***AUTOMATIC*: *SOFTWARE* *DESKTOP* QUE BUSCA AUTOMAÇÃO NO PROCESSO DE GERENCIAMENTO DE UMA MERCEARIA.**

Ulisses Andrade Carvalho, [gfulisses3@gmail.com](mailto:gfulisses3@gmail.com)¹

¹Instituto Federal de Minas Gerais – *Campus* São João Evangelista, Avenida Primeiro de Junho, 1043 – São João Evangelista – MG.

**RESUMO**

Devido à necessidade geral da nossa população por um tempo mais otimizado e uma maior facilidade para a tomada de decisões vitais para uma mercearia, esse trabalho foi construído buscando construir um *software* *desktop* de gerenciamento de mercearia que consiga atender as necessidades de uma mercearia local. Para a construção do projeto foram usados os conceitos de analise e projeto de sistemas que ajudou durante a fase inicial do projeto com a elicitação de requisitos e a analise desses requisitos, conceitos do SOLID que contribuem para o desenvolvimento do *software*, estimulando o reaproveitamento do código, facilitando na refatoração, diminuindo acoplamentos e separando responsabilidades, a linguagem de programação java que é uma linguagem já muito conhecida e difundida que possui compatibilidade com códigos orientados a objetos e um conjunto de bibliotecas que suprem todas as necessidades básicas para o desenvolvimento desse projeto e um servido de banco de dados MySQL que é atualmente um dos mais utilizados mundialmente contando com interface simples e sendo gratuito. Para a metodologia desse projeto foram utilizadas algumas listas e tabelas que descrevem as etapas do projeto, recursos necessários e o cronograma das atividades. Os resultados do projeto de certa forma foram satisfatórios levando em conta o tempo disponível para sua realização que não fora suficiente, dessa forma, foi implementado toda a parte de cadastro e armazenamento de entidades em um banco de dados, porém não foi possível programar a parte de gerenciamento desses dados.

**Palavras-chave:** Mercearia. Gerenciamento. *Software*. *Desktop*.

1. INTRODUÇÃO

Segundo o escritor americano Joseph Wood Krutch (1893-1970) “a tecnologia torna as grandes populações possíveis, e as grandes populações tornam a tecnologia indispensável” (apud BUCHSBAUM; BUCHSBAUM, 2006, p. 238). A partir dessa frase podemos compreender o nosso cenário atual onde já temos uma grande população, que ainda está crescendo, que demanda cada vez mais de novas tecnologias para auxiliar no dia a dia. Para que consigamos suprir essa demanda devemos buscar sempre uma organização racional do tempo e do espaço.

Trazendo todo esse pensamento para um ramo de negócios vemos uma grande necessidade na utilização de *softwares* de gerenciamento de dados em empresas de todos os tipos, pois as que não fazem a utilizações desses *softwares* acabam por ficarem atrasadas e defasadas.

Dessa forma foi observada uma grande necessidade em uma mercearia local por um *software* de gerenciamento para tal, que gerencie as vendas, cadastros, estoque e faça relatórios, garantindo todos os requisitos do cliente, que consiga também garantir a segurança de todos os dados, que seja reutilizável, de fácil manutenção e de fácil utilização de forma que qualquer pessoa possa utilizar com um mínimo de explicação sobre o mesmo. Assim o tempo será otimizado graças a maior organização do trabalho no local do estabelecimento proporcionada pelo *software*, aumentando consideravelmente a produtividade do local. Além disso, o programa ajudara na tomada de decisões a partir da coleta e organização de dados de vendas, compras entre outros.

Esse projeto será realizado com o auxilio dos professores do Instituto Federal de Minas Gerais (IFMG) em suas respectivas matérias, sendo eles: Dayler Vinícius Miranda Alves, Denis Rocha de Carvalho, Ítalo Magno Pereira, Rosinei Soares de Figueiredo. O projeto ainda contará com o auxilio de comerciantes locais, assim sendo viável a observação e entendimento de suas necessidades de automação na gerencia de seus respectivos estabelecimentos.

