

SISTEMA PARA INFORMATIZAÇÃO DE COMÉRCIO EM JAVA COM BANCO DE DADOS MYSQL

MÁRCIA REGINA DOS SANTOS¹
FELIPE DINIZ DALILLO²
FABIANA FLORIAN³

RESUMO: O objetivo deste trabalho é informatizar um sistema para de comércio. São apresentados os processos de elaboração do programa Netbeans na Linguagem JAVA com foco no Pacote view. Foi utilizada Linguagem Java com conexão ao banco de dados MySql e o padrão MVC, que consiste em um modelo de divisão de camadas model, view e controller, associado com a tecnologia de metodologia scrum e o método cascata. No Pacote View foram elaborados as telas referentes ao sistema: Tela de Login e Senha; Tela Principal, Menu Cadastro, Menu Estoque e o Menu Encomenda. Conclui-se que o sistema permitiu organizar os dados dos clientes, dos produtos e dos fornecedores atendendo os principais requisitos do estabelecimento.

Palavras-chave: Sistema comercial, banco de dados MySQL, padrão MVC.

ABSTRACT:

The objective of this work is to computerize a system for commerce. The processes of elaboration of the program Netbeans in the Language JAVA with focus in the View package are presented. Java language with connection to the MySql database and the MVC standard, which consists of a model division model, view and controller, associated with the scrum methodology technology and the cascade method were used. In the View Package were elaborated the screens referring to the system: Login Screen and Password; Main Menu, Registration Menu, Stock Menu, and the Order Menu. It is concluded that the system allowed to organize the data of the customers, the products and the suppliers, taking into account the main requirements of the establishment.

Keywords: Business system, MySQL database, MVC standard.

¹ Graduanda em Sistemas de Informação da Universidade de Araraquara-UNIARA. E-mail: marciaheavy@hotmail.com

² Orientador. Mestre da Universidade de Araraquara-UNIARA. E-mail: felipedalillo@hotmail.com

³ Coorientadora. Doutora da Universidade de Araraquara-UNIARA. E-mail: fflorian@uniara.com.br

1 INTRODUÇÃO

Sistemas de banco de dados servem dentre outras atividades, para armazenar informações, no sistema apresentado, armazena-se pedidos de compra e de pagamentos executados (como também suas formas de pagamentos), para gerenciamento do setor comercial, financeiro e fiscal. O objetivo destes sistemas são criar soluções que facilitem o dia a dia nas atividades do comércio com sistemas de controle de vendas e cadastro de clientes de maneira mais eficaz. (<https://pt.linkedin.com/pulse/o-que-%C3%A9-um-banco-de-dados-e-qual-sua-import%C3%A2ncia-para-william-miranda>)

O desenvolvimento do sistema apresentado consiste em um sistema de fluxo de armazenamento de dados e controle para o comércio ou serviços em geral, tem como objetivo agilizar e tornar mais fácil a rotina diária do trabalho nestes departamentos, visando ter mais dinamismo e eficácia no armazenamento e consulta e na agilidade nos pedidos dos clientes. Esse sistema visa não apenas facilitar o desempenho dos colaboradores, como também a tomada de decisões da parte da gestão como, por exemplo, em que deve-se investir mais, apresentar melhorias, em que ponto deve-se focar mais atenção, para saber como vai a saúde financeira de sua empresa, ou seja, se obtém lucros ou prejuízos.

O objetivo específico deste sistema é dinamizar e facilitar o processo de armazenamento de informações, de maneira a tornar a rotina do setor e dos departamentos mais ágil e dinâmica, como também a eficácia de encontrar informações.

Ultimamente, há a necessidade de um sistema para tornar mais ágil o funcionamento dos setores e facilite o trabalho de um estabelecimento que haja um grande fluxo de clientes e exige mais velocidade de atendimento, de modo a satisfazer o cliente e facilitar os colaboradores para exercer suas atividades diárias.

Este sistema procura facilitar essas atividades e satisfazer os clientes com um atendimento mais rápido e preciso, minimizando alguns possíveis erros, e por meio dessa tecnologia, conseguir obter maior desempenho, com a finalidade de não atrasar o processo.

A competitividade no setor comercial em geral exige uma tecnologia que ajude o cliente, por essa razão, torna-se essencial qualquer diferencial que coloque a empresa a frente. Esse software se apresenta como uma forma de ajudar a solucionar o planejamento estratégico, aumentando a chance das empresas à sobreviverem no mercado e a agilizar o trabalho da equipe.

Ao programar softwares comerciais, podem haver diversos problemas, como por exemplo, dificuldade de implementação do sistema e a necessidade de oferecer treinamento e qualificação a equipe, para operar o sistema. Pode-se também ocorrer erros, e nesse caso, é necessário minimiza-los, e para desenvolver um software de qualidade, é necessário que se resolva de maneira eficaz, havendo um gasto financeiro se não houver a implementação de um sistema que atenda a essas necessidades.

O tempo de atendimento e o treinamento aos colaboradores são cruciais no sucesso da empresa, o que implica em ter um software que necessite ser de fácil manuseio, e com manutenção e atualização periódica, pois, provavelmente, a intenção do empresário é crescer e expandir os seus negócios.

Antes de tudo, é necessário fazer um estudo do estabelecimento, suas necessidades, seus pontos fortes e fracos, para podermos implementar um software personalizado que vá de encontro as suas necessidades.

Foi realizada pesquisa bibliográfica sobre Banco de Dados e MYSQL. Foi utilizada a Linguagem Java por ser considerada mais versátil, dinâmica, robusta e ágil, que se adapta a diversas formas de programação, desde operações em sistemas embarcados até em operações web e aplicações. Pode ser utilizada em qualquer sistema operacional e em qualquer tipo de aparelho, como smartphones e tablets, juntamente com o banco de dados MYSQL.(<https://www.embarcados.com.br/editorial-linguagens-para-sistemas-embarcados/>)

Depois de levantado todos os eventos e ações, foi feito diagrama de classe e o diagrama de caso de uso, para facilitar na implementação do software, e utilizar o Scrum para fazer os planejamentos do projeto, para poder ter o controle de cada etapa do processo.

