

FACULDADE DE BALSAS
CURSO DE SISTEMAS DE INFORMAÇÃO

**DESENVOLVIMENTO DE UM SISTEMA PARA O CONTROLE DO
ESTOQUE DA EMPRESA AUTO MOLAS JM**

ADRIANO PEREIRA DA COSTA

FACULDADE DE BALSAS
CURSO DE SISTEMAS DE INFORMAÇÃO

**DESENVOLVIMENTO DE UM SISTEMA PARA O CONTROLE DO
ESTOQUE DA EMPRESA AUTO MOLAS JM**

Por

Adriano Pereira da Costa

Trabalho de conclusão de curso II
apresentado como exigência para
obtenção do título de Bacharel no Curso
de Sistemas de Informação à Faculdade
de Balsas, sob orientação do Prof. Me.
Junior Marcos Bandeira.

BALSAS – MA
2013

FACULDADE DE BALSAS
CURSO DE SISTEMAS DE INFORMAÇÃO

A Comissão Examinadora, abaixo assinada, aprova o trabalho de
Conclusão de Curso (TCC).

**DESENVOLVIMENTO DE UM SISTEMA PARA O CONTROLE DO
ESTOQUE DA EMPRESA AUTO MOLAS JM**

Elaborada por

Adriano Pereira da Costa

Como requisito para obtenção do título de Bacharel em Sistemas de Informação

BANCA EXAMINADORA

Prof. Msc. Junior Marcos Bandeira
Prof. Orientador

Prof. Msc. Jefferson Fontinele da Silva
Membro da Banca Examinadora

Prof. Msc. Fábio Roberto Pillatt
Membro da Banca Examinadora

AGRADECIMENTOS

Agradeço primeiramente a Deus por tudo que ele tem proporcionado em minha vida, pelo o que ele tem me dado.

Agradeço a minha família por todo o apoio e força que eles me oferecem, em especial a minha mãe que sem ela eu não estaria realizando esse sonho, a minha companheira, esposa e amiga Jainara pelo incentivo que ela me deu, aos meus avós, às minhas tias Wanda e Cléia, que sempre me ajudaram para estar aqui hoje realizando mais essa conquista.

Obrigado a todos que nunca deixaram de acreditar em mim.

Há conhecimento de dois tipos: sabemos sobre um assunto, ou sabemos onde podemos buscar informação sobre ele.

Samuel Johnson

RESUMO

As empresas necessitam cada vez mais de suporte para seus negócios. O Software é um recurso que pode auxiliar no controle e gestão de um negócio. Com isso foi desenvolvido neste trabalho um software de controle de estoque. A especificação do sistema foi possível através de uma entrevista com o proprietário da empresa Auto Molas JM. O Software de controle de estoque da empresa Auto Molas JM é uma ferramenta para auxiliar no controle de entrada e saída de produtos no ramo de auto peças e oficina mecânica na cidade de São Raimundo das Mangabeiras. A ferramenta auxiliará no controle do estoque com cadastramento de produtos, de clientes e fornecedores, movimentação de entrada e saída dos produtos no estoque e relatório de produtos no estoque. Essa ferramenta desktop foi desenvolvida na linguagem Java através do ambiente de desenvolvimento NetBeans IDE versão 7.2.1. Conforme os requisitos levantados o sistema foi desenvolvido com suas principais funcionalidades, cadastro de produtos, de clientes, de fornecedores, movimentação dos produtos no estoque e relatório de produtos no estoque, alcançando os objetivos do trabalho proposto.

Palavras chaves: Software, Java e NetBeans

ABSTRACT

Companies increasingly need to support their business. The Software is a resource that can help control and management of a business. It was developed with this work a software inventory control. The system specification was made possible through an interview with the business owner Auto Molas JM. Software inventory control company Auto Molas JM is a tool to help control the input and output of products in the field of auto parts and machine shop in São Raimundo Mangabeiras. The tool will assist in inventory control with registration of products, customers and suppliers, handling input and output of products in stock and in stock products report. This desktop tool was developed in Java using the development environment NetBeans IDE version 7.2.1. As the requirements raised the system was developed with its main features, registration of products, customers, suppliers, handling of products in stock and product inventory report in achieving the goals of the proposed work.

Keywords: Software, Java and NetBeans

LISTA DE FIGURAS

Figura 1 - Modelo de Diagrama de casos de uso.....	20
Figura 2 - Ciclo do Scrum.....	23
Figura 3 - Escolha do projeto	27
Figura 4 - Edição de código.....	28
Figura 5 - Serviços Básicos.....	28
Figura 6 - Exemplo de interface gráfica.....	29
Figura 7 - Gerenciamento de Projeto	30
Figura 8 - Diagrama de Classe do Sistema CE.....	31
Figura 9 - Casos de Uso do Sistema CE.....	33
Figura 10 - Tela de autenticação do sistema.....	40
Figura 11 - Código que compara as informações de autenticação	41
Figura 12 - Tela principal do Sistema CE	42
Figura 13 - Trecho do código da tela principal do sistema	42
Figura 14 - Formulário de cadastro de produtos	43
Figura 15 - Relatório do estoque de produtos	44
Figura 16 - Tela de entrada dos produtos	44
Figura 17 - Tela de venda dos produtos.....	45

SUMÁRIO

1	INTRODUÇÃO	11
1.2	Objetivos	12
1.3	Justificativa	12
1.4	Metodologia	13
2	REVISÃO BIBLIOGRÁFICA	14
2.1	Metodologia de Desenvolvimento de Software	14
2.1.1	Premissas de desenvolvimento de sistemas	14
2.1.2	Fases da metodologia de desenvolvimento de sistemas	15
2.1.2.1	Definições das fases de desenvolvimento de sistemas	15
2.1.3	Metodologia Ágil	16
2.2	Processos de Software	17
2.2.1	Especificação de software	17
2.2.1.1	Engenharia e análise de requisitos	17
2.2.2	Projeto e implementação de software	18
2.2.3	UML (Linguagem de Modelagem Unificada)	19
2.2.3.1	Diagrama de casos de uso	19
2.2.3.2	Diagrama de classe	20
2.2.4	Implementação	20
2.2.4.1	Programação Orientada a Objetos	20
3	SCRUM	22
3.1	Ciclo do Scrum	22
3.2	Papéis do Scrum	23
3.3	Etapas do Sprint	23
4	LINGUAGEM JAVA	25
5	NETBEANS IDE	27
5.1	Suporte para as tecnologias mais recentes de Java	27
5.2	Bancos de Dados e Serviços	28
5.3	Desenvolvimento de Interface de Usuário rápida	29
5.4	Suporte	29
5.5	Projeto Exibição	30
6	DOCUMENTAÇÃO DO SISTEMA CE (Controle do Estoque)	31
6.1	Diagrama de Classe	31

6.2	Diagrama de Casos de Uso	32
6.2.1	Descrição dos casos de uso.....	33
7	SISTEMA CE (CONTROLE DO ESTOQUE)	40
7.1	Acesso ao sistema	40
7.2	Tela principal do sistema.....	41
7.3	Cadastro dos produtos	43
7.4	Relatório do estoque.....	43
7.5	Movimentação do estoque	44
8	CONCLUSÃO	47
	REFERÊNCIAS BIBLIOGRÁFICAS	48
	APÊNDICE	49

1 INTRODUÇÃO

Com o mundo cada vez mais tecnológico e informatizado, as empresas estão buscando a melhor forma de desenvolver suas funções perante o mercado e a sociedade em que estão envolvidas. Com isso as inovações tecnológicas vêm trazendo diversas soluções, para grandes ou pequenos problemas, muitas destas inovações são possíveis através de sistemas de informação. Sistemas usados no auxílio das atividades desenvolvidas pela empresa, alguns desses sistemas podem ser desenvolvidos em uma tecnologia chamada Java, que utiliza da Linguagem de Programação Orientada a Objeto.

Há muitos aplicativos e sites que funcionam somente com o Java instalado, e muitos outros aplicativos e sites são desenvolvidos e disponibilizados com o suporte dessa tecnologia todos os dias. O Java é rápido, seguro e confiável. A tecnologia Java está em todo lugar. Ela pode ser encontrada em laptops, datacenters, consoles de jogo, supercomputadores científicos, telefones celulares e até na Internet.

O Java é a base para diversos tipos de aplicações em rede e é o padrão global para o desenvolvimento e distribuição de aplicações móveis, jogos, conteúdo baseado na Web e softwares corporativos. Com muitos desenvolvedores em todo o mundo de uma forma eficiente, o Java permite que você desenvolva, implante e use aplicações e serviços estimulantes para notebooks e datacenters, consoles de games e supercomputadores científicos, telefones celulares e Internet. (JAVA.COM 2013).

Neste trabalho é apresentado uma proposta para o desenvolvimento de um sistema na linguagem Java para o controle do estoque da empresa Auto Molas JM. Tal solução possui cadastro de produtos, de clientes e fornecedores, movimentação de entrada e saída dos produtos do estoque e relatório de produtos no estoque. Atualmente a empresa não conta com um sistema para gerenciar seu estoque, o que traz muitas dificuldades de controle e gestão das mercadorias. Não é possível saber, por exemplo, quantos produtos estão disponíveis.

Para isso foi realizado uma análise da empresa, levantamento de requisitos necessários para o desenvolvimento do software, um projeto preparatório contendo toda a análise e os requisitos do sistema.

Efetivamente, a elaboração de software de computador é um processo iterativo de aprendizado, e o resultado que Baetjer chamaria de “essência de

software”, é uma incorporação de conhecimentos coletados, destilados e organizados, à medida que o processo é conduzido. (PRESSMAN 2006)

Esse processo é de grande importância, pois fornece estabilidade, controle e organização para essa atividade, que sem ele pode se tornar bastante desorganizada ou sem controle. Para chegar nesse resultado foi adotado uma metodologia ágil, para agilizar o processo.

Sendo assim, foi adotado o processo de desenvolvimento de software que prevê o levantamento de requisitos, projeto e desenvolvimento. Essas atividades foram conduzidas dentro da metodologia ágil, pois a mesma é mais adequada em função da equipe ser composta por somente um desenvolvedor e ao mesmo tempo para que se possa fazer entregas rápidas através de produtos de software intermediários em constante iteração e aprovação do cliente.

1.2 Objetivos

Objetivo geral

Desenvolver um sistema de controle de estoques para a oficina mecânica Auto Molas JM, para controlar a entrada e saída dos produtos do estoque.