* 1. Objetivos

Resolvendo o problema, foi traçado o objetivo geral do trabalho, sendo esse, desenvolver um *software* *desktop* para o gerenciamento de uma mercearia que atenda todas as necessidades do local de forma a melhorar o seu funcionamento.

Devido ao tamanho e complexidade do trabalho foram definidos alguns objetivos específicos:

1. Realizar um *Brainstorming* com os funcionários de uma mercearia local;
2. Realizar uma entrevista com os donos de uma mercearia local;
3. Realizar uma observação direta em uma mercearia local;
4. Realizar a analise dos requisitos do *software*;
5. Preparar o ambiente de desenvolvimento;
6. Programar a aplicação;
7. Construir o banco de dados;
8. Testar a aplicação e promover as adaptações necessárias;
9. Entregar o trabalho;
10. REFERENCIAL TEÓRICO

É aqui que serão definidos e tratados os assuntos teóricos relacionados ao projeto, que serão divididos em subseções. O referencial teórico possui uma base geral nas aulas do curso técnico em informática do IFMG, com destaque para as matérias de Analise e Projeto de Sistemas, Banco de Dados I e Linguagem de Programação I.

* 1. Análise e projeto de sistemas

O processo de análise e projeto de sistemas, pode ser dividido em elicitação e análise de requisitos, conceitos fundamentais da engenharia de *software*, sendo essas as fases iniciais do desenvolvimento de qualquer projeto. Durante a elicitação a equipe responsável pelo projeto começa a levantar requisitos do *software*, sendo que esse time utiliza de várias técnicas para tal. Com esses requisitos em mente começa a fase de análise de requisitos.

A fase de análise e projeto de sistemas, possuí uma enorme importância para qualquer projeto, sendo a base para tal. Porém geralmente não é empregado muito esforço nessa fase do projeto por vários motivos, o que leva ao mau entendimento das necessidades do *software*, levando a construção de um *software* totalmente diferente do proposto. Para evitar tais problemas deve se dar a atenção necessária a essa fase, pois se isso não for feito todo o desenvolvimento do *software* será prejudicado, o que pode ser visto na [Figura 1] a seguir:

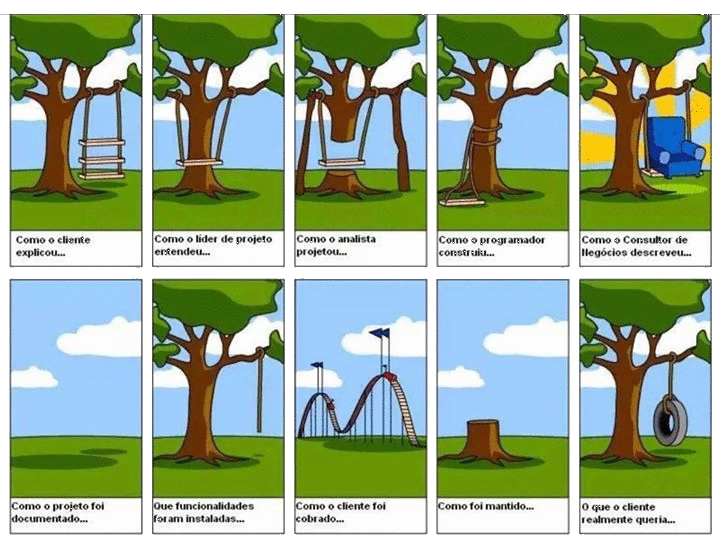


Figura 1: Erros comuns da analise e projeto de sistemas

Fonte: TEIXEIRA; Danielle, 2018.