A próxima etapa é a de levantamento de requisitos para o banco de dados, funções do sistema e níveis de acesso, para poder-se implementar o sistema e suas funcionalidades.

2 SISTEMA GERENCIADOR DE BANCO DE DADOS (SGBD)

Um Sistema Gerenciador de Banco de Dados (SGBD) é constituído por um conjunto de dados associados a um conjunto de programas para acesso a esses dados. O conjunto de dados, comumente chamado de banco de dados, contém informações sobre um determinado local (empresa ou instituição) em particular. O principal objetivo de um SGBD é proporcionar um ambiente tanto conveniente quanto eficiente para a recuperação e armazenamento de informações do banco de dados. (SILBERSCHATZ; KORTH; SUDARSHAN, 2012).

Neste caso, os autores explicam que SGBD's são sistemas que cujo qual é possível manipular e armazenar informações de um determinado setor, garantindo a segurança e a integridade das informações guardadas para que possam ser acessadas ou modificadas quando necessário, com um conjunto de algoritmos que tratam essas informações arquivadas.

Para poder se planejar e estruturar esses dados de maneira visual, há os Diagramas de caso de uso, os Diagramas de Classes, entre outros:

Um SGBD é uma coleção de arquivos e programas inter-relacionados que permitem ao usuário o acesso para as consultas e alterações desses dados. O maior benefício de um banco de dados é proporcionar ao usuário uma nova visão abstrata dos dados. Isto é, o sistema acaba por ocultar determinados detalhes sobre a forma de armazenamento e manutenção desses dados (SILBERSCHATZ, KORTH, SUDARSHAN).

Para se projetar um sistema, é necessário passar por várias etapas para garantir uma boa funcionalidade do sistema, como por exemplo, planejamento e compatibilidade de software e hardware, planejamento de infraestrutura, segurança, entre outros.

Segundo Sommerville (2008), o papel da medição de software é quantificar alguns atributos de um processo de software, comparando essas informações é possível tirar conclusões sobre a qualidade do software, possibilitando mensurar se as mudanças organizacionais (adoção de novas ferramentas ou metodologias), sendo positivas ou não para os processos de desenvolvimento. Nesse caso são feitas medições antes e depois da mudança para verificar se foi positiva ou não para a organização.

As principais etapas para se projetar um sistema são: Levantamento de requisitos; Requisitos de hardware; Requisitos de infraestrutura e segurança; Requisitos de usabilidade.

O levantamento de requisitos é uma das etapas mais importantes para compor um projeto, sendo que ele é um esquema que nos ajuda a entender as diversas etapas do mesmo, para compreender aquilo que o cliente necessita e as regras do negócio. É uma das principais funções que compõem a Engenharia de Requisitos. (<https://www.devmedia.com.br/o-gerenciamento-do-escopo-do-projeto/28418>)

Junto ao levantamento de requisitos, realiza-se também o mapeamento dos processos, que é muito importante para a melhoria de resultados referentes ao levantamento de requisitos. Muitos projetos atrasam na fase de definição de escopo ou então até tornam-se impossível dar continuidade se esta etapa for feita de maneira incorreta, o que pode gerar um sistema de alto custo e difícil manutenção. (Fonte – Jones, Carpen - 1997.Applied Software Measurement. <https://www.baguete.com.br/artigos/296/ricardo-verissimo/05/11/2007/levantamento-de-requisitos-e-mapeamento-de-processos>)

Define-se requisitos de hardware o que diz respeito a funcionalidade da máquina e a capacidade de memória, disco rígido, placas de vídeo, enfim, que suportam o tamanho de um determinado sistema. ([https://technet.microsoft.com/pt-br/library/cc263100\(v=office.12\).aspx](https://technet.microsoft.com/pt-br/library/cc263100(v=office.12).aspx))

Os Projetos de Infraestrutura de TI são multidisciplinares, ou seja, envolvem mais de uma área técnica. O responsável por um projeto desta natureza precisa estar muito atento de que forma as tecnologias envolvidas deverão ser implementadas. (<https://pt.linkedin.com/pulse/projetos-de-infraestrutura-ti-pre-requisitos-vinicius-camoes>)

Em um projeto de Infraestrutura de TI podemos envolver áreas como:

- Energia (KVA)
- Cabeamento de Redes (UTP e Fibra)
- Configuração de Redes (Switches, Roteadores, Firewalls e etc)
- Hardware (Servidores, Storages, Appliances e etc)
- Sistemas Operacionais (Unix, Linux, Windows, z/OS e etc)
- Aplicações de apoio (Banco de Dados, Anti-Vírus, Backup e etc)

Desta forma, na fase de planejamento o Gerente de Projetos deverá contar fortemente com a colaboração do corpo técnico, fazer com que o time esteja efetivamente envolvido, tendo em vista que o levantamento das precedências entre as atividades, o tempo de duração de cada uma, bem como a identificação dos pré-requisitos e riscos do projeto serão cruciais para o desenvolvimento do trabalho e um planejamento mais assertivo. (<https://pt.linkedin.com/pulse/projetos-de-infraestrutura-ti-pre-requisitos-vinicius-camoes>)

No levantamento dos pré-requisitos poderemos desmembrar uma série de ações. Caso não sejam considerados, poderão acarretar na paralisação, atraso ou até mesmo inviabilidade do projeto.

Seguem abaixo apenas alguns exemplos clássicos:

- Capacidade

Sempre que um projeto for tratar da implementação de equipamentos, deve-se estar atento à capacidade do local (Sala Cofre, CPD ou similar) quanto à disponibilidade de energia (KVA), pontos de rede, espaço físico para alocação de racks, compatibilidade do equipamento a ser instalado nos racks disponíveis e etc.