Objetivos específicos:

- Fazer levantamento bibliográfico sobre conteúdos relacionados a proposta
- Fazer o levantamento de requisitos para o sistema
- Projetar os principais diagramas do sistema
- Implementar o sistema

1.3 Justificativa

A proposta em questão permitiu o desenvolvimento de um sistema de controle de estoque da empresa Auto Molas JM da cidade de São Raimundo das Mangabeiras. Essa ferramenta servirá para auxiliar a empresa verificar os produtos que a mesma possui para venda, isso permitirá um maior controle sobre os produtos que tem na empresa.

Com os dados registrados é possível a construção de relatório em relação a quantidade de produtos. Com isso os gestores da empresa podem manter um

estoque daqueles produtos que vendem mais. Tal constatação pode permitir decisões mais estratégicas sobre quantidade de produtos que pode possuir em seu estoque. Com isso a empresa pode otimizar seu controle sobre o estoque de produtos.

Além disso o fato do proponente do projeto conhecer a empresa auxilia bem o negócio e ter um bom entrosamento com a equipe para fazer o mapeamento de requisitos e possibilitar a iteração necessária para validar a especificação do software.

1.4 Metodologia

A presente pesquisa foi realizada na loja Auto Mola JM, uma oficina mecânica e revenda de peças, com os funcionários da mesma. Foi feita uma entrevista para o levantamento de requisitos. Para o desenvolvimento do projeto foi utilizada a modelagem UML, com os diagramas de casos de uso e de classes.

A linguagem de programação utilizada foi o Java, com a técnica de programação orientada a objeto.

O método de desenvolvimento de software é o ágil, porque metodologias ágeis agregam valor para equipe, para a organização e clientes. Metodologias ágeis adotam práticas de desenvolvimento iterativo e incremental, ter foco em gerar valor agregado para o cliente e entregar produtos de alta qualidade.

O sistema foi desenvolvido com a participação do cliente, para o cliente ir validando-o de acordo como as funcionalidades iriam ficando prontas.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo pretende demonstrar os principais conteúdos e conceitos necessários para a realização dos objetivos almejados neste trabalho. Inicialmente será abordada a metodologia de desenvolvimento de software, em seguida o processo de software com: Especificação de Software, Engenharia e Análise de Requisitos, Projeto e Implementação de Software, Linguagem de Modelagem Unificada, Metodologia Ágil, sobre o Scrum, Linguagem Java, Netbeans IDE e enfim a conclusão sobre a pesquisa bibliográfica realizada.

2.1 Metodologia de Desenvolvimento de Software

Uma metodologia completa constitui-se de uma abordagem organizada para atingir um objetivo, por meio de passos preestabelecidos. É um roteiro, um processo dinâmico e iterativo para desenvolvimento estruturado de projetos, sistemas ou software, visando a qualidade, produtividade e efetividade de projetos (REZENDE, 1997).

Metodologia de software não é apenas uma técnica, porque pode utilizar qualquer outra técnica de desenvolvimento de projeto, sistema ou software. Alguns exemplos de técnicas são: Análise Estruturada, Análise de Pontos por Função, Análise Orientada a Objetos, Unified Modeling Language (UML) entre outras.

A metodologia deve auxiliar o desenvolvimento de projetos, consiste em várias, sistemas ou software, de modo que os mesmos atendam de maneira adequada às necessidades do cliente ou usuário, com os recursos disponíveis e dentro de um prazo ideal definido em conjunto com os envolvidos. Não deve limitar a criatividade profissional, mas deve ser um instrumento que determine um planejamento metódico, que harmonize e coordene as áreas envolvidas. O que limita a criatividade não é a metodologia, mas os requisitos de qualidade, produtividade e efetividade de um projeto (REZENDE, 2005).

2.1.1 Premissas de desenvolvimento de sistemas

As premissas da metodologia são a modularidade e sua própria existência. A modularidade não tolera o desenvolvimento de projetos, sistemas ou software sem

metodologia. A segunda premissa retrata que sempre um sistema ou software deve ser desenvolvido com uma metodologia, mesmo que ainda não esteja fortemente sedimentada (REZENDE, 2005).

2.1.2 Fases da metodologia de desenvolvimento de sistemas

As fases de uma metodologia também podem ser chamadas de ciclo de vida de sistema ou processos de software. Embora essas fases sejam apresentadas em sequência, o processo de desenvolvimento de Sistemas de Informação deve ser dinâmico e interativo. Essa interatividade permite que a equipe desenvolvedora retorne às fases anteriores ou desmembre o sistema em módulos ou protótipos, possibilitando a continuação da construção de uma parte do Sistema de Informação. Todavia, sempre devem ser respeitados os pontos avaliação da qualidade e aprovação, que requerem revisão da qualidade do projeto e validação formal dos envolvidos.

As fases para o desenvolvimento de projetos, sistemas ou software são: estudo preliminar, ou anteprojeto, ou estudo inicial, ou primeira visão; análise do sistema atual, ou reconhecimento do ambiente; projeto lógico, ou especificação do projeto, ou design; projeto físico, ou execução, ou implementação do projeto, ou programação; projeto de implantação, ou projeto de disponibilização e uso (REZENDE, 2005).

2.1.2.1 Definições das fases de desenvolvimento de sistemas

- Estudo preliminar: É a visão global do projeto, com a primeira definição dos requisitos funcionais, objetivos, abrangências, integrações, limitações, impactos e áreas envolvidas.
- Análise do Sistema: É a visão do atual sistema, relatando requisitos funcionais atuais.
- Projeto Lógico: É a confecção de propostas de soluções, definição dos requisitos funcionais, desenho da lógica do projeto. Definição de o que o projeto fará.
- Projeto Físico: Execução, confecção de programas e seus testes, layout das entradas e saídas.

- Projeto de implantação: Disponibilidade, execução do planejamento de implantação, treinamento e capacitação do cliente ou usuário.

Para o desenvolvimento do software foi utilizado uma metodologia, onde através dela foi feita a análise após a realização da entrevista com o proprietário, a divisão do projeto em fases como estudo preliminar, Linguagem UML com diagramas de classe e de caso de uso e a implementação do projeto disposto para a equipe.

2.1.3 Metodologia Ágil

Em 2001, Kent Beck e 16 outros notáveis desenvolvedores, produtores e consultores de software [BEC01a] (conhecidos como a “Aliança Ágil”) assinaram o “Manifesto para o Desenvolvimento Ágil de Software”. Um manifesto é normalmente associado com um movimento político emergente – um movimento que ataque a velha guarda e sugira mudanças revolucionárias. De algum modo, isso é exatamente o que o desenvolvimento ágil é. Embora as ideias subjacentes que guiam o desenvolvimento ágil tenham estado conosco há muitos anos, somente durante a década de 190 é que essas ideias foram cristalizadas em um “movimento”. Em essência, os métodos ágeis foram desenvolvidos em um esforço para vencer as fraquezas percebidas e reais da engenharia de software convencional. O desenvolvimento ágil pode fornecer importantes benefícios, mas ele não é aplicável a todos os projetos, produtos, pessoas e situações. Ele também não é contrário à sólida prática de engenharia de software e pode ser aplicado como uma filosofia prevaiente a todo o trabalho de software (PRESSMAN, 2006).

A engenharia de software ágil combina uma filosofia e um conjunto de diretrizes de desenvolvimento. A filosofia encoraja a satisfação do cliente e a entrega incremental do software logo de início; equipes de projeto pequenas, altamente motivadas, métodos informais; produtos de trabalho de engenharia de software mínimos e simplicidade global do desenvolvimento. As diretrizes de desenvolvimento enfatizam a entrega em contraposição à análise e ao projeto (apesar dessas atividades não serem desencorajadas) e a comunicação ativa e contínua entre desenvolvedores e clientes (PRESSMAN, 2006).

2.2 Processos de Software

Na engenharia de software, processos podem ser definidos para atividades como Especificação de software, Projeto e Implementação de software. E para cada atividade dessa, podem-se definir subprocessos, como determinação de requisitos funcionais, análise de sistema, diagramação, programação, testes e implantação.

Sommerville (2003) define um processo de software como um conjunto de atividades e resultados associados que levam à produção de um produto de software. Os processos de software são complexos e, como todos os processos intelectuais, dependem de julgamento humano. Não há um processo ideal, e diferentes organizações desenvolveram abordagem inteiramente diferente. Até dentro da mesma empresa pode haver muitos processos diferentes utilizados para o desenvolvimento de software.

2.2.1 Especificação de software

Segundo Sommerville (2003) a especificação de software destina-se a estabelecer quais funções são requeridas pelo sistema e as restrições sobre operação e o desenvolvimento do sistema. Essa etapa do processo de software geralmente é chamada de engenharia de requisitos que se encontrados erros produzem problemas posteriores no projeto e na implementação do sistema.

2.2.1.1 Engenharia e análise de requisitos

Na etapa de especificação no desenvolvimento de um software são coletados requisitos ou informações necessárias para modelagem do sistema. Para poder analisar e entender o problema a ser solucionado e estabelecer com precisão o que o sistema deve fazer.

Com a análise de requisitos serão determinados e analisados requisitos com a finalidade de converter a explicação de alto nível desses requisitos em uma lista precisa que possa ser usada como informação para o restante da fase de análise.

A organização dos requisitos em grupos com diferentes níveis de descrições permite que não apareçam problemas durante o processo de engenharia de requisitos.

A análise de requisito é fundamental para elaborar um sistema ou software que atenda e satisfaça plenamente os desenvolvedores, clientes e usuários. Quando bem definido e relatados evitam muitos problemas futuros e um deles é a alta manutenção de sistemas e software (REZENDE, 2005).

2.2.1.1.1 Requisitos funcionais

Os requisitos de um software, também chamados de requerimento de software ou de requisitos funcionais de um sistema, devem ser elaborados no início de um projeto de sistema ou software. (REZENDE, 2005).

Os requisitos funcionais são declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações. Em alguns casos os requisitos funcionais podem também explicitamente declarar o que o sistema não deve fazer. (SOMMERVILLE, 2003).

2.2.1.1.2 Requisitos não funcionais

Os requisitos não funcionais, como o nome sugere, são aqueles que não dizem respeito diretamente às funções específicas fornecidas pelo sistema. Eles podem estar relacionados a propriedades de sistema emergentes, como confiabilidade, tempo de resposta e espaço em disco. Como alternativa, eles podem definir restrições para o sistema, como a capacidade dos dispositivos de E/S (entrada/saída) e as representações de dados utilizadas nas interfaces de sistema.

Muitos requisitos não funcionais dizem respeito ao sistema como um todo, e não a características individuais do sistema. Com isso ele se torna na maioria das vezes mais importantes do que os requisitos funcionais individuais. Pois quando falha pode deixar o sistema inútil, enquanto o requisito funcional apenas degrada o sistema. Por exemplo se um sistema de controle falha em cumprir o requisito de desempenho as funções de controle não irá funcionar direito.