* + 1. *Elicitação de requisitos*

Elicitar pode ser entendido como um processo cujo objetivo e obter algo, assim a elicitação busca obter requisitos, o que é de extrema importância para o projeto e deve ser feito de forma muito bem organizada. Assim para realiza-lo são usadas várias técnicas que tem como objetivo auxiliar durante esse processo, sendo elas: Entrevista, Questionário, Observação Direta, Rastreamento do Processo, *Brainstorming*. Nesse trabalho o processo de Entrevista, *Brainstorming* e Observação Direta serão mais bem detalhados devido sua importância para a construção do projeto final.

A entrevista é uma ótima estratégia para se elicitar os requisitos de um projeto de *software*, pois nessa entrevista temos uma conversa entre a equipe desenvolvedora e o cliente, possibilitando que esses troquem informações sobre o que é desejado por cada parte. O *Brainstorming* geralmente é usado para obter informações de forma rápida sobre o problema. A observação direta é uma técnica extremamente útil para descobrir aspectos novos de um problema.

* + 1. *Análise de requisitos*

É no processo de Análise de Requisitos que documentamos e consolidamos as ideias levantadas durante a elicitação de requisitos. Na análise deve-se atentar a algumas questões básicas, como a sua funcionalidade (Função do *software*), interfaces externas (interação do *software*, usuário-*software*-*hardware*), desempenho (tempo de resposta), atributos (características sobre o produto) e restrições (padrões políticos e limitações). Devemos tomar cuidado com a análise de requisitos, elas devem ser corretas, não ambíguas, completas, consistentes, verificáveis e modificáveis.

Temos dois tipos de requisitos: os funcionais, que são funcionalidades do sistema a ser desenvolvido e são o reflexo das necessidades do cliente e os não funcionais, que estão normalmente atribuídos à robustez, segurança, organização e requisitos legais do sistema, e são ligados diretamente às necessidades do projeto.

* 1. Desenvolvimento de sistemas

A fase de desenvolvimento de sistemas engloba todo o processo de construção do *software*. Nessa fase descrevemos como exatamente o *software* irá funcionar utilizando-se dos dados da etapa anterior, que são passados para uma linguagem mais técnica e padroniza, de forma que o desenvolvedor consiga programar o *software*. Aqui também é onde o ambiente de desenvolvimento é preparado, escolhendo, de acordo com nossas necessidades as tecnologias, a quantidade e qualidade da mão de obra e ferramentas a serem utilizadas e também é onde por fim a aplicação e feita testada e entregada ao cliente.

Nessa fase é onde é gerado um valor ao *software*, pois aqui já temos algo que podemos observar e avaliar. Essa fase á de muita importância, porém, geralmente não é atribuído a essa fase tempo ou atenção suficiente, o que não deve ser feito, pois com um mau planejamento o desenvolvimento do sistema poderá apresentar muitos erros e imprevistos.

Durante o desenvolvimento de um *software* temos alguns princípios da programação orientada a objetos (POO), um dos mais conhecidos e utilizados conjuntos desses princípios é o SOLID que foi criado por Michael Feathers. Esses princípios têm como objetivo auxiliar o desenvolvedor a criar códigos estimulando o reaproveitamento do código, facilitando na refatoração, diminuindo acoplamentos e separando responsabilidades (SILVA, 2013).

* 1. A linguagem de programação Java

Java é uma linguagem de programação orientada a objetos muito conhecida em seu meio, a linguagem foi desenvolvida por uma equipe da Sun Microsystems (atualmente de propriedade da Oracle) e lançada em 1995. Desde então a linguagem se adaptou a várias mudanças que ocorreram desde seu lançamento até os dias atuais, o que garantiu sua popularidade (INDRUSIAK, 1996).

O Java possui inúmeras características, mas dentre elas vamos destacar algumas das principais que foram disponibilizadas em The Java Language: A White Paper em 1995. Apresenta uma linguagem muito simples, prova disso é a utilização da linguagem para se introduzir o aprendizado na área de programação, conta também com uma eficiência de código orientado a objetos, com um conjunto de bibliotecas que suprem todas as necessidades básicas de um desenvolvedor, promove o reuso e manutenção de códigos, possui código interpretado e portável, o que possibilita serem executadas em diferentes plataformas e apresenta uma segurança ao se desenvolver, pois essa linguagem possui um esquema para garantir a integridade de código e conta com uma detecção prévia de possíveis erros (*The Java Language Specification*, 1996).