- Segurança

Verificar com a equipe de segurança sobre a viabilidade de abertura de portas para determinados serviços a serem implementados. Hoje em dia, além dos firewalls de borda, também há a firewalls internos, além de outras ferramentas de segurança (<https://pt.linkedin.com/pulse/projetos-de-infraestrutura-ti-pre-requisitos-vinicius-camoes>)

À princípio, deve-se começar identificando o sistema, quem o solicitou e quem fornece, e onde será instalado.

No caso de Banco de Dados, deve-se atentar-se a quantidade de espaço em disco necessário e ao aumento de custo. Há vários tópicos que devem ser analisados, como, por exemplo: (<https://gestaoemti.com/2016/06/22/requisitos-de-infraestrutura-de-ti/>)

- Qual o sistema de banco de dados e versão;
- Se o banco de dados é suportável em cluster;

- Requisito mínimo de CPU, memória e disco;
 - Requisitos específicos sobre a configuração dos discos, exemplos raid-10 ou discos de 15 krpm;
 - Se existe um sistema operacional compatível ao banco de dados;
 - Tipo de autenticação para acesso ao banco, exemplos: para cada usuário da aplicação corresponderá a um usuário único para a conexão do banco de dados ou autenticação via Windows;
 - A existência de um requisito específico para o backup e a necessidade de instalação de algum aplicativo adicional no servidor de SGBD;
 - Periodicidade de rotinas de limpeza de dados;
- (<https://gestaoemti.com/2016/06/22/requisitos-de-infraestrutura-de-ti/>)

Os requisitos de segurança, definem-se em conjuntos de necessidades de segurança que o software deve atender, sendo tais necessidades influenciadas fortemente pela política de segurança da organização, e compreendendo aspectos funcionais e não funcionais. Os aspectos funcionais descrevem comportamentos que viabilizam a criação ou a manutenção da segurança, e geralmente podem ser testados diretamente. Na maioria dos casos, remetem a mecanismos de segurança como, por exemplo, controle de acesso baseado em papéis de usuários (administradores, usuários comuns, entre outros), autenticação com uso de credenciais (usuário e senha, certificados digitais), entre outros. (<https://gestaoemti.com/2016/06/22/requisitos-de-infraestrutura-de-ti/>)

Já, a engenharia de usabilidade é um processo que as características de usabilidade são especificadas antecipadamente, de forma quantitativa no processo de desenvolvimento, e medidas durante todo o processo. O conceito de usabilidade é para facilitar o processo de aplicações complexas, visando melhorar as características dos produtos e tornando-os mais rentáveis. (http://www.funpar.ufpr.br:8080/rup/process/workflow/requiem/co_req.htm)

3 MODELO SCRUM

Scrum é uma metodologia ágil usada para gestão dinâmica de planejamento de projetos de software, que permite controlar de forma eficaz a equipe, potencializando o trabalho das equipes envolvidas, pois os projetos são divididos em

vários ciclos, denominados Sprints, que são etapas que se referem ao conjunto de atividades específicos que deverão ser realizadas. (<https://www.significados.com.br/scrum/>)

Um exemplo de metodologia Scrum pode ser observado na Figura 1.

Dentre as várias etapas do processo, as funcionalidades são mantidas em uma lista, o Product Backlog, que é uma lista onde são mantidas as implementações dos projetos que serão realizados e se transformarão em um produto. (<https://www.significados.com.br/scrum/>)

O Kanban é a organização de todos os trabalhos que deverão ser feitos pela equipe, são divididos em estágios: A fazer; fazendo; em testes; concluído. (<http://www.jrmcoaching.com.br/blog/como-funciona-a-metodologia-scrum/>)

Daily Scrum, se trata de uma reunião rápida para o grupo verificar o que cada um fez no dia anterior, se há algum problema impedindo o andamento do trabalho e o que farão no dia atual. (<https://www.significados.com.br/scrum/>)

Sprint Review Meeting é uma reunião para ficar a par dos progressos de cada etapa. (<http://www.jrmcoaching.com.br/blog/como-funciona-a-metodologia-scrum/>)

Burndown é o gráfico de medição de evolução de tarefas, listando o número de tarefas e a complexidade de cada uma. (<http://www.jrmcoaching.com.br/blog/como-funciona-a-metodologia-scrum/>)

O método scrum tem como objetivo auxiliar no gerenciamento de maneira mais efetiva e melhorar na produtividade e no foco da solução do problema. (<http://www.jrmcoaching.com.br/blog/como-funciona-a-metodologia-scrum/>)

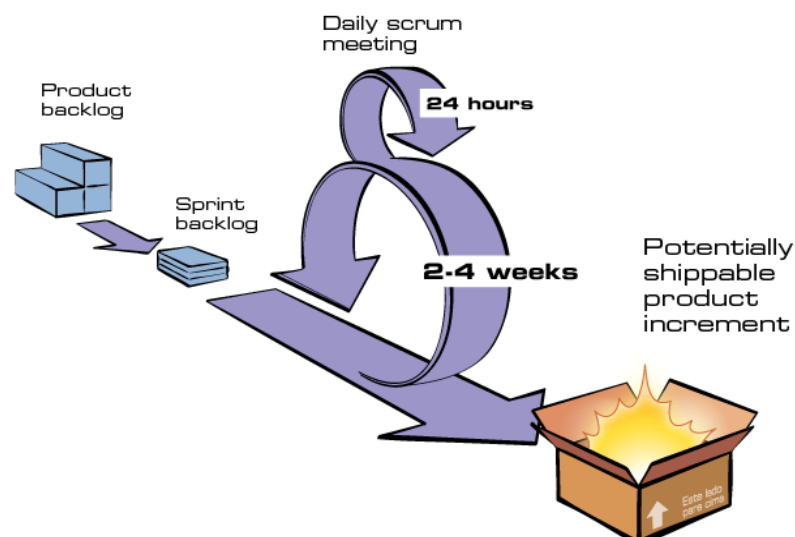


Figura 1 – Exemplo de metodologia Scrum

Fonte: SCRUM

4 MODELO EM CASCATA

O Modelo em cascata é um dos mais importantes da Engenharia de Softwares, que consiste em estruturar os projetos em formato de cascata, pois um método dará sequência para outro método a seguir, incluindo desde o estudo de viabilidade até a implementação, tendo como finalidade determinar a ordem no desenvolvimento de grandes produtos de software. (Pressman, R. Engenharia de Software: Uma abordagem Profissional. 7ª edição. Editora Bookman)

