

**Unip – Universidade Paulista**

**Ciências Da Computação (CC)**

**DESENVOLVIMENTO DE UM SISTEMA COMPUTACIONAL PARA ANÁLISE E CLASSIFICAÇÃO DE FORMAS.**

**Atividades Práticas Supervisionadas – 3º SEMESTRES – CC**

**Curso: Ciência da Computação**

**Lucas Campanuchi Corrêa – RA: F0359E-1 – CC3Q18**

**Vinicius Alves Panobianco – RA:** **N485HJ-6 – CC3Q18**

**Wesley Marcos Morais dos Santos– RA:** **N3855J-3 – CC3P18**

**Wellington Fagundes de Oliveira – RA: N42916-0 – CC3P18**

**Sumário**

[Objetivo do trabalho 3](file:///C:\Users\Joyc\Desktop\APS\APS%203%20semestre.docx#_Toc25924830)

[Introdução 4](file:///C:\Users\Joyc\Desktop\APS\APS%203%20semestre.docx#_Toc25924831)

[Referencial teórico 7](file:///C:\Users\Joyc\Desktop\APS\APS%203%20semestre.docx#_Toc25924832)

[Plano de desenvolvimento e aplicação 11](file:///C:\Users\Joyc\Desktop\APS\APS%203%20semestre.docx#_Toc25924833)

[Estrutura do Programa](file:///C:\Users\Joyc\Desktop\APS\APS%203%20semestre.docx#_Toc25924834) 19

[Relatório com linhas de código do Programa](file:///C:\Users\Joyc\Desktop\APS\APS%203%20semestre.docx#_Toc25924835) 32

[Bibliografia](file:///C:\Users\Joyc\Desktop\APS\APS%203%20semestre.docx#_Toc25924836) 43

**Objetivo do Tabalho**

Com os avanços tecnologicos nos anos 80, decorrido dos investimentos pós-Segunda Guerra, processos complexos em empresas e negocios do ramo vendas, como controle de: pedidos de cliente, fornecedores, estoque disponivel, entre outro. Foram gradativamente migrando de arquivos fisicos pra virtuais ao longo do tempo, possibilitando uma gestão mais rapida e pratica.

Com o desvolvimento tecnologico cada vez mais acelerado nos dias atuais, e com a alta competitividade no mercado consumidor entre empresas decorrido desses avanços tecnologicos, manter um estoque organizado é um meio extremamente importante pra qualquer empresa, para que a qual possa constatar a entrada e saida de mercadorias, demanda de certo produto, e o lucro liquido final.

“Além de ajudar no gerenciamento da empresa, o controle de estoque oferece uma maior eficiência e organização de alguns dos processos internos de sua empresa. Ele também diminui desperdícios, tanto de tempo e energia na hora de fazer gestão quanto de seus recursos financeiros.” Fonte:(https://blog.egestor.com.br/o-que-e-controle-de-estoque/)

Hoje possuir um sistema de controle de estoque adequado, em uma empresa ou negócio é estremamente vital para a qual possa competir e se manter, com relação às demais concorentes.

O objetivo deste trabalho é desenvolver com uso da linguagem