* 1. O servidor de banco de dados MySQL

O MySQL é um Sistema de Gerenciamento de Banco de Dados (SGBD), que fornece mecanismos para armazenar, organizar e recuperar dados, criado por David Axmark, Allan Larsson e Michael Wideniusem nas décadas de 80 e 90 e atualmente em 2020 é propriedade da Oracle (OFICINA DA NET, 2007). O MySQL utiliza uma linguagem de banco de dados relacionais para realizar consultas e manipular dados, sendo o SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) a linguagem padrão desse tipo de banco de dados (MILANI, 2006).

O MySQL é um dos bancos de dados mais utilizados da atualidade, tendo constantes atualizações para melhorias do *software* e conserto de bugs. O servidor de banco de dados MySQL possui várias boas características que contribuíram para a sua escolha para o desenvolvimento desse trabalho, sendo elas: Interface simples proporcionando uma grande facilidade de uso, velocidade de resposta graças sua implementação leve das instruções SQL e principalmente a utilização da ferramenta ser gratuita (MILANI, 2006).

* 1. O funcionamento da mercearia

Tudo começa com o cadastro de produtos que serão vendidos no local, fornecedores de produtos e também no mínimo um vendedor, um gerente e um cliente com o nome de “consumidor final”. Uma venda só pode ser feita por um vendedor, onde o cliente terá três opções de pagamento: a prazo, dinheiro ou cartão. Apenas clientes cadastrados poderão fazer compras a prazo. As informações das contas dos clientes serão guardadas e essas contas deverão ser pagas a um gerente, tendo as contas um prazo de vencimento, sendo que a partir do final do prazo poderá ser acrescentado um valor em porcentagem sobre o valor da conta do cliente.

Os produtos terão um gerenciamento de seu estoque, onde sempre que mais produtos sejam comprados, será necessário fazer a “entrada” desses produtos informando a quantidade de cada produto que foi comprado e seu fornecedor. Sempre que um produto for vendido o estoque desse produto ira ser subtraído pela quantidade comprada pelo cliente. Os produtos poderão ter data de validade, sendo que quando essa data for ultrapassada o estoque será atualizado automaticamente. O *software* ainda fara relatórios de todas as vendas e compras do estabelecimento e relatórios sobre a movimentação do estoque dos produtos, onde esses poderão ser solicitados a qualquer momento e por qualquer período de tempo, sendo os dados organizados pelo período de um dia. Ao final de qualquer venda, pagamento ou compra de produtos o relatório diário será atualizado.

1. METODOLOGIA

Nessa parte é onde assuntos relacionados ao planejamento do *software* propriamente dito serão tratados, sendo esses assuntos divididos em subseções, afim de uma maior organização. A metodologia possui uma base geral nas aulas do curso técnico em informática do IFMG-SJE, com destaque para a matéria de Projeto Integrador I.

* 1. Natureza do projeto

De acordo com os objetivos já definidos anteriormente, o projeto será realizado conforme as atividades do [Quadro 1] a seguir.