Segundo Pressman (2011), o modelo em cascata, conhecido também como ciclo de vida clássico, apresenta várias etapas (Figura 2) - sugere iniciar pelo levantamento de necessidades por parte do cliente, passando para a fase de planejamento, modelagem, construção, emprego e culminando no suporte contínuo do software concluído.

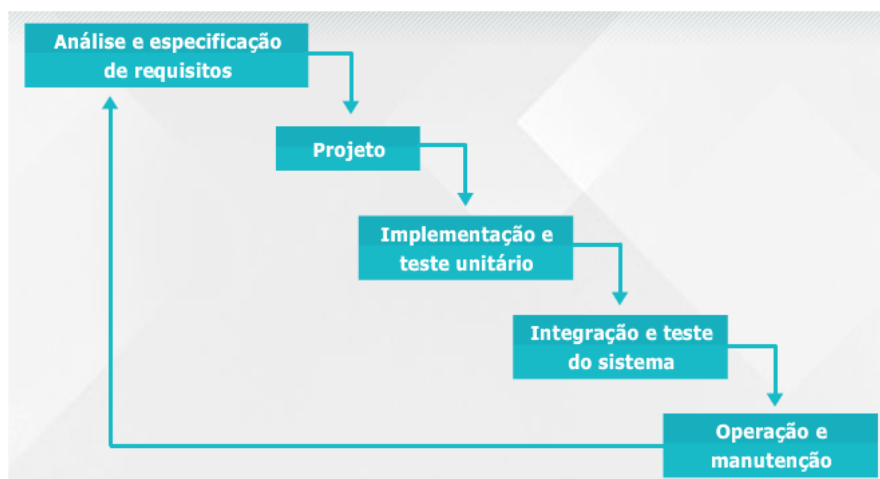


Figura 2 - Etapas no modelo cascata

Fonte: PRESSMAN, 2006.

5 LINGUAGEM JAVA

A linguagem Java foi criada em 1991 por um grupo de engenheiros da Sun Microsystems, chefiados por James Gosling, que procuravam fazer uma linguagem

que pudesse ser utilizada em pequenos equipamentos como controles de TV, telefones, fornos, etc que fosse capaz de executar códigos pequenos e simples, e praticamente infalíveis. (INEAD)

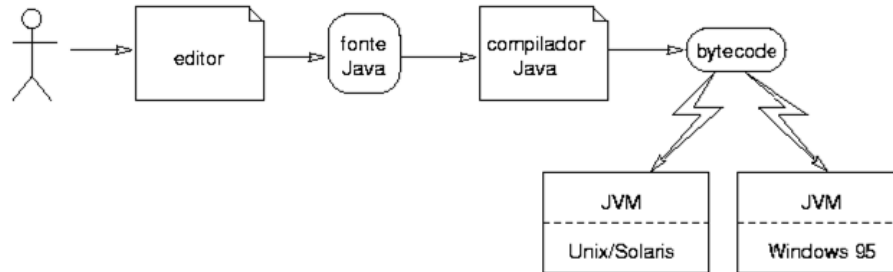


Figura 3 – Exemplo de bytecodes e JVM

<http://www.dca.fee.unicamp.br/cursos/PooJava/javaenv/bytecode.html>

Java contém como características ser compilada para um bytecode que é interpretado por uma máquina virtual JVM - Java virtual Machine. (Figura 3) A linguagem de programação Java é a linguagem convencional da Plataforma Java. (<http://www.dca.fee.unicamp.br/cursos/PooJava/javaenv/bytecode.html>)

Os quatro pilares fundamentais da Linguagem JAVA são: Orientada a Objetos; Polimorfismo; Encapsulamento; Herança. (<https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>)

A programação Orientada a Objetos (POO) segue um determinado padrão de desenvolvimento que é seguido por muitas linguagens, como C# e Java. A programação orientada a objetos é diferente da programação estruturada, na primeira os métodos são aplicados aos dados de cada objeto, enquanto que na programação estruturada, há os procedimentos, ou as funções, que são aplicados globalmente.

Polimorfismo é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que tem a mesma identificação (assinatura) mas comportamentos distintos, especializados para cada classe derivada, usando para uma referência a um objeto do tipo da superclasse. A decisão sobre qual o método que deve ser selecionado, de acordo com o tipo da classe derivada, é tomada em tempo de execução, através do mecanismo ligação tardia. No polimorfismo é necessário que os métodos tenham exatamente a mesma identificação, sendo utilizado o mecanismo de redefinição de métodos. Esse

mecanismo de redefinição não deve ser confundido com o mecanismo de sobrecarga de métodos.(<https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>)

O Encapsulamento é um conceito da Programação Orientada a Objetos onde o estado de objetos (as variáveis da classe) e seus comportamentos (os métodos da classe) são agrupados em conjuntos segundo seus graus de relação. Assim sendo, o propósito do encapsulamento é organizar os dados que sejam relacionados, agrupando-os em objetos (classes), reduzindo as colisões de nomes de variáveis e, da mesma forma, reunindo métodos relacionados às suas propriedades (ou variáveis de classe). Este padrão ajuda a manter um programa com centenas ou milhares de linhas de código mais legível e fácil de trabalhar e manter.(<https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>)

A herança é um dos paradigmas da POO, que permite a criação de novas classes a partir de outras previamente criadas, essas novas classes são chamadas de subclasses ou classes base. É possível criar uma hierarquia dessas classes, tornando classes mais amplas e classes mais específicas. Uma das grandes vantagens de usar o recurso da herança é a reutilização de códigos. Esse aproveitamento pode ser acionado quando se identifica que o atributo ou método de uma classe será igual para as outras. Para efetuar a herança de uma classe é utilizada a palavra reservada chamada `extends`.(<https://www.devmedia.com.br/entendendo-e-aplicando-heranca-em-java/24544>).