2.2.2 Projeto e implementação de software

Sommerville (2003) descreve que a etapa de implementação do

desenvolvimento de software é o processo de conversão de uma especificação de sistema em um sistema executável que envolve processo de projeto e programação de software podendo ser também um aperfeiçoamento da especificação de software quando aliado a uma metodologia de desenvolvimento.

Um projeto de software é uma descrição de estrutura de software a ser implementada, dos dados que são parte do sistema, das interfaces entre componentes do sistema e, algumas vezes, dos algoritmos utilizados. O processo de projeto de software envolve acrescentar forma e detalhes, à medida que o projeto é desenvolvido com retornos constantes a fim de corrigir projetos anteriores. (SOMMERVILLE, 2003).

2.2.3 UML (Linguagem de Modelagem Unificada)

UML é uma linguagem para visualização, especificação, construção e documentação de artefatos de um software orientado a objeto.

Sua vantagem é que ela serve para as quatro atividades: análise, design, implementação e teste.

Ela é composta por itens, relacionamentos e diagramas, sendo eles:

Itens: Estruturais (classe, interface, casos de uso e componentes); Comportamentais (interações, máquinas de estado); Grupos de elementos (pacotes, frameworks e subsistemas) e Anotações (notas).

Relacionamentos: Dependência; Associação; Generalização e Realização. Os relacionamentos são entre itens como classes e casos de uso.

Diagramas: são divididos em 9 diagramas que são: de classes; de objetos; de casos de uso; de sequência; de colaborações; de gráficos de estados; de atividades; de componentes e de implantação.

Neste trabalho foram utilizados os diagramas de classe e de casos de uso.

2.2.3.1 Diagrama de casos de uso

O Diagrama de Casos de Uso tem o objetivo de auxiliar a comunicação entre os analistas e o cliente. Esse diagrama descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário. Podemos ver o diagrama feito neste trabalho no capítulo 6, página 32.

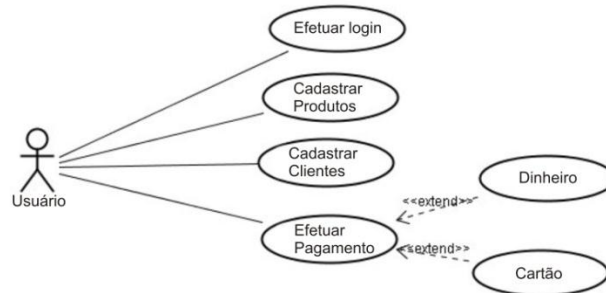


Figura 1: Modelo de Diagrama de casos de uso

O diagrama de caso de uso é representado por atores, casos de uso e relacionamentos. Os relacionamentos podem ser associações entre atores e casos de uso; generalizações entre os atores e generalizações, extends e includes.

2.2.3.2 Diagrama de classe

A funcionalidade externa de um sistema orientado a objetos é fornecida por meio de colaboração entre objetos que colaboram uns com os outros para produzir os resultados visíveis no desenvolvimento do sistema. Essa colaboração permite compreender como o sistema está estruturado internamente (BEZERRA, 2007).

A relação entre esses objetos é representados por um diagrama de classe. No qual são representadas por classes e associações entre as mesmas é um diagrama rico em termos de notação e importante para nível de análise.

2.2.4 Implementação

Implementação é a fase do ciclo de vida de um software (programa computacional, documentação e dados), no contexto de um sistema de informação, que corresponde à elaboração e preparação dos módulos necessários à sua execução. Podemos encontrar parte da implementação do software no capítulo 7.

2.2.4.1 Programação Orientada a Objetos

Orientação a objetos é uma maneira de programar que ajuda na organização e resolve muitos problemas enfrentados pela programação procedural.

A programação orientada a objeto, segundo a maioria dos especialistas,

fundamenta-se em quatro princípios: abstração, encapsulamento, herança e polimorfismo. (MECENAS, 2008).

O paradigma Orientado a Objetos (POO) é o mais avançado no quesito de reusabilidade de código e não resolve tudo aquilo que se propõe a fazer, ferindo, inclusive, seus próprios preceitos e ocasionando no sistema dificuldade de compreensão, manutenção, forte acoplamento e até mesmo redundância de código (WINCK & JUNIOR, 2006).

A programação orientada a objetos se ocupa de realizar um projeto de software utilizando uma linguagem de programação orientada a objetos na qual aceita implementação direta de objetos e fornece recursos para definir as classes de objetos. Os sistemas orientados a objetos devem ser de fácil manutenção, uma vez que os objetos são independentes. (SOMMERVILLE, 2003)

3 SCRUM

Scrum é um processo Ágil ou ainda um framework para gerenciamento de projetos Ágeis. É um processo de gerência de projetos, certamente não uma metodologia, pois isto seria pesado demais (SCHWABER, 2004).

O Scrum não é um processo previsível, ele não define o que fazer em toda circunstância. Ele é usado em trabalhos complexos nos quais não é possível prever tudo o que irá ocorrer e oferece um framework e um conjunto de práticas que torna tudo visível. Isso permite aos praticantes do Scrum saber exatamente o que está acontecendo ao longo do projeto e fazer os devidos ajustes para manter o projeto se movendo ao longo do tempo visando alcançar os seus objetivos (SCHWABER, 2004).

O Scrum possui componentes para organizar o desenvolvimento de softwares e interação entre o cliente e a equipe de desenvolvimento, são eles: Sprints, Product Backlog e Sprint Backlog.

Sprints: O Sprint é o ciclo de desenvolvimento do Scrum, caracterizado por ter um curto período onde a equipe foca na entrega de uma meta específica.

Product Backlog ou Backlog do Produto: Requisitos de produtos organizados em uma lista de itens.

Sprint Backlog: O Sprint Backlog é gerado a partir das histórias retiradas do Product Backlog e que serão implementadas durante o Sprint.

3.1 Ciclo do Scrum

O Scrum tem o progresso de desenvolvimento baseado em iterações com duração que pode variar de duas a seis semanas, chamadas de Sprints. A primeira etapa dentro do Sprint é uma reunião de planejamento (Sprint Planning), onde o time (Scrum Team), em conjunto com o cliente (Product Owner) define o que será implementado na iteração, sendo responsabilidade do cliente realizar a priorização do trabalho a ser feito. (VARASCHIM, 2009)

Outra etapa é a de execução, onde o time detalha as tarefas necessárias para implementar o que foi solicitado pelo cliente e depois inicia a execução das mesmas. Durante o Sprint o time realiza reuniões diárias (Daily Meeting) com 15 minutos para averiguar o acompanhamento do progresso do desenvolvimento, usando para isto o

Burn Down Chart, que é um gráfico usado para o acompanhamento das tarefas a realizar, em andamento e já realizadas.

Ao final do Sprint é realizada uma reunião para a validação da entrega (Sprint Review), onde o cliente e quem mais tiver interesse no produto pode verificar se o objetivo do Sprint foi atingido. Logo após, é realizada apenas pelo time uma reunião (Sprint Retrospective) onde o Sprint é avaliado sob a perspectiva de processo, time ou produto, quais foram os acertos e os erros com o objetivo de melhorar o processo de trabalho.

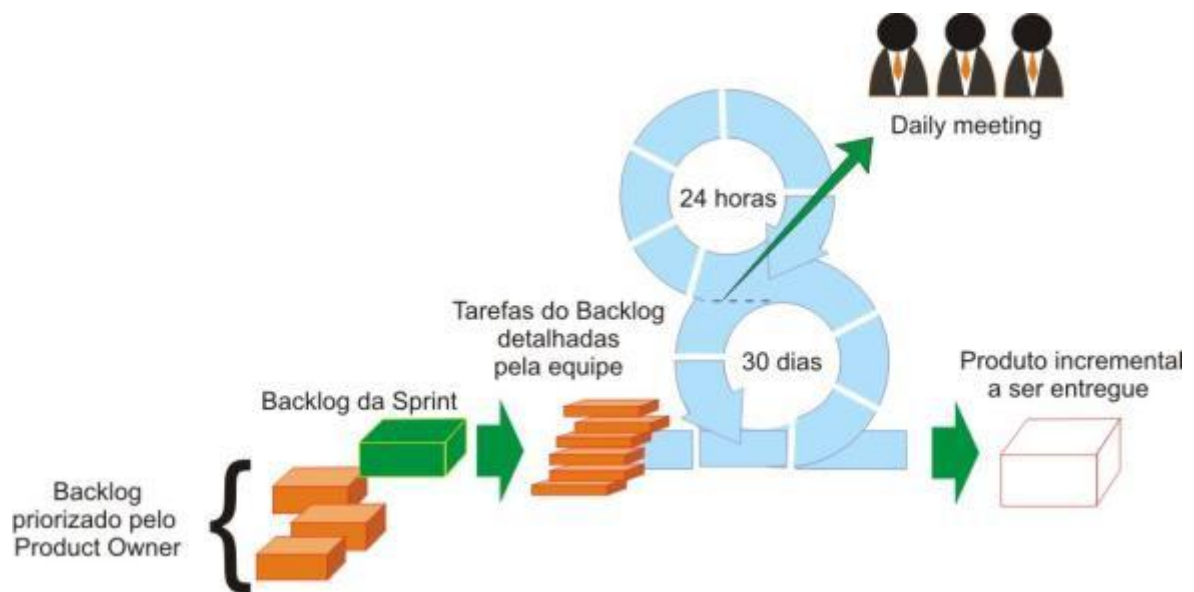


Figura 2: Ciclo do Scrum

3.2 Papéis do Scrum

Scrum Master: é o gerente do projeto, sua função é ser o facilitador do processo e ajudar as pessoas a resolver os problemas.

Scrum Team: é o time do projeto com no máximo 10 a 15 participantes e deve ser multidisciplinar e auto organizado.

Product Owner: é o cliente de um time Scrum, ele é responsável pelo retorno do investimento de um produto.

3.3 Etapas do Sprint

O Sprint é definido como um ciclo de desenvolvimento curto em que o time foca em atingir o objetivo proposto pelo Product Owner. Durante este ciclo o time

deve ter total autoridade para seu gerenciamento, não devendo sofrer interferências externas do cliente ou de outras pessoas.

Sprint Planning: é a primeira atividade dentro do Sprint, uma reunião, durante essa reunião é feita uma seleção das histórias que serão implementadas durante aquele ciclo.

Daily Scrum: é uma reunião rápida que deve ter o tempo fechado, realizada todo dia, reuniões em pé no qual os membros do time explicam entre si o que foi feito desde a última Daily Scrum.

Sprint Review: acontece no final de cada Sprint, durante esta reunião o time mostra o resultado do seu trabalho para o Product Owner e para outros clientes.