Quadro 1 - Atividades a serem desenvolvidas durante o projeto

|  |  |
| --- | --- |
| **Objetivo** | **Atividades** |
| Preparar o ambiente de desenvolvimento | a) Selecionar computador;  b) Instalar sistema operacional, se necessário;  c) Instalar o Java;  d) Instalar e configurar o NetBeans  e) Instalar e configurar o MySQL;  f) Testar o ambiente de desenvolvimento, criando um pequeno projeto. |
| Programar a aplicação | a) Criar os diagramas de classes;  b) Criar a Interface da aplicação;  c) Implementar os atributos e funcionalidades da aplicação. |
| Construir o banco de dados | a) Criar os diagramas de entidade e relacionamento;  b) Modelar o banco de dados;  c) Preparar o banco de dados. |
| Testar a aplicação e promover as adaptações necessárias | a) Testar as funcionalidades da aplicação;  b) Realizar testes com a presença do cliente, mostrando a ele as funcionalidades do *software*;  c) Analisar os resultados dos testes;  d) Fazer a correções dos erros, se necessárias. |
| Entregar o trabalho | a) Conferir a integridade da aplicação;  b) Entregar o projeto. |

Fonte: Elaborado pelo autor, 2020.

* 1. Instrumentos Utilizados

Os recursos que serão utilizados para a realização do projeto ou são *softwares* e linguagens de uso gratuito ou serão de propriedade dos pesquisadores. A seguir serão listadas as principais ferramentas e tecnologias que serão utilizadas para todo o desenvolvimento da aplicação:

1. Computador para o desenvolvimento: equipamento essencial para a realização de pesquisas, de programação da aplicação, de construção do banco de dados e elaboração de qualquer tipo de documentação do projeto. Para suprir a necessidades dos programas a serem usados para o desenvolvimento das atividades acima o computador deve atender também a seus requisitos mínimos, especificado em seus respectivos sites;
2. Pacote Microsoft Office: conjunto de *softwares* da Microsoft que auxiliam na criação, modificação e manutenção de documentos. Deve ser utilizada a versão do Microsoft Office 10 ou superior. Será usado para fazer a documentação do projeto;
3. NetBeans: é um Ambiente de Desenvolvimento Integrado (IDE) que auxilia no desenvolvimento de códigos na linguagem Java. Para esse projeto deve ser utilizada a versão 8.2 ou superior;
4. MySQL: SGBD que será usado para a construção do banco de dados do projeto, deve ser usada a versão 8.0 ou superior.
   1. Cronograma das atividades

As tarefas serão distribuídas em torno do terceiro trimestre que começa no dia 4 de janeiro até dia 12 de março, ou seja, um período de 10 semanas. A seguir o [Quadro 2] que apresenta a distribuição das atividades em torno do período disponível para a realização do trabalho.

Quadro 2 - Cronograma de realização das atividades do projeto

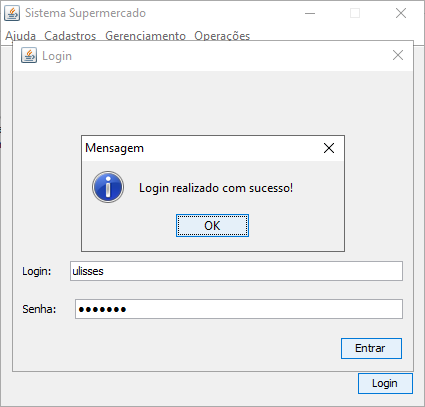
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Atividades | Semanas | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Selecionar computador. | X |  |  |  |  |  |  |  |  |  |
| Instalar sistema operacional, se necessário. | X |  |  |  |  |  |  |  |  |  |
| Instalar o Java. | X |  |  |  |  |  |  |  |  |  |
| Instalar e configurar o NetBeans. | X |  |  |  |  |  |  |  |  |  |
| Instalar e configurar o MySQL. | X |  |  |  |  |  |  |  |  |  |
| Testar o ambiente de desenvolvimento. | X |  |  |  |  |  |  |  |  |  |
| Criar os diagramas de classes. |  | X | X |  |  |  |  |  |  |  |
| Criar a Interface da aplicação. |  |  | X | X | X | X |  |  |  |  |
| Implementar os atributos e funcionalidades da aplicação. |  |  |  | X | X | X | X |  |  |  |
| Criar os diagramas de entidade e relacionamento. |  | X | X |  |  |  |  |  |  |  |
| Modelar o banco de dados. |  |  | X | X | X |  |  |  |  |  |
| Preparar o banco de dados. |  |  |  | X | X | X |  |  |  |  |
| Testar as funcionalidades da aplicação. |  |  | X | X | X | X | X |  |  |  |
| Realizar testes com a presença do cliente. |  |  |  |  |  |  |  | X |  |  |
| Analisar os resultados dos testes. |  |  |  |  |  |  |  | X | X |  |
| Fazer a correções dos erros, se necessárias. |  |  |  |  |  |  |  |  | X | X |
| Conferir a integridade da aplicação. |  |  |  |  |  |  |  |  |  | X |
| Entregar o projeto. |  |  |  |  |  |  |  |  |  | X |
| Elaborar a documentação do projeto. | X | X | X | X | X | X | X | X | X | X |