5.1 Padrão MVC

O padrão MVC consiste em um formato da linguagem de programação que o sistema é dividido em 3 camadas (model, view, controller) e alguns divididos em até 4 camadas (model, view, controller, persistent).(Aula Introdução a Arquitetura de Software - Prof. Marcelo Torres UNIARA).

Na camada model está contido as classes de aplicações, métodos construtores e métodos getters e setters, na camada view, representam a parte visual do sistema, que seriam as telas, ou seja, as Jframes, que é a interface do sistema com o usuário, a camada controller é responsável por interligar a camada view à camada model e direciona o fluxo da aplicação, o que permite escolher qual

view será exibida ao usuário em resposta a entrada e a lógica de negócio que será processada. (<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>).

O persistente, utilizado em alguns sistemas, ou Java Persiste API é uma padrão do Java para persistência de dados que deve ser implementada por frameworks que queiram seguir este padrão, como o JPA é usado na camada de persistência com o objetivo de ter maior produtividade dos desenvolvedores, com ele controlarmos a persistência dentro do Java, com isso teremos um padrão para mapear nossos objetos para Banco de Dados. (Nelson Nunes Campos - <http://fabrica.ms.senac.br/2013/06/java-persistence-api-jpa/>)

6 DIAGRAMAS COMPORTAMENTAIS

Os diagramas comportamentais têm a função de auxiliar a visualizar, construir e documentar de maneira dinâmica o sistema, mostrando como os elementos se relacionam entre si e para descrever a lógica de determinado componente do sistema. Se classificam em: Diagrama de Classe e Diagrama de Caso de Uso. (Artigo UML: Diagramas comportamentais – Autor Eduardo Figueiredo http://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/uml-diagramas-comportamento_v01.pdf)

O Diagrama de Classe tem como finalidade especificar os tipos de objetos e o relacionamento entre eles. (<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/classes/classes1.htm>)

O diagrama de Caso de Uso, por sua vez, tem a finalidade de auxiliar a comunicação entre os analistas e o cliente, descrevendo um cenário que mostra as funcionalidades do sistema. A utilização de um diagrama de caso de uso é importante para expressar sobre a “fronteira” do seu sistema e modelar os requisitos do mesmo, permitindo uma visão geral dos relacionamentos entre os casos de uso e seus atores. (<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/classes/classes1.htm>)

7 FUNCIONALIDADES DO SISTEMA

A seguir são apresentados os vários módulos de elaboração do programa Netbeans na Linguagem JAVA com foco no Pacote view.

No Pacote View foram elaborados as telas referentes ao sistema: Tela de Login e Senha; Tela Principal, Menu Cadastro, Menu Estoque e o Menu Encomenda.

7.1 Tela de Login e Senha

A tela de login e senha (Figura 4) tem por objetivo iniciar o sistema. Para que o usuário possa ter acesso a parte interna do sistema, deve efetuar o login (usuário e senha), para que assim possa ir para a tela de cadastros(cadastro de clientes, cadastro de produtos, cadastro de fornecedores), ou a demais funcionalidades do sistema. Na tela de login e senha o usuário faz a sua autenticação, se caso a senha estiver correta, é possível acessar o sistema, caso contrário, é exibida a mensagem “Usuário ou senha inválidos”.



Figura4 - Tela de login e senha

Fonte – Própria

7.2 Tela Principal

Na tela inicial do sistema há os menus Cadastro, Estoque e Encomenda (Figura 5) com os seguintes sub menus no Cadastro:

- Cliente: no cadastro de cliente vai se cadastrar os dados mais importantes do cliente.
- Produto: no cadastro de produtos vai se cadastrar os produtos existentes na loja, caso chegue novos produtos faz-se um novo cadastro de produto.
- Fornecedores: no cadastro de fornecedores cadastram-se todos os dados necessários dos fornecedores de materiais e produtos á loja bem como prazos e preços.

Figura 5 - Tela Inicial do Sistema

Fonte - Própria

7.3 Menu Cadastro

O menu Cadastro é composto pelos sub-menus: Cadastro de Clientes, Cadastro de Produtos e Cadastro de Fornecedores. Ao clicar nesses sub-menus abrem os respectivos formulários de cadastro.

7.3.1 Sub menu Cadastro de Clientes

A tela Submenu Cadastro de Clientes (Figura 6) é composta por um formulário onde é possível cadastrar um determinado cliente e manter seu contato, como também enviar novidades e ofertas.

No submenu Cliente é possível realizar ações como cadastrar um novo cliente, atualizar dados, excluir cliente e exibir informações de um determinado cliente. O

Botão Salvar efetua o cadastro dos dados dos clientes que são inseridos no formulário, o botão Atualizar serve para modificar informações do cliente, o botão Exibir procura um determinado cliente pelo seu CPF ou nome, e o botão Excluir apaga determinado cliente do sistema.

