Tela: 1024x768, "High Color de 16 bits" – DirectX 9 (para ser executado em modo Direct X)

### 11.2 CONFIGURAÇÃO RECOMENDADA

- Sistema operacional: Windows 7, 8 e 10.
- CPU: Pentium 4 de 2,4 GHz+ ou AMD 2400xp+
- Memória do sistema (RAM): 1 GB ou mais
- Disco rígido: 2 GB de espaço livre ou mais
- Velocidade de rede: 768 Kbits/s
- Placa gráfica: DirectX 9 e 3D, com 256 MB de VRAM
- Tela: 1280 x 1024, "True Color de 32 bits"

## 12. FERRAMENTAS DE DESENVOLVIMENTO E LICENÇA DE USO

- a. NetBeans IDE.
- b. MySQL.

**APÊNDICE B – PROTÓTIPOS DO SISTEMA**  
**PROTÓTIPOS DO SISTEMA**

**Mago Som System**  
Versão 1.0

**Desenvolvedores/Analistas**

Lucas Campelo Vieira

**Rio Branco – AC**  
**2018**

Os protótipos apresentados a seguir, foram desenhados com base nos requisitos e nas descrições do cliente, de como ele esperaria que a interface fosse feita.

### 1. Tela de Login

Mago System - Sistema de Gerenciamento de Estoque

*Mago Som*

Sistema de Gerenciamento de Estoque

Login:

Senha:

Acessar

### 2. Tela Principal

Mago System - Sistema de Gerenciamento de Estoque

Gerenciamento	Cadastro	Consulta	Ajuda	Opções
Nova Venda	Produto	Produto	Sobre	Sair
Nova Compra	Fornecedor	Fornecedor		
Relatórios				

*Mago Som*

### 3. Tela de Compras de produtos

Mago System - Sistema de Gerenciamento de Estoque

Gerenciamento Cadastro Consulta Ajuda Opções

Nova Compra

Nova Compra:  01/01/18

Produto:  Quantidade:

+ -

Código	Produto	Quantidade
<input type="text"/>		

Finalizar Venda

### 4. Tela de vendas de produtos

Mago System - Sistema de Gerenciamento de Estoque

Gerenciamento Cadastro Consulta Ajuda Opções

Nova Venda

Nova Venda:  01/01/18

Produto:

Quantidade:  Valor unt.  Subtotal:  + -

Código	Produto	Quantidade	Valor	Subtotal
<input type="text"/>				

Forma de Pagamento:  Desconto:  Total:

Finalizar Venda