Sprint Retrospective: é a reunião do time realizada sempre ao final de cada Sprint e deve ser utilizada para a correção dos problemas detectados pelo time no processo de desenvolvimento.

O scrum é um modelo de implementação simples, faz parte poucas especificações e artefatos, o desafio é integrá-lo as necessidades do cliente, do time e da empresa. É importante saber que a prática do Scrum cria equipes motivadas, e com o tempo criam laços com o produto tendendo cada vez mais a entregar qualidade e eficiência.

Após ver e analisar todas essas características foi escolhido o scrum como um método de gerenciamento do projeto, pois o mesmo nos permite fazer isso. Criando assim as sprints para ir mostrando as funcionalidades do sistema ao cliente conforme elas iam sendo feitas. Essas sprints tinham em média duas semanas cada, pois assim havia tempo hábil para a criação das funcionalidades.

Ao final das sprints foram realizadas reuniões com o cliente para ir verificando se estava de acordo o produto do projeto. Também foi realizada uma última reunião após o término do processo de desenvolvimento para apresentação do produto final e aprovação do cliente.

4 LINGUAGEM JAVA

Java é uma linguagem derivada a partir da linguagem de programação C. C foi criada, influenciada e testada por programadores profissionais. A linguagem possui poucas restrições, códigos estruturados e um conjunto de palavras-chaves limitado.

Essas características mostram que C trabalha no mais baixo nível, já outras linguagens trabalham no nível mais alto.

Java é uma das filhas de C. Criada por James Gosling, da Sun Microsystems, com o nome de “Oak”, a linguagem deveria incorporar os benefícios da programação orientada a objetos sem, no entender do criador, a desnecessária complexidade de C (MECENAS 2008).

Segundo Mecnas (2008) a linguagem java é dividida nos seguintes componentes:

- **Applets:** applet é um componente de software que executa uma função limitada em outro ambiente de programa, como um navegador da Web. Os applets Java fornecem recursos interativos em um navegador da Web por meio do Java Virtual Machine (JVM).
- **JVM (Java Virtual Machine):** O mecanismo responsável pela interpretação do código intermediário Java. A Máquina Virtual Java, embora possa parecer estranho, não conhece a linguagem Java. Ela somente reconhece o formato de arquivo .class que contém os bytecodes, os quais serão traduzidos para o código binário da plataforma hospedeira.
- **JSDK (Java Software Development Kit) ou JDK (Java Development Kit):** O conjunto de ferramentas de desenvolvimento oferecido pela Sun e outros fabricantes, que inclui, como estrutura mínima, um compilador e um interpretador.
- **JRE (Java Runtime Environment):** O ambiente de execução Java existente nas máquinas dos usuários finais, que apenas executam aplicações Java. O ambiente inclui, portanto, a Máquina Virtual Java e as classes que formam a plataforma Java.
- **JDBC (Java Database Connectivity):** Arquitetura baseada em interface e especificações que permite acesso multiplataforma a bancos de dados. O

acesso aos dados baseia-se em drivers JDBC, que são implementados por conjuntos de classes que dominam a arquitetura e os comandos dos bancos de dados.

- **Servlets:** Extensões do padrão CGI, escritas em puro Java, usadas para a geração de conteúdo dinâmico para a Web.
- **JSP (JavaServer Pages):** Conjunto de APIs que permitem a criação de páginas web dinâmicas, com acesso a dados e interação com o usuário. Atualmente, há uma forte tendência a se utilizar JSP para a parte visual das aplicações e Servlets para a lógica de validação e controle.

Há muitos aplicativos e sites que funcionam somente com o Java instalado, e muitos outros aplicativos e sites são desenvolvidos e disponibilizados com o suporte dessa tecnologia todos os dias. O Java é rápido, seguro e confiável. A tecnologia Java está em todo lugar, pode ser encontrada em laptops, datacenters, consoles de jogo, supercomputadores científicos, telefones celulares e até na Internet.

Por isso Java é uma tecnologia muito utilizada no mundo, e foi por essas características e qualidades que foi escolhida para o processo de desenvolvimento de software do presente trabalho.

5 NETBEANS IDE

NetBeans IDE (Integrated Development Environment) é uma ferramenta de desenvolvimento modular para uma ampla gama de tecnologias de desenvolvimento de aplicações. O IDE de base inclui um editor avançado, bem como ferramentas para controle de versão e colaboração de desenvolvedores. (Netbeans.org).

5.1 Suporte para as tecnologias mais recentes de Java

NetBeans IDE oferece suporte abrangente para as mais recentes tecnologias Java e as últimas melhorias de Java antes de outros IDEs. Esse IDE fornece suporte para JDK 7, o JDK é um ambiente de desenvolvimento para construir aplicações, e componentes usando a linguagem de programação Java uma máquina virtual, Java EE Enterprise Edition é o padrão em software empresarial voltada para a comunidade, e Java FX é o próximo passo na evolução do Java como uma plataforma rich client. Ele é projetado para fornecer uma plataforma leve acelerada por hardware Java UI para aplicações de negócios da empresa. (Oracle.com).

O IDE fornece modelos de projeto e exemplos de projetos que ajudam a criar aplicativos Java, aplicações Java EE, aplicações HTML5, aplicação PHP e aplicativos C/C++ além de outras linguagens disponíveis no IDE.

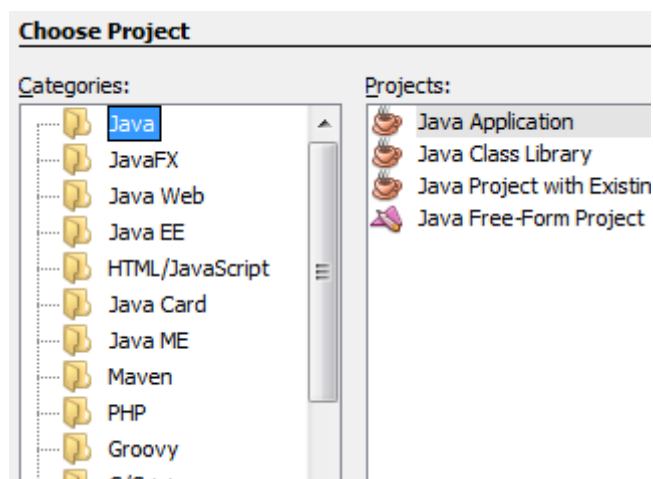


Figura 3: Escolha do projeto

O IDE suporta várias linguagens divididas por categorias, entre elas estão: Java, Java FX, Java Web, Java EE, HTML/Java Script, PHP, Groovy e outras. Após selecionar a categoria temos ainda opções de projetos disponíveis para cada uma dessas categorias, como mostra a figura 2.

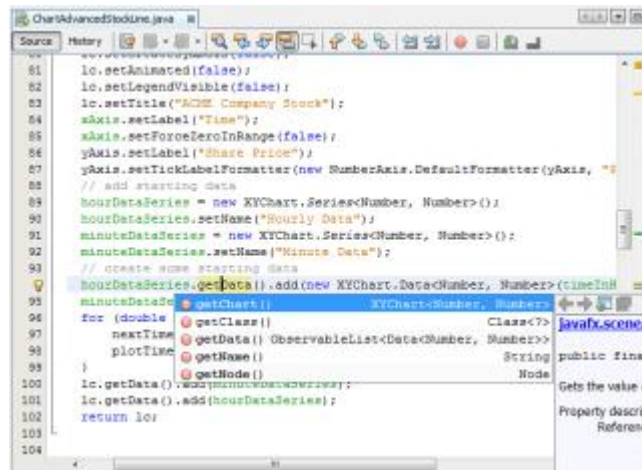


Figura 4: Edição de código

A figura 3 mostra um exemplo da edição de um código dentro do NetBeans IDE.

NetBeans IDE oferece aplicações de esqueleto em forma de modelos de projeto para todas as tecnologias que ele suporta. Além disso, proporciona um conjunto de aplicações de exemplo.

5.2 Bancos de Dados e Serviços

A janela Serviços dá acesso a muitos recursos auxiliares, tais como bancos de dados, servidores, serviços web e rastreadores de problemas.

Você pode iniciar e parar os bancos de dados e servidores diretamente no IDE. Ao trabalhar com bancos de dados, você pode adicionar, remover e modificar os seus dados no IDE. Quando você implantou um aplicativo para um servidor, você pode gerenciar seus recursos mobilizados, porque eles são exibidos no nó Servidores.

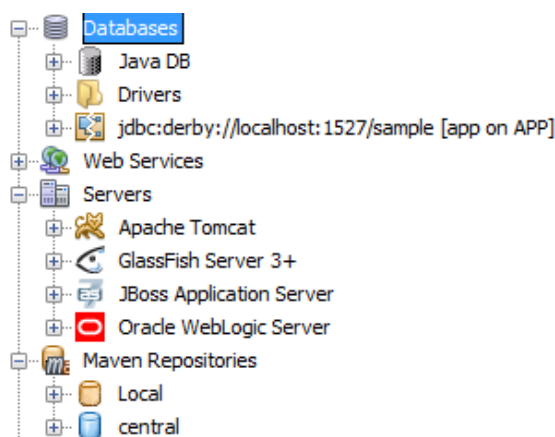


Figura 5: Serviços Básicos

5.3 Desenvolvimento de Interface

Projeto de interface gráfica pode ser desenvolvido dentro do próprio IDE, com isso facilita a integração da interface e do código fonte do projeto em desenvolvimento, vantagem essa muito utilizada pelos usuários dessa ferramenta facilitando assim a visualização do usuário final de um software.