ID	CPF	DATA NAS.	RG	NOME	EMAIL	ENDEREÇO	BAIRRO	TELEFONE	CELULAR	CELULAR2	CIDADE	UF
1	123.456.7...	11/01/2000	12345678-9	Maria Crist...	mariacris...	Rua São B...	Centro	(16)3322-...	(16)99887-...	(16)98765-...	Araraquara	SP

Figura 6 - Tela de Cadastro de Cliente

Fonte: Própria

7.3.2 Sub menu Cadastro de Produtos

Na tela Sub menu de cadastro de produtos (Figura 7) é composta por um formulário onde se realiza o cadastro dos dados dos produtos produzidos e vendidos.

No submenu Produtos é possível realizar ações como efetuar cadastro de produto, atualizar dados do produto, excluir e exibir informações de um determinado produto. Os botões de ação Salvar, Pesquisar e Excluir, cujos quais o Salvar é responsável por incluir dados do cadastro de produtos, o botão alterar é através dele que é possível fazer modificações nas informações do produto o botão exibir por encontrar o produto através de seu código e o botão Excluir excluirá o produto cadastrado do sistema.

Código	Marca	Característica	Descrição	Fornecedor	Data de Fabric...	Data de Valida...	Preço
1	Mimos e Capri...	Boneca African...	Bon. Africana	Mimos & Capri...	12/04/2018	28/04/2018	25,00

Figura 7 - Tela de cadastro de Produtos

Fonte - Própria

7.3.3 Sub menu Cadastro de Fornecedores

A tela Sub menu cadastro de fornecedores (Figura 8) é composta por um formulário que se cadastra os fornecedores de materiais e produtos já manufaturados para o estabelecimento, para se organizar os prazos de entrega, prazos de pagamentos, entre outros.

No submenu Fornecedores é possível realizar ações como efetuar cadastro de fornecedor, atualizar dados, excluir e exibir informações de um determinado fornecedor. Os botões de ação Salvar, Pesquisar e Excluir, cujo qual o Salvar é responsável por incluir dados do cadastro de fornecedores, o botão alterar é através dele que é possível fazer modificações nas informações do fornecedor, o botão exibir tem a função de encontrar o fornecedor através de seu id e o botão Excluir excluirá o fornecedor cadastrado do sistema.

Cadastro - FORNECEDORES

Dados do Fornecedor

ID: CNPJ: IE:

Razão Social: Nome Fantasia:

Contato

Endereço:

Bairro: Cidade: UF:

Telefone: Celular:

Email:

Fornecimento

Código Produto:

Produto Fornecido:

Preço: Data de Pagamento:

Data de Entrega: Quantidade fornecida:

ID	CNPJ	IE	Razã...	Nom...	Ende...	Bairro	Telef...	Celul...	Cida...	UF	Email	Codi...	Prod...	Preço	Data ...	Data ...
1	07.2...	2124...	Mari...	Mun...	Rua ...	Centro	(16)3...	(16)9...	Arara...	SP	7896...	Betu...	10 lts	02/0...	22/0...	10 lts

Figura 8 - Tela de cadastro de fornecedores

Fonte - Própria

7.4 Menu Estoque

O menu estoque (Figura 9) realiza o controle de estoque de mercadorias ou matérias primas que chegam e saem da empresa. Ele é composto pelo sub menu Controle de estoque que realiza o procedimento de controlar as entradas e saídas de mercadorias, as quantidades de mercadorias e as datas de entrada, de saída e o tempo de estocagem.

O controle de estoque é composto pelos botões salvar, que cadastra o produto que chega, tem o controle da quantidade de produtos que chegam, produtos que saem e a quantidade que permanecem, e o controle da data de entrada e saída do produto, e como os demais sub menus, tem os botões alterar, caso seja necessário alterar algum produto, o botão exibir para mostrar informações referentes a determinado produto, os botões excluir, que exclui determinado produto caso pare de trabalhar com ele e o botão sair que encerra a tela.

Controle - ESTOQUE

Dados do Produto

ID: Código Produto: Quantidade Entrada: Quantidade Saída:

Descrição: Quantidade Restante:

Controle de Estoque

Data de Fabricação: Data de Validade: Data de Estocagem: Data de Saída:

ID	Codigo d...	Quantida...	Quantida...	Quantida...	Descrição	Data de F...	Data de V...	Data de E...	Data de S...
1	78965401	20	3		Jarros Jo...	22/04/2018	//	30/03/2018	24/04/2018

Figura 9 - Tela de cadastro de Estoque

Fonte - Própria

7.5 Menu Encomenda

O menu encomenda (Figura 10) é composto pelo sub menu Pedido, que realiza procedimentos de pedidos de encomendas para os clientes.

Esse sub menu de pedido de encomendas é quando um cliente deseja um determinado produto que não tem no momento, que necessita de um tempo para ser manufaturado, para que se possa ter um controle de quanto tempo será necessário para fazer o produto.

O sub menu Pedido de Encomendas possuem os botões salvar, que tem como função registrar um pedido de encomenda, botão alterar, caso o cliente deseje que seu produto seja feito de maneira diferente, botão exibir, para exibir os produtos pedidos, botão sair, caso se queira excluir um determinado pedido e botão sair que encerra a tela.

Encomendas - PEDIDOS

Pedido de Encomendas

ID: Nome do Cliente:

Endereço: Bairro:

Cidade: UF: Telefone:

Email: Celular:

Produto: Data do Pedido:

Quantidade: Preço: Data Prevista de Entrega:

ID	Nome...	Ender...	Bairro	Cidade	UF	Telefo...	Celular	Email	Produto	Quant...	Preço	Data ...	Data ...
1	Maria ...	Rua S...	Tanctr...	Arara...	SP	(16)3...	(16)9...	marial...	cofre ...	1	15,00	26/04/...	02/05/...