Fonte: Elaborado pelo autor, 2020.

1. **RESULTADOS**

O *software* quando iniciado abre uma tela principal, que tem a função de fazer a ligação entre todas as telas de cadastro, vendas e gerenciamento. Porém para se acessar qualquer funcionalidade do *software* precisa-se realizar o login de um funcionário, esses logins são divididos em três níveis de acesso, sendo que cada um dá permissões de acesso a funcionalidades especificas do *software*. Também na tela de login, porém em um primeiro momento de inicialização do *software* onde não há nenhum funcionário cadastrado no sistema ao se clicar no botão de entrar o mesmo redireciona o usuário a uma tela de cadastro de funcionário para que o mesmo possa cadastrar um funcionário com nível de acesso máximo. Veja a seguir a [Figura 2]:

Figura 2: Tela de login do *software* mostrando um login realizado com sucesso



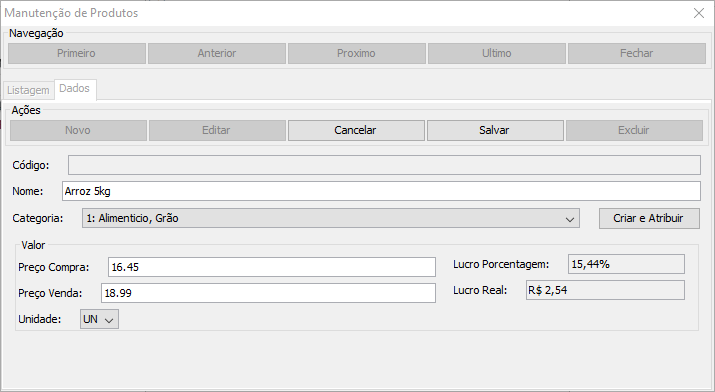
Fonte: Elaborado pelo autor, 2021.

Na tela principal como pode ser observado acima podemos ver um menu de ajuda onde no dado momento apenas tem um item de menu que retorna uma mensagem simples sobre o sistema. Em seguida temos o menu de cadastros que contem vários item de menus para o cadastro de categoria de produtos, cliente, endereço, fornecedor, funcionário, login que só pode ser acessado pelo cadastro de funcionários e produto.

Todos esses cadastros tem uma logica bem semelhante mudando apenas alguns detalhes entre os mesmos. Dentre esses detalhes vale destacar os formulários que contem datas, pois essas datas tem uma classe de conversão que faz a conversão da data em escrita pelo usuário em um objeto para que esse possa ser armazenado.

Temos também os formulários “auxiliares” que são chamados por outros formulários que precisam de outros objetos (de forma simples pode ser entendido como uma representação virtual de algo real) para serem cadastrados, esses formulários “auxiliares” quando chamados possibilitam apenas a edição ou a criação de apenas um objeto.