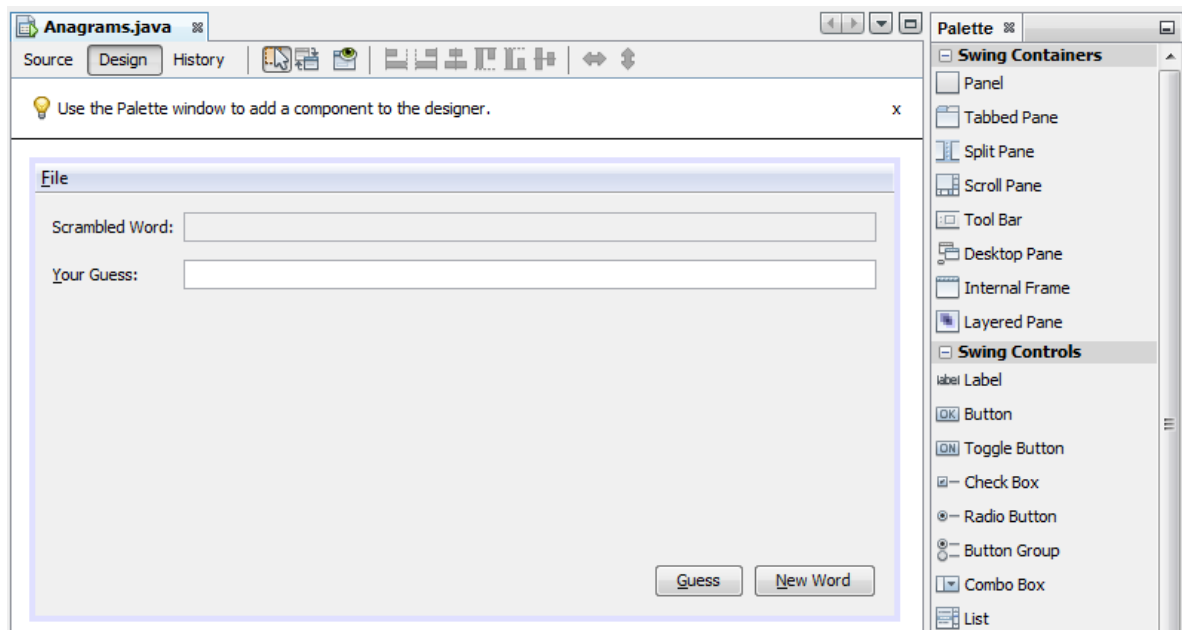


Figura 6: Exemplo de interface gráfica

Criação de uma interface gráfica de um software contendo dois botões, um título e duas caixas de texto mostrados na figura 6.

5.4 Suporte

O NetBeans IDE oferece suporte para desenvolvedores C/C++ e PHP, fornecendo editores e ferramentas para os seus quadros e tecnologias relacionadas. Além disso, o IDE tem editores e ferramentas para XML, HTML, PHP, Groovy, Javadoc, Java Script e JSP.

NetBeans IDE pode ser instalado em todos os sistemas operacionais que suportam Java, a partir do Windows para o Linux para sistemas Mac OS. Porque o próprio NetBeans IDE é escrito em Java.

5.5 Projeto Exibição

A janela Projetos é o principal ponto de entrada para as fontes do projeto. Ele mostra uma visão lógica do conteúdo do projeto.

Além da janela Projetos, o IDE fornece a janela Arquivos, para que você possa ver todos os arquivos que pertencem a um projeto, e a janela de Favoritos, onde você pode adicionar pastas e arquivos de modo que eles podem ser pesquisados dentro do IDE.

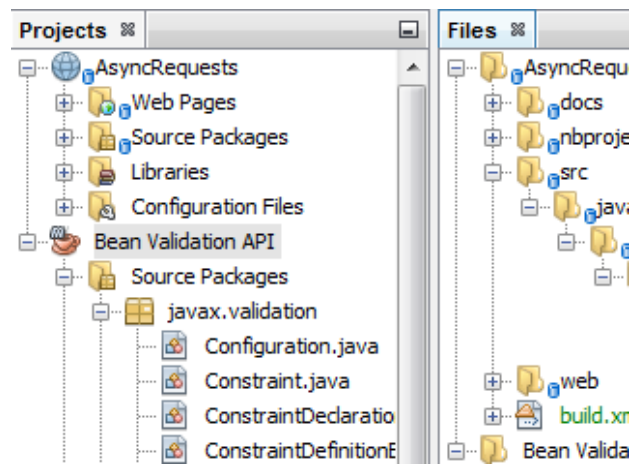


Figura 7: Gerenciamento de Projeto

Figura ilustrando uma janela detalhando de um projeto, juntamente com uma janela com os arquivos compostos nesse projeto.

6 DOCUMENTAÇÃO DO SISTEMA CE (Controle do Estoque)

A documentação de um sistema tem suma importância por proporcionar um grande conhecimento sobre as funcionalidades. Com essas informações um desenvolvedor consegue pegar os dados para o desenvolvimento do software.

A seguir serão mostrados os documentos do sistema como: os diagramas de classe e de casos de uso com sua descrição.

6.1 Diagrama de Classe

A funcionalidade externa de um sistema orientado a objetos é fornecida por meio de colaboração entre objetos que colaboram uns com os outros para produzir os resultados visíveis no desenvolvimento do sistema. Essa colaboração permite compreender como o sistema está estruturado internamente (BEZERRA, 2007).

Na figura 8 é mostrado o diagrama de classes do sistema dividido em atributos e métodos das classes e contendo as associações entre si.

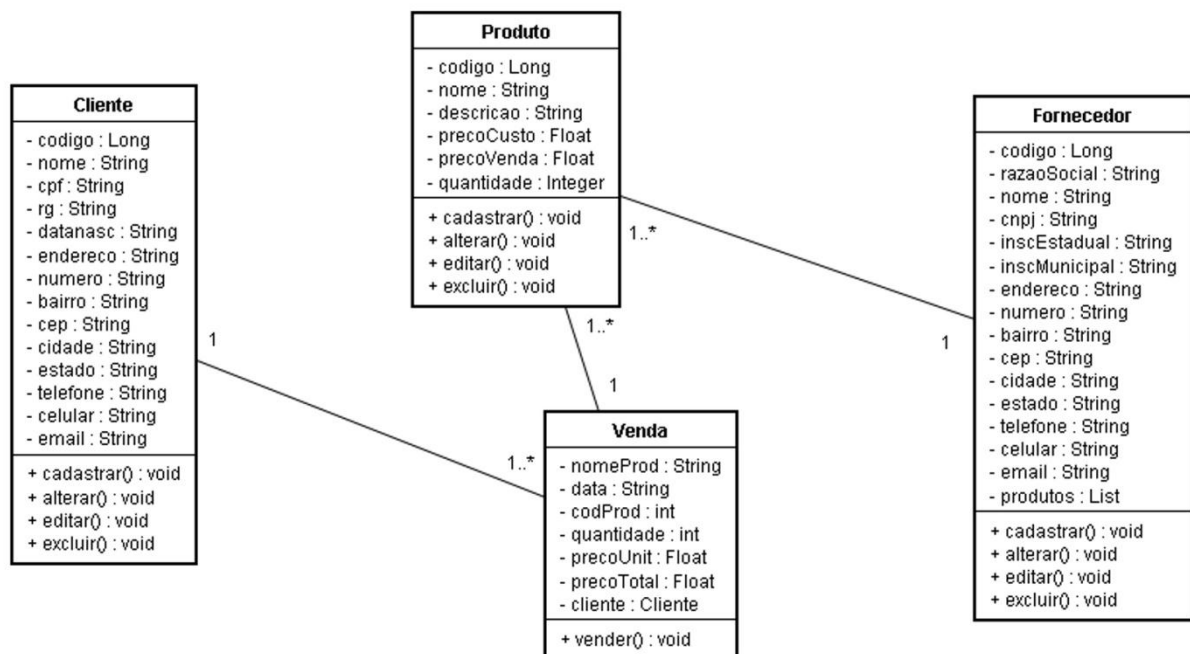


Figura 8: Diagrama de Classe do Sistema CE

Na figura 8 mostra como as classes fazem a interação uma com a outra, a entidade produto pode ter um fornecedor, mais o fornecedor pode ter um ou vários produtos, um produto só pode ter uma venda, já a venda pode ter um ou vários

produtos e o cliente pode fazer um ou várias compradas, mas a venda só pode ser realizada para um cliente.

6.2 Diagrama de Casos de Uso

Um caso de uso é a especificação de uma sequência completa de interações entre um sistema e um ou mais agentes externos a esse sistema (BEZERRA, 2007). As características do caso de uso são: sempre iniciados por um ator; sempre retornam um resultado ao autor; especifica uma funcionalidade completa.



Figura 9: Casos de Uso do Sistema CE

6.2.1 Descrição dos casos de uso

[UC 01]: Logar no Sistema

Descrição: O usuário deverá entrar com seus dados: login e senha. O Sistema deverá permitir acesso ao conteúdo do software somente se os dados estiverem corretos.

Atores envolvidos: Usuário do sistema.

Pré-condição O usuário já devesse possuir seu cadastro no sistema.

Cenário Principal de Sucesso:

1. O caso de uso é iniciado com usuário fornecendo o login e senha.
2. O usuário deverá entrar com seus dados.
3. O sistema busca no banco de dados se os dados estão corretos.
4. O sistema é iniciado, liberando as funcionalidades.

Cenário Secundário de Falha:

- 4.1 O sistema retorna a mensagem de erro login inválido.

[UC 02] Cadastrar Produto

Descrição: O usuário deverá fazer entrada dos dados referente a cada produto.

Atores envolvidos: Usuário do sistema.

Pré-condição: O usuário deverá estar logado no sistema. E nome e código do produto ainda não cadastrado.

Pós-Condições: Retorno mensagem de produto cadastrado.

Cenário Principal de Sucesso:

1. O usuário deve ir ao cadastro do produto.
2. O usuário deverá informar os dados do produto.
3. O usuário submete os dados necessários para armazenamento no banco de dados.
4. O sistema validará os dados e retorna mensagem “Cadastro com sucesso”.

Cenário Secundário de Falha:

- 4.1. O sistema aborta a validação dos dados e retorna mensagem de erro, “Cadastro não realizado”.

[UC 03] Alterar Produto

Descrição: O usuário deverá fazer entrada dos dados referente à busca do produto, código ou nome do produto.

Atores envolvidos: Usuário do sistema.

Pré-condição: O usuário deverá estar logado no sistema. E o usuário deve fornecer um dos dados para encontrar o produto.

Pós-Condições: Retorno mensagem de alteração, realizada com sucesso.

Cenário Principal de Sucesso:

1. O usuário deve ir ao relatório do produto.
2. O usuário deverá informar os dados do produto referente à busca.
3. O usuário terá os dados do produto, podendo alterá-los conforme sua necessidade, exceto código identificador do produto.
4. O usuário submete a alteração e o sistema validará os dados.
5. O sistema retornará mensagem de sucesso, "Produto Atualizado".

Cenário Secundário de Falha:

- 4.1. O sistema aborta a validação dos dados e retorna mensagem de erro, e mostra "Atualização Não Realizada", "Faltando Dados".

[UC 04] Excluir Produto

Descrição: O usuário deverá fazer entrada dos dados referente a busca código identificador ou nome do produto.

Atores envolvidos: Usuário do sistema.

Pré-condição: O usuário deverá estar logado no sistema. E o usuário deve fornecer um dos dados para encontrar o produto.

Pós-Condições: Retorno mensagem de exclusão realizada com sucesso.