Figura 10 – Tela de Pedido de Encomendas

Fonte – Própria

8 IMPLEMENTAÇÃO DO MYSQL

Na sequência são apresentadas as criações das tabelas do banco de dados, com seus respectivos atributos e tipos (varchar, int, double, date), foi utilizado o banco de dados MySQL:

```

create database db_loja;
create table Cliente(
    id int not null auto_increment,
    cpf int not null,
    rg int not null,
    nome varchar (50) not null,
    endereço varchar(50) not null,
    email varchar(30),
    bairro varchar(20) not null,
    cidade varchar(30) not null,
    uf varchar(2) not null,
    telefone varchar(15),
    celular varchar(20),
    celular2 varchar(20),

```

dtNascimento date,
primary key(id));

create table Produto(
codigo int not null auto_increment,
descricao varchar(70),
caracteristica varchar(100),
fornecedor varchar(50) not null,
marca varchar(30),
preço double not null,
dtFabricacao date,
dtValidade date,
primary key(codigo));

create table Fornecedor(
id int not null auto_increment,
cnpj int not null,
ie int not null,
codigoP int not null,
razaoSocial varchar(50) not null,
nomeFantasia varchar(50),
endereco varchar(50) not null,
email varchar(30),
bairro varchar(20) not null,
cidade varchar(30) not null,
uf varchar(2) not null,
telefone varchar(15) not null,
celular varchar(20),
preco double not null,
produtoFornecido varchar (50) not null,
dtEntrega date,
dtPagamento date,
primary key(id));

create table Estoque(
id int not null auto_increment,
codigoP int not null,
quantidadeEntrada int not null,
quantidadeSaida int not null,
quantidadeRestante int not null,
descricao varchar(50) not null,
dtFabricacao date,

```
dtValidade date,  
dtEstocagem date,  
dtSaida date,  
primary key(id));
```

```
create table Encomenda(  
  id int not null auto_increment,  
  quantidade int not null,  
  nome varchar(50) not null,  
  endereco varchar(50) not null,  
  email varchar(30),  
  bairro varchar(20) not null,  
  cidade varchar(30) not null,  
  uf varchar(2) not null,  
  telefone varchar(15) not null,  
  celular varchar(20),  
  preco double not null,  
  descricao varchar (50) not null,  
  caracteristica varchar (80),  
  dtPedido date,  
  dtPrevista date,  
  primary key(id));
```

```
create table Venda(  
  codigo int not null,  
  descricao varchar (70) not null,  
  quantidade int not null,  
  preco double not null,  
  total double not null,  
  primary key(codigo));
```

9 CONSIDERAÇÕES FINAIS

Conclui-se que a plataforma Netbeans Java, juntamente com o MYSQL dinamiza de maneira bastante eficiente a produção de sistema, tornando-a funcional e facilitando na programação do sistema de maneira eficaz, sendo possível solucionar o problema de organização da infraestrutura de uma empresa. O sistema

permitiu organizar os dados dos clientes, dos produtos e dos fornecedores para que seja possível atender os principais requisitos do estabelecimento, tornando-o mais ágil e prático, resolvendo diversos problemas como perda de informações e até mesmo perda de vendas. Devido a essa informatização, foi possível otimizar os ganhos, reduzir tempo do processo de venda, além ter maior controle de todas as etapas e de todos os setores que compõem a empresa.

REFERENCIAS

BREVE HISTÓRICO JAVA – Apostila INEAD - Referencias da apostila do curso INEAD

-Centro de Computação da Unicamp

-<http://java.sun.com/docs/white/langenv/>

-<http://pt.scribd.com/doc/18510556/Java-Basico>

-Thinking in Java – Eckel, Bruce. Prendice Hall PTR

(www.phpprt.com, www.briceeckel.com)

-Just Java – Linden, Peter Van Der. Makron Books, SunSoftPress.

-Core Java 2 – Fundamentals – Horstmann, Cay S.

-Cornell, Gary

The Sun Microsystems Press),

<<http://www.dca.fee.unicamp.br/cursos/PooJava/javaenv/bytecode.html>>

CARACTERISTICAS DA LINGUAGEM JAVA. Disponível em:

<<https://www.devmedia.com.br/conceitos-da-linguagem-java/5341>> Acesso em: 16 Fev. 2018.

DEFINIÇÃO DE ENCAPSULAMENTO. Disponível em:

<<https://www.devmedia.com.br/conceitos-encapsulamento-programacao-orientada-a-objetos/18702>>,<<https://www.devmedia.com.br/abstracao-encapsulamento-e-heranca-pilares-da-poo-em-java/26366>> Acesso em 20 Fev. 2018

DEFINIÇÃO MODELO EM CASCATA. Disponível em:

<<http://www.devmedia.com.br/introducao-ao-modelo-cascata/29843>> - Fonte:

Pressman, R. Engenharia de Software: Uma abordagem Profissional. 7º edição. Editora Bookman. Acesso em 15 Fev. 2018.

DIAGRAMAS COMPORTAMENTAIS. BOOCH, G., RUMBAUGH, J., JACOBSON, I. UML, Guia do Usuário. 2ª Ed., Editora Campus, 2005. Capítulos 1 e 2 Fonte:

Eduardo Figueiredo Referencias Disponível em:

<https://www.google.com.br/url?sa=t&source=web&rct=j&url=http://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/uml-diagramas-comportamento_v01.pdf&ved=2ahUKEwj5m-i-vOjbAhWFj5AKHXxDw4QFjABegQIARAB&usg=AOvVaw2-1WSpxv1zOxNcWGPXZwWv> Acesso em 24 Fev. 2018.

Disponível em: <<https://www.embarcados.com.br/editorial-linguagens-para-sistemas-embarcados/>> Fonte: André Curvello, Cleiton Bueno, Fábio Souza, Henrique Rossi, Rodrigo Pereira e Thiago Lima.

EDITORIAL: LINGUAGENS DE PROGRAMAÇÃO PARA SISTEMAS EMBARCADOS

Especificação da Linguagem UML Versão 1.4, OMG

ESTRUTURADA VERSUS ORIENTAÇÃO A OBJETOS: OS 4 PILARES DA PROGRAMAÇÃO ORIENTADA A OBJETOS. Disponível em:

<<https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>>

EXEMPLO DE ANALISE DE REQUISITO DE HARDWARE Disponível em:

<[https://technet.microsoft.com/pt-br/library/cc263100\(v=office.12\).aspx](https://technet.microsoft.com/pt-br/library/cc263100(v=office.12).aspx)> Acesso em 11 Fev. 2018.