Pode ser destacado também o formulário de cadastro de produtos que quando informado o preço de compra e o preço de venda de um produto o formulário apresentara em outro campo o lucro em porcentagem e o lucro real que será obtido pela venda do mesmo. Veja a seguir a [Figura 3]:

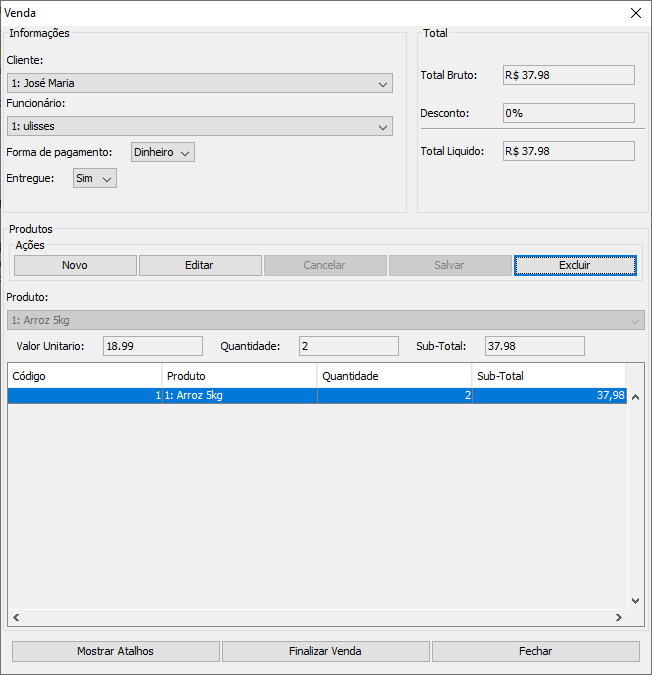
Figura 3: Tela de manutenção de produtos mostrando um produto sendo cadastrado

Fonte: Elaborado pelo autor, 2021.

Ainda na tela principal temos o menu gerenciamento que possui quatro itens de menu, sendo eles: contas, entregas, estoque e financeiro. Os dois primeiros itens de menu ainda não forem implementados. Já o formulário de estoque possui no momento apenas uma tabela que permite a visualização da quantidade de cada produto em estoque e o valor que se tem por cada item, sendo esse subtotal o resultado da multiplicação da quantidade pelo valor de venda dos produtos. O formulário financeiro também está no mesmo nível que o de estoque, possuindo apenas uma tabela para as compras e uma para as vendas.

Agora indo para o principal do programa as telas de venda e compra, que são acessadas pelo menu operações e selecionando o item de menu desejado (venda ou compra). As duas telas são praticamente idênticas, porém uma vai realizar apenas compras e armazenar seus dados na lista de compras e a venda ira realizar vendas e armazenar os dados na lista de vendas. Outras diferenças entre esses formulários são o campo de cliente e fornecedor, onde o fornecedor aparece apenas no de compra e o cliente apenas no de venda, e o formulário de vendas ainda possui um requisito a mais, que é verificar antes de adicionar um item à venda se o estabelecimento possui um estoque para suprir a demanda. Veja a seguir a [Figura 4]:

Figura 4: Tela de Venda do *software* mostrando a realização de uma venda



Fonte: Elaborado pelo autor, 2020.

1. CONSIDERAÇÕES FINAIS

Primeiramente vale destacar que o projeto como um todo teve um resultado de certa forma gratificante, com toda a parte básica do projeto funcionando perfeitamente. O *software* até o momento consegue realizar o cadastro de todos os itens propostos, consegue realizar vendas e compras tudo forma bem simples e funcional, além de realizar as operações de venda e compra tendo varias funcionalidades, como exemplo: a atribuição de descontos, juros e prazo de vencimento da compra. Contudo ainda falta implementar algumas partes do *software* para que o mesmo fique completo segundo o planejamento do projeto. Para o complemento total do projeto deve-se desenvolver a parte de gerencia do projeto, que foi apenas iniciada e desenvolver uma interface gráfica mais elaborada para que o programa fique mais intuitivo para o usuário.