Cenário Principal de Sucesso:

1. O usuário deve ir ao relatório do produto.
2. O usuário deverá informar os dados do produto referente à busca.
3. O sistema retornará os dados do cadastro do produto. O usuário terá a visualização dos dados do produto no sistema.
4. O usuário submete a ação para excluir o produto.
5. O sistema retornará mensagem de sucesso, "Produto Excluído".

[UC 05] Cadastrar Cliente

Descrição: O usuário deverá fazer entrada dos dados referente a cada cliente.

Atores envolvidos: Usuário do sistema.

Pré-condição: O usuário deverá estar logado no sistema. E CPF e código do cliente ainda não cadastrado.

Pós-Condições: Retorno mensagem de cliente cadastrado.

Cenário Principal de Sucesso:

1. O usuário deve ir ao cadastro de cliente.
2. O usuário deverá informar os dados do cliente.
3. O usuário submete os dados necessários para armazenamento no banco de dados.
4. O sistema validará os dados e retorna mensagem, “Cliente Cadastrado!”, “SUCESSO”.

Cenário Secundário de Falha:

- 4.1. O sistema aborta a validação dos dados e retorna mensagem de erro, “CPF não informado!” ou “Falta Dados”.

[UC 06] Alterar Cliente

Descrição: O usuário deverá fazer entrada dos dados referente à busca do produto, como código, CPF ou nome do cliente.

Atores envolvidos: Usuário do sistema.

Pré-condição: O usuário deverá estar logado no sistema. E o usuário deve fornecer um dos dados para encontrar o cliente.

Pós-Condições: Retorno mensagem de alteração realizada com sucesso.

Cenário Principal de Sucesso:

1. O usuário deve ir ao cadastro do cliente.
2. O usuário deverá informar os dados do cliente referente à busca.
3. O usuário terá os dados do cliente, podendo alterá-los conforme sua necessidade, exceto código identificador do cliente.
4. O usuário submete a alteração e o sistema validará os dados.
5. O sistema retornará mensagem, “Cliente Atualizado”.

Cenário Secundário de Falha:

- 4.1. O sistema aborta a validação dos dados e retorna mensagem de erro, e mostra “Dados não atualizados”, “Tente Novamente”.

[UC 07] Excluir Cliente

Descrição: O usuário deverá fazer entrada dos dados referente a busca como seu código, CPF ou nome do cliente.

Atores envolvidos: Usuário do sistema.

Pré-condição: O usuário deverá estar logado no sistema. E o usuário deve fornecer um dos dados para encontrar o cliente.

Pós-Condições: Retorno mensagem de exclusão realizada com sucesso.

Cenário Principal de Sucesso:

1. O usuário deve ir ao cadastro do cliente.
2. O usuário deverá informar os dados do cliente referente à busca.
3. O sistema retornará os dados do cadastro do cliente. O usuário terá a visualização dos dados do cliente no sistema.
4. O usuário submete a ação para excluir o cliente.
5. O sistema retornará mensagem “Cliente Excluído”.

[UC 08] Cadastrar Fornecedor

Descrição: O usuário deverá fazer entrada dos dados referente a cada fornecedor.

Atores envolvidos: Usuário do sistema.

Pré-condição: O usuário deverá estar logado no sistema. E CNPJ, Razão Social e código ainda não cadastrado.

Pós-Condições: Retorno mensagem de fornecedor cadastrado.

Cenário Principal de Sucesso:

1. O usuário deve ir ao cadastro de fornecedor.
2. O usuário deverá informar os dados do fornecedor.
3. O usuário submete os dados necessários para armazenamento no banco de dados.
4. O sistema validará os dados e retorna mensagem, “Fornecedor Cadastrado!”, “SUCESSO”.

Cenário Secundário de Falha:

- 4.1. O sistema aborta a validação dos dados e retorna mensagem de erro, “CNPJ não informado!” ou “Falta Dados”.

[UC 09] Alterar Fornecedor

Descrição: O usuário deverá fazer entrada dos dados referente à busca do fornecedor, código, CNPJ ou razão social do fornecedor.

Atores envolvidos: Usuário do sistema.

Pré-condição: O usuário deverá estar logado no sistema. E o usuário deve fornecer um dos dados para encontrar o fornecedor.

Pós-Condições: Retorno mensagem de alteração realizada com sucesso.

Cenário Principal de Sucesso:

1. O usuário deve ir ao cadastro de fornecedor.
2. O usuário deverá informar os dados do fornecedor referente à busca.
3. O usuário terá os dados do fornecedor, podendo alterá-los conforme sua necessidade, exceto código identificador do fornecedor.
4. O usuário submete a alteração e o sistema validará os dados.
5. O sistema retornará mensagem, “Cliente Atualizado”.

Cenário Secundário de Falha:

- 4.1. O sistema aborta a validação dos dados e retorna mensagem de erro, e mostra “Dados não atualizados”, “Tente Novamente”.

[UC 10] Excluir Fornecedor

Descrição: O usuário deverá fazer entrada dos dados referente a busca com código, CNPJ ou razão social do fornecedor.

Atores envolvidos: Usuário do sistema.

Pré-condição: O usuário deverá estar logado no sistema. E o usuário deve fornecer um dos dados para encontrar o fornecedor.

Pós-Condições: Retorno mensagem de exclusão realizada com sucesso.

Cenário Principal de Sucesso:

1. O usuário deve ir ao cadastro de fornecedor.
2. O usuário deverá informar os dados do fornecedor referente à busca.
3. O sistema retornará os dados do cadastro do fornecedor. O usuário terá a visualização dos dados do fornecedor no sistema.
4. O usuário submete a ação para excluir o fornecedor.
5. O sistema retornará mensagem “Fornecedor Excluído”.

[UC 11] Visualizar Estoque

Descrição: Será solicitada a abertura da tela de estoque.

Atores envolvidos: Usuário do sistema

Pré-condição: O usuário já deveria estar logado no sistema.

Cenário Principal de Sucesso:

1. O caso de uso é iniciado com a solicitação de abertura da tela de estoque.
2. O sistema abrirá a tela contendo a lista com todos os produtos do estoque.

Cenário Secundário de Falha:

2.1 Caso não possua produto no estoque o usuário fecha a tela e volta para tela principal do sistema.

[UC 12] Registrar Movimentação de Produtos

Descrição: O sistema será responsável por controlar o fluxo do estoque da empresa. Com todas as movimentações de entrada e saída dos produtos.

Atores envolvidos: Usuário do sistema.

Pré-condição: O usuário deverá estar logado no sistema. O usuário deverá encontrar o produto a ser movimentado.

Pós-Condições: Retorno mensagem de sucesso.

Cenário Principal de Sucesso:

1. O usuário deve ir até a opção de entrada.
2. Para entrada o usuário deverá informar o nome do produto o sistema busca o produto.
3. Se encontrado o sistema mostra o código do produto e o valor cadastrado no sistema.
4. O usuário informa a data da operação de entrada, o número da nota fiscal, a quantidade do produto e o nome do fornecedor o sistema realiza uma busca.
5. O usuário submete a ação para atualizar o estoque, o sistema registra a movimentação no banco e retorna mensagem de sucesso.

Cenário Secundário de Sucesso:

1. O usuário deve ir até a opção de venda.
2. Para venda o usuário deverá informar o nome do produto o sistema busca o produto.
3. Se encontrado o sistema mostra o código do produto e o valor cadastrado no sistema.
4. O usuário informa a data da operação da venda, a quantidade desejada e o

cliente o sistema realiza um busca.

5. Após o usuário informar a quantidade o sistema mostrar o valor total da venda do produto.

6. O usuário submete a ação para atualizar o estoque, o sistema registra a movimentação no banco e retorna mensagem de sucesso.

Cenário Primário de Falha:

3.1 Caso o produto não seja encontrado, o sistema retornará uma mensagem “Produto Não Encontrado” e retorna para o passo 2.

4.1 Caso o fornecedor não esteja cadastrado, o sistema informa ao usuário uma mensagem “Fornecedor Não Cadastrado”, e retorna para o passo 4.

Cenário Secundário de Falha:

3.1 Caso o produto não seja encontrado, o sistema retornará uma mensagem “Produto Não Encontrado” e retorna para o passo 2.

4.1 Caso o cliente não esteja cadastrado, o sistema informa ao usuário uma mensagem “Cliente Não Cadastrado”, e retorna para o passo 4.

7 SISTEMA CE (CONTROLE DO ESTOQUE)

O Sistema CE foi desenvolvido para a empresa Auto Molas JM da cidade de São Raimundo das Mangabeiras – MA, com a finalidade de controlar o estoque dos seus produtos e também ter o cadastro de seus clientes e seus fornecedores. Esse sistema foi desenvolvido na linguagem Java, através da ferramenta NetBeans IDE que tem suporte tanto para Java como para outras linguagens de programação, mas também suporte para banco de dados a exemplo do mysql que foi o utilizado na realização deste trabalho e nos permite a realização de uma conexão e até mesmo a criação do banco através do ide, e tem como paradigma orientado a objeto um dos mais utilizados na linguagem Java, nos próximos parágrafos será apresentado parte do sistema.

O sistema está dividido em:

1. Tela de autenticação com login e senha;
2. Tela principal contendo botões e menus de acesso as outras tela como: cadastro de cliente, cadastro de produtos, cadastro de fornecedores, estoque dos produtos, movimentação dos produtos e sair do sistema;
3. Telas de cadastros tais como cadastro dos produtos, de clientes e fornecedores;
4. Tela de relatório do estoque;
5. Tela de movimentação.

A partir das telas que o usuário fará toda a interação com o sistema.

7.1 Acesso ao sistema

Para se fazer acesso ao sistema o usuário terá que informar um usuário e a senha corretamente para entrar no sistema.

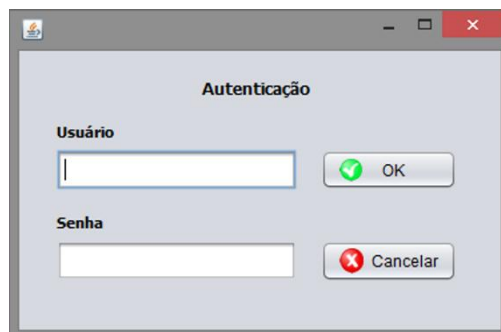


Figura 10: Tela de autenticação do sistema

Foi criado uma tabela no banco de dados chamada credencial onde foi cadastrado um usuário e sua senha, nessa tabela que possui três campos id do usuário, login (nome de usuário) e senha que está criptografada no modo md5¹. Quando executa o software, ele cria a conexão com o banco de dados, o usuário insere seu login e senha, que será verificado se o mesmo está cadastrado no banco, estando cadastrado o usuário e se a senha estiver correta o sistema abre a tela principal, como podemos ver na figura 11.

```

125 private void bokiActionPerformed(java.awt.event.ActionEvent evt) {
126
127     usuario.requestFocus();
128     String senha_campo = new String(senha.getPassword());
129     if (!usuario.getText().equals("") && !senha_campo.equals("")) {
130         try {
131             conecta.executaSQL("select * from credencial where usuario = '" + usuario.getText()
132                 + "' and senha = '" + sen.md5(senha_campo) + "' limit 1");
133             if (conecta.rs.next() == true) {
134                 new TelaPrincipal().setVisible(true);
135             } else {
136                 JOptionPane.showMessageDialog(null, "Erro ao login");
137             }
138         } catch (SQLException ex) {
139             JOptionPane.showMessageDialog(null, "Usuário não encontrado");
140             usuario.requestFocus();
141         }
142     }
143 }

```

Figura 11: Código que compara as informações de autenticação

Na figura 11 podemos observar na linha 129 a verificação dos campos usuário e senha se eles não estão vazios, na linha 131 é realizada a busca no banco de dados onde estando correto entra no sistema, caso não esteja correto o login e a senha é mostrado uma mensagem de erro para o usuário.