EXEMPLO DE BYTECODES E JVM – DCA/UNICAMP Disponível em:

<<http://www.dca.fee.unicamp.br/cursos/PooJava/javaenv/bytecode.html>> Acesso em: 15 Fev 2018

EXEMPLO DE METODOLOGIA SCRUM. Disponível em: <[scrum desenvolvimentoAgil.com.br](http://desenvolvimentoAgil.com.br)> Acesso em: 14 Fev. 2018

HERANÇA. Disponível em: <<https://www.devmedia.com.br/entendendo-e-aplicando-heranca-em-java/24544>> Acesso em 22 Fev.2018

INTERFACES GRAFICAS JAVA COM CONEXÃO COM BANCO DE DADOS

Disponível em: <Série: interface gráfica java no netbeans>,<Tutorial Java – Java e banco de dados MySql>,<<http://www.dsc.ufcg.edu.br>> Acesso em 01 Mar. 2018.

JAVA PERSISTENCE API – JPA Autor: Nelson Nunes Campos – Disponível em: <<http://fabrica.ms.senac.br/2013/06/java-persistence-api-jpa/>> Acesso em 20 Mai 2018

LEVANTAMENTO DE REQUISITOS Disponível em:

<<http://www.devmedia.com.br/revista-engenharia-de-software-magazine>> Acesso em 10 Fev. 2018.

LEVANTAMENTO DE REQUISITOS E MAPEAMENTO DE PROCESSOS Disponível em: <<https://www.baguete.com.br/artigos/296/ricardo-verissimo/05/11/2007/levantamento-de-requisitos-e-mapeamento-de-processos>> - Fonte – Jones, Carpen - 1997.Applied Software Measurement.

MELO, A. C. **Desenvolvendo aplicações com UML 2.2** – EDITORA: Brasport, 2004.

ORIENTADA A OBJETOS. Disponível em: <<https://www.caelum.com.br/apostila-java-orientacao-objetos/heranca-reescrita-e-polimorfismo/#repetindo-codigo>>, Tutorial Java – Java e banco de dados MySql – Acesso em 16 Fev. 2018

PADRÃO MVC. Disponível em: <<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>> Acesso em 23 Fev. 2018.

PLANEJAMENTO DE PROJETOS. Disponível em:

<<https://www.tiespecialistas.com.br/2011/08/planejamento-de-projetos-como-fazer/>> Acesso em 12 Fev. 2018.

POLIMORFISMO EM JAVA. Disponível em: <xtiuniversity.com> Acesso em: 17 Fev. 2018.

POLIMORFISMO. Disponível em:

<<http://www.dca.fee.unicamp.br/cursos/PooJava/javaenv/bytecode.html>>

PRESSMAN, R.S. **Engenharia de Software Uma abordagem Profissional**. Sétima edição Porto Alegre: AMGH, 2011.

PRESSMAN, R.S. **Engenharia de Software**. Sexta edição McGrawHill, 2006.

Referências: Embedded Systems Programming Languages, Índice TIOBE

referências do artigo - BOOCH, G., RUMBAUGH, J., JACOBSON, I. UML, Guia do Usuário. 2ª Ed., Editora Campus, 2005. Capítulos 1 e 2) Disponível em:

<http://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/uml-diagramas-comportamento_v01.pdf>

REQUISITOS DE INFRAESTRUTURA E SEGURANÇA Disponível em:

<<https://pt.linkedin.com/pulse/projetos-de-infraestrutura-ti-pre-requisitos-vinicius-camoes>> <<https://gestaoemti.com/2016/06/22/requisitos-de-infraestrutura-de-ti/>> Acesso em 11 Fev. 2018.

REQUISITOS DE USABILIDADE. Disponível em:

<http://www.funpar.ufpr.br:8080/rup/process/workflow/requiem/co_req.htm> Acesso em 13 Fev. 2018

SCRUM Disponível em <<http://www.mindmaster.com.br/scrum/>> ,

<<https://www.significados.com.br/scrum/>> ,

<<http://www.jrmcoaching.com.br/blog/como-funciona-a-metodologia-scrum/>> Acesso em: 14 Fev. 2018.

SILBERSCHATZ, A.; KORTH, H F.; SUDARSHAN. S. **Sistemas de banco de dados**. Rio de Janeiro: 6ª ed., Elsevier, 2012.

SISTEMA GERENCIADOR DE BANCO DE DADOS. Disponível em:
<<https://docs.microsoft.com/pt-br/sql/sql-server/install/hardware-and-software-requirements-for-installing-sql-server>> Acesso em 20 Fev. 2018.
Software Architecture#and the UML, Grady Booch (Seminário)

SOMMERVILLE, I. **Engenharia de Software**. São Paulo: Pearson, 2008.

UML Distilled Applying the Standard Object Modeling Language, Martin Fowler Curso on-line da TogetherSoft,

UML Material sobre UML – Prof Ms. Marcus Costa Sampaio Disponível em:
<<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/classes/classes1.htm> > REFERENCIAS: Inside the Unified Modeling Language, Material da Rational

UML: DIAGRAMAS COMPORTAMENTAIS – Autor: Eduardo Figueiredo
(REFERENCIAS: Inside the Unified Modeling Language, Material da Rational
UML Distilled Applying the Standard Object Modeling Language, Martin Fowler
Curso on-line da TogetherSoft, www.togethersoft.com/services/practical_guides/umlonlinecourse/
Especificação da Linguagem UML Versão 1.4, OMG
Software Architecture#and the UML, Grady Booch (Seminário))