Dessa forma fica para um futuro próximo a melhoria dessas interfaces gráficas de forma e melhorar a experiência do usuário proporcionando uma maior produtividade no uso do *software* e o desenvolvimento da parte de gerencia do *software*, disponibilizando ao usuário várias informações sobre seu negocio organizadas de diferentes formas, proporcionando assim ao usuário tomar decisões melhor e mais rapidamente e também um mais fácil gerenciamento do estabelecimento.

1. REFERÊNCIAS

BUCHSBAUM, André; BUCHSBAUM, Paulo (Org.). **Do bestial ao genial: frases da política**. Rio de Janeiro: Ediouro, 2006.

GOSLING, J.; JOY, B.; STEELE,G. **The Java Language Specification**. Sun Microsystems, 1996. Disponível por WWW em <http://java.sun.com/doc/language_specification.html>. Acesso em: 06/12/2020.

INDRUSIAK, Leandro Soares. **Linguagem Java.** 1996. Disponível em: <https://www.cin.ufpe.br/~arfs/introjava.pdf>. Acesso em: 06/12/2020.

MILANI, André. **MySQL**: Guia do Programador. São Paulo: Novatec, 2006. Disponível em: [https://books.google.com.br/books?hl=pt-BR&lr=&id=81EwMDA-pC0C&oi=fnd&pg=PA19&dq=MySQL+banco+de+dados+&ots=xPwraboT\_G&sig=p4CyrSrS-PuCamg\_vD-4ieUO1i4#v=onepage&q&f=false](https://books.google.com.br/books?hl=pt-BR&lr=&id=81EwMDA-pC0C&oi=fnd&pg=PA19&dq=MySQL+banco+de+dados+&ots=xPwraboT_G&sig=p4CyrSrS-PuCamg_vD-4ieUO1i4" \l "v=onepage&q&f=false). Acesso em: 12/12/2020.

OFICINA DA NET, **Conheça um pouco sobre o MySQL**. 2007. Disponível em: [https://www.oficinadanet.com.br/artigo/390/conheca\_um\_pouco\_sobre\_o\_mysql#:~:text=As%20principais%20caracter%C3%ADsticas%20que%20destacam,de%20desenvolvimento%20de%20sites%20din%C3%A2micos%2C](https://www.oficinadanet.com.br/artigo/390/conheca_um_pouco_sobre_o_mysql" \l ":~:text=As principais características que destacam,de desenvolvimento de sites dinâmicos%2C). Acesso em: 12/12/2020.

SILVA, Álvaro César Pereira da. Princípios S.O.L.I.D. de Design. *In*: SILVA, Álvaro César Pereira da. **Princípios S.O.L.D de design aplicativos na melhora de códigofonte em sistemas orientados a objetos**. Lavras: 2013. p. 55-71. Disponível em: [http://www.bsi.ufla.br/wp-content/uploads/2013/10/PRINC%C3%8DPIOS-S.O.L.I.D.-DE-DESIGN-APLICADOS-NA-MELHORIA-DE-C%C3%93DIGO-FONTE-EM-SISTEMAS-ORIENTADOS-A-OBJETOS.pdf](http://www.bsi.ufla.br/wp-content/uploads/2013/10/PRINCÍPIOS-S.O.L.I.D.-DE-DESIGN-APLICADOS-NA-MELHORIA-DE-CÓDIGO-FONTE-EM-SISTEMAS-ORIENTADOS-A-OBJETOS.pdf). Acesso em: 06/12/2020.

TEIXEIRA; Danielle. **Como escrever requisitos de *software* de forma simples e garantir o mínimo de erros no sistema/app??**. 09 feb. 2018. Disponível em: [https://medium.com/lfdev-blog/como-escrever-requisitos-de-*software*-de-forma-simples-e-garantir-o-m%C3%ADnimo-de-erros-no-sistema-app-74df2ee241cc](https://medium.com/lfdev-blog/como-escrever-requisitos-de-software-de-forma-simples-e-garantir-o-mínimo-de-erros-no-sistema-app-74df2ee241cc). Acesso em: 13/12/2020.