7.2 Tela principal do sistema

Nesta tela mostrada na figura 12 o usuário poderá ir para qualquer parte do sistema, pois ela dá acesso a todo o sistema, sempre que o usuário fizer alguma operação retornará para essa tela. É a partir da tela principal que são feitas as chamadas de todas as telas.

¹ MD5 Message-Digest algorithm 5 é um algoritmo de hash de 128 bits unidirecional desenvolvido pela RSA Data Security.

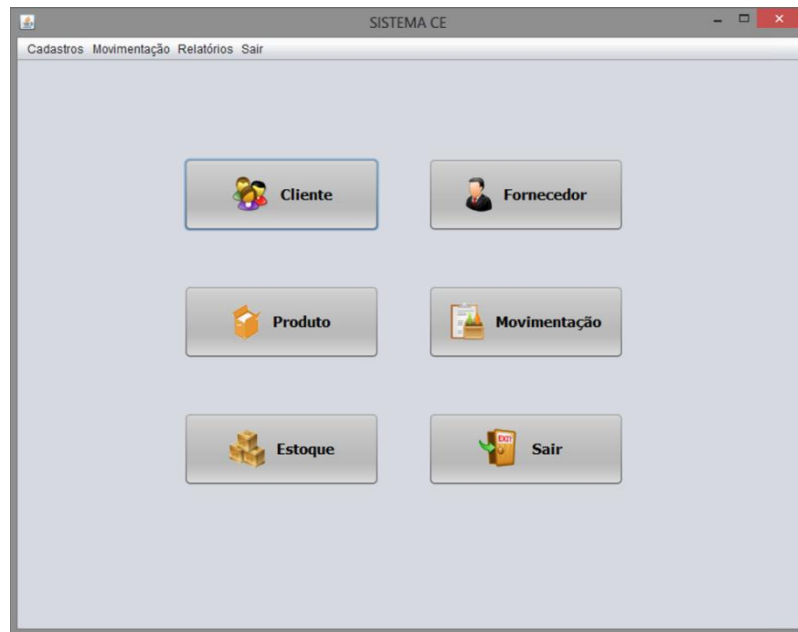


Figura 12: Tela principal do Sistema CE

Figura 13 é uma parte do código onde o sistema faz a instância da classe conexão e chama a função que faz a conexão com o banco de dados, tem um exemplo da ação de um botão no caso o cadastro de cliente, quando for clicado ele torna visível a tela de cadastro de cliente para o usuário e desabilita a tela principal enquanto não está sendo usada e abaixo tem a ação do botão sair que quando clicado ele chama a função de desconectar o banco de dados e fecha o sistema.

```
public class TelaPrincipal extends javax.swing.JFrame {

    Conexao conecta = new Conexao();

    public TelaPrincipal() {
        initComponents();
        this.setLocationRelativeTo(null);
        conecta.conexao();
    }

    @SuppressWarnings("unchecked")
    Generated Code

    private void BotCadClienteActionPerformed(java.awt.event.ActionEvent evt) {
        // desabilitando tela atual para não ser clicável
        this.setEnabled(false);

        //botão Cadastrar Cliente pressionado
        new CadastroCliente(this).setVisible(true);
    }

    private void BotSairActionPerformed(java.awt.event.ActionEvent evt) {
        // botão fecha o programa
        conecta.desconecta();
        System.exit(0);
    }
}
```

Figura 13: Trecho do código da tela principal do sistema

7.3 Cadastro dos produtos

O cadastro dos produtos é onde o usuário informa os dados de todos os produtos tais como o nome, a descrição, o preço de custo, o preço de venda e a quantidade a serem gravados no banco de dados do sistema conforme pode ser visto na figura 14, através do cadastro dos produtos o usuário poderá inserir novos produtos, pesquisar por esses produtos e até mesmo alterá-los.

A imagem mostra uma janela de software intitulada "Cadastro de Produtos". No topo, há uma barra de título com o nome da janela e botões de minimizar, maximizar e fechar. O formulário principal contém campos para "Código" e "Nome" na primeira linha. Abaixo, há um campo grande para "Descrição do Produto". À direita da descrição, há campos para "Preço de Custo" e "Preço de Venda" na mesma linha, e um campo para "Quantidade" na linha seguinte. Um botão "Carrega Produto" com um ícone de pasta está à direita do campo de preço de venda. Na base da janela, há uma barra com cinco botões: "Salvar" (ícone de disquete), "Alterar" (ícone de seta circular), "Excluir" (ícone de X vermelho), "Limpar" (ícone de borracha) e "Fechar" (ícone de X vermelho).

Figura 14: Formulário de cadastro de produtos

7.4 Relatório do estoque

No relatório do estoque conforme figura 15, o usuário saberá quais os produtos possui em seu estoque, os preços tanto de custo quanto de venda e sua quantidade. Nesta parte do sistema o usuário também poderá selecionar um produto no qual ele queira alterar alguma informação do mesmo através do botão alterar, este botão abre a tela de cadastro de produtos com os dados do produto onde o usuário altera a informação a ser alterada e salva-o novamente da mesma forma o usuário poderá excluir algum produto.



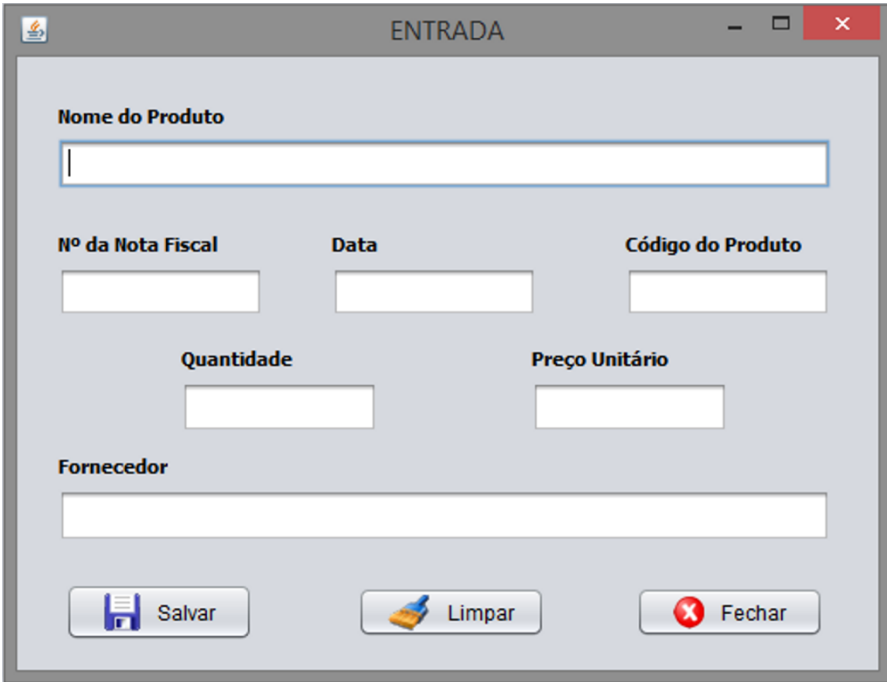
Código	Nome	Preço Unitário	Quantidade
1	Bucha Traseira	145	28
2	Parafuso 3/4	1,5	10
3	Kit Mola Dianteiro	187,75	5
4	Mola Curva	50	7
5	Produto 5	2	10
6	Produto 6	1000	20
7	Produto 7	500	17
8	Produto 8	20	30

Figura 15: Relatório do estoque de produtos

A figura 15 mostra os produtos que estão cadastrados e disponíveis no estoque para a venda dos mesmos.

7.5 Movimentação do estoque

Nesta parte do sistema o usuário fará toda a movimentação dos seus produtos como a entrada de mais produtos com suas informações e a saída dos mesmos, atualizando o estoque.



ENTRADA

Nome do Produto

Nº da Nota Fiscal **Data** **Código do Produto**

Quantidade **Preço Unitário**

Fornecedor




 **Salvar**  **Limpar**  **Fechar**

Figura 16: Tela de entrada dos produtos

Na figura 16 o usuário fará a entrada dos produtos no estoque da seguinte forma: informa o nome do produto se não encontrado o sistema retorna uma mensagem de erro “Produto Não Encontrado”, sendo encontrado o sistema retorna o código e o preço unitário do produto já cadastrado, depois é preciso inserir a data, a quantidade do produto, o número da nota fiscal e o fornecedor, se não for encontrado o usuário poderá ir para tela de cadastro de fornecedor clicando em um botão que tem ao lado do campo fornecedor, também pode ser atualizado o preço unitário no caso de necessidade, para finalizar clicado no botão salvar a quantidade do produto é alterada.

Após ser feita a entrada dos produtos no estoque poderão ser realizadas as vendas desses produtos através da funcionalidade vendas que é mostrada na figura 17.

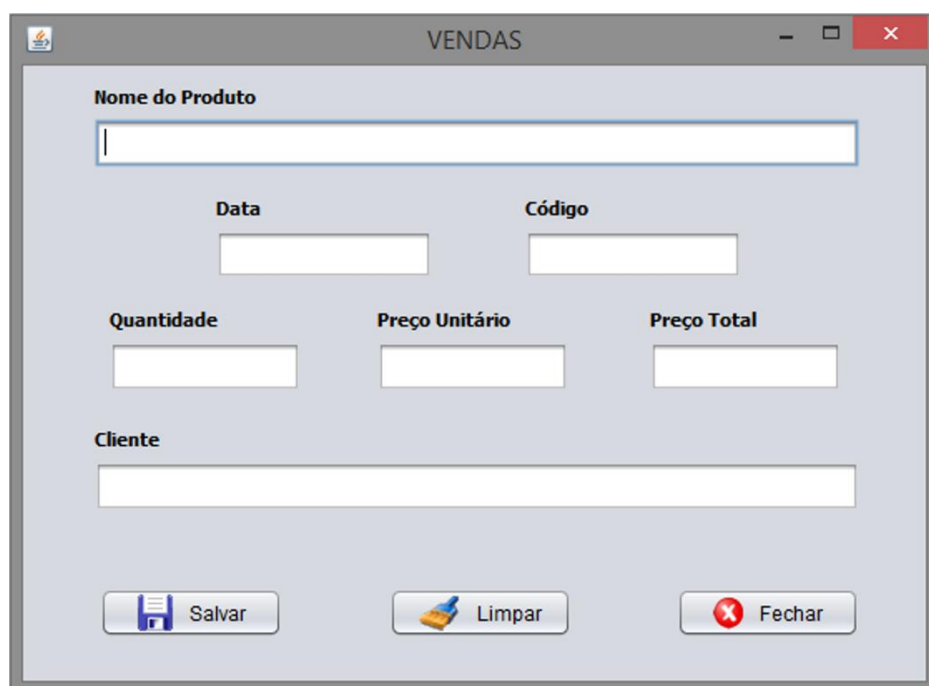
A imagem mostra uma janela de software intitulada "VENDAS". No topo, há uma barra de título com o ícone do sistema, o nome "VENDAS" e os botões de minimizar, maximizar e fechar. O formulário principal contém os seguintes campos: "Nome do Produto" (um campo de texto largo), "Data" (um campo de texto), "Código" (um campo de texto), "Quantidade" (um campo de texto), "Preço Unitário" (um campo de texto), "Preço Total" (um campo de texto) e "Cliente" (um campo de texto largo). Na base da janela, há três botões: "Salvar" com um ícone de disquete, "Limpar" com um ícone de borracha e "Fechar" com um ícone de uma cruz vermelha.

Figura 17: Tela de venda de produtos

A funcionalidade de vendas mostrada na figura 17, funciona da seguinte maneira, o usuário informa o nome do produto a ser buscado no banco, caso não estiver cadastrado o sistema retorna uma mensagem de erro “Produto Não Encontrado”, sendo encontrado serão preenchidos os campos código e preço unitário do produto, logo após o usuário terá que preencher os outros campos data da venda, quantidade a ser vendida, e o nome do cliente, onde o sistema também

faz a busca, caso não encontrado o sistema mostra a mensagem de erro “Cliente não cadastrado”, se o usuário achar necessário tem um botão onde abre a tela de cadastro de cliente, podendo ser cadastrado no sistema, se não, informa um cliente consumidor e faz a venda do produto. Quando for inserida a quantidade o sistema faz o calculo e retorna o valor total da venda e mostra no campo referente. No final quando o usuário clicar no botão salvar é decrementado a quantidade do produto que foi vendido, fazendo assim o controle.

8 CONCLUSÃO

Neste trabalho foi feito o estudo, descrição e implementação de um sistema para o controle de estoque da empresa Auto Molas JM, para isso foi realizado a análise e levantamento de requisitos, sendo possível o desenvolvimento do sistema proposto nesse trabalho de conclusão de curso e alcançando assim o objetivo.

Foi usada a tecnologia java seguindo as técnicas de orientação a objeto. Pode-se dar continuidade implementando novos módulos para o sistema como: pedidos, orçamentos, impressão de relatórios, gerar relatórios de compras, relatórios de vendas por clientes e datas, assim como outros módulos ficando a critério de novos estudos.

REFERÊNCIAS BIBLIOGRÁFICAS

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas UML**, Rio de Janeiro, Elsevier, 2007 – 2ª Reimpressão.

CORADI, Cleomar. **Desenvolvimento de uma ferramenta de auxílio ao controle de veículos e pós-vendas da empresa Irmãos Coradi**. Balsas, MA, 2012.

DEITEL, Harvey M.; DEITEL, Paul J. **Java Como Programar**. 6ª edição, São Paulo, 2005.

MECENAS, Ivan. **Java 6 Fundamentos, Swing, BlueJ e JDBC**. 3ª Edição, Rio de Janeiro, Alta Book, 2008.

PRESSMAN, Roger S. **Engenharia de Software**. 6ª ed. Editora McGraw-Hill, São Paulo, 2006.

REZENDE, D. A. **Engenharia de Software Empresarial**. Rio de Janeiro: Brasport, 1997.

REZENDE, Denis Alcides. **Engenharia de Software e Sistemas de Informação**. 3ª ed. rev. e ampl., Rio de Janeiro: Brasport, 2005.

SCHWABER, K. **Agile Project Management With Scrum**, Microsoft, 2004.

SOMMERVILLE, Ian. **Engenharia de Software**. 6ª edição, São Paulo, Addison Wesley, 2003.

SOUSA, Ernando Gomes de. **Usando o paradigma orientado a aspectos na análise e projeto do silma (sistema de locação de multimídias para a faculdade de balsas)**. Balsas, MA, 2011.

VARASCHIM, Jacques Douglas. **Implantando o SCRUM em um Ambiente de Desenvolvimento de Produtos para Internet**. Rio de Janeiro, 2009.

Java.com. Disponível em: <http://www.java.com/pt_BR/about>. Acesso em: 24 mar. 2013.

Netbeans.org. Disponível em: <https://netbeans.org/index_pt_BR.html>. Acesso em: 23 maio 2013.

Oracle.com. Disponível em: <<http://http://www.oracle.com/technetwork/java/javaee/overview/index.html>>. Acesso em: 1 nov. 2013.

APÊNDICE

APÊNDICE 1: Documento Requisitos

Este documento tem a finalidade de mostrar os requisitos funcionais e não funcionais do sistema Controle do Estoque, desenvolvido para a Empresa Auto Molas JM.

Requisito Funcionais: descreve as funções que o sistema deve conter, como o sistema reage a entradas específicas e como deve se comportar em determinadas situações.

Requisitos Não Funcionais: descreve as qualidades e restrições sobre os serviços e funções oferecidas pelo sistema como usabilidade e desempenho.

1.1 Requisitos Funcionais

[RF - 01] Cadastrar Produto

O sistema deverá permitir cadastrar novos produtos com todos os seus atributos (código, nome, descrição, preço de custo, preço de venda, quantidade). O cadastro não poderá ser realizado no caso de já existir no estoque um produto com o mesmo código.

[RF - 02] Excluir Produto

O sistema deverá permitir a exclusão de produtos pelo código.

[RF - 03] Alterar Produto

O sistema atualizará os dados específicos de cada produto existente na base de dados. Exceto código.

[RF - 04] Consultar Produto

O sistema permitirá a consulta de um produto já existente no banco de dados. Através do sistema padrão, a consulta sempre será feita pelo código identificador do produto ou pelo seu nome.

[RF - 05] Logar no Sistema

Todas as funcionalidades do sistema devem ficar acessíveis ao usuário mediante autenticação no sistema. Na autenticação o operador informa login e

senha, caso estejam corretos deve ser aberta sua sessão no sistema, caso a senha ou o login esteja errado deve ser mostrada mensagem de senha ou login incorretos e abrir novamente a tela de autenticação.

[RF - 06] Cadastrar Cliente

O sistema irá possuir um cadastro para os clientes, aonde o funcionário irá cadastrar o cliente com seus dados pessoais (Nome completo, CPF, RG, data de nascimento, endereço, CEP, telefone, celular e e-mail).

[RF - 07] Alterar Cliente

O sistema permitirá alterar todos os atributos do cliente com exceção de CPF e RG. Essas alterações serão feitas pelo usuário.

[RF - 08] Excluir Cliente

O sistema irá permitir a exclusão do cliente do sistema, deve ser fornecido nome, identidade ou cpf do cliente e em seguida pressionado o botão de exclusão, assim excluindo o cliente do banco de dados.

[RF - 09] Cadastrar Fornecedor

O sistema irá possuir um cadastro para os fornecedores, aonde o usuário irá cadastrar o fornecedor com seus dados (Razão social, nome fantasia, CNPJ/CPF, Insc. Estadual, Insc. Municipal, endereço, cidade, estado, cep, telefone, celular e e-mail), para facilitar futuras compras desse mesmo fornecedor.

[RF - 10] Alterar Fornecedor

Caso o fornecedor tenha mudado alguma de suas informações, o sistema permitira alterar todos os seus atributos com exceção do código.

[RF - 11] Excluir Fornecedor

O sistema irá permitir a exclusão do fornecedor do sistema, caso seja solicitado, aonde o usuário irá realizar essa tarefa. O sistema irá apagar completamente as informações daquele fornecedor.

[RF - 12] Registrar Movimentação de Produtos

O sistema registrará toda movimentação dos produtos no banco de dados,

guardando as informações como entrada e saída. A entrada (compra) que é feita através de uma tela do sistema que o usuário utiliza para atualizar a quantidade de produtos no estoque. A saída (venda) é realizada através de uma tela de vendas desses produtos onde é realizada a retirada desses produtos do estoque.

[RF - 13] Visualizar Estoque

O sistema possuirá um modulo de visualização do estoque, aonde poderá ser acessado pelo usuário. O usuário irá escolher a opção estoque para visualizar os produtos cadastrados e disponíveis para a venda.

1.2 Requisitos Não Funcionais

PROCESSO

[RNF - 01] Sistema desenvolvido na linguagem Java de modo a ser compatível com o sistema operacional Windows, podendo ser estendido aos demais sistemas operacionais. Utilizando um sistema gerenciador de banco de dados *open source*.

[RNF - 02] Será criado um documento contendo um diagrama de classes e diagrama de caso de uso.

[RNF - 03] Será criado um cronograma detalhado para o processo de desenvolvimento no qual constem: as atividades a serem desenvolvidas e em que período e com que recursos humanos serão desenvolvidos o sistema.

SEGURANÇA

[RNF - 04] Os usuários terão que ter permissão para utilizar algumas funcionalidades do sistema, deverá utilizar do login e senha para manipular estoque dos produtos.

USABILIDADE

[RNF - 05] A interface do sistema será agradável e objetiva ao usuário. Suas funcionalidades e informações deveram estar bem visíveis e disponíveis.

[RNF - 06] Comunicação sistema e usuário com mensagens simples, explicativas do erro gerado.

DESEMPENHO

[RNF - 07] O Sistema usará um banco de dados MySQL, assim garantindo a segurança dos dados, mas também agilizando desempenho do sistema.

[RNF - 08] Para um melhor desempenho do sistema é recomendada uma máquina razoável. Com os seguintes requisitos mínimos: Processador Dual Core, 512Mb de Memória, espaço mínimo no HD de 5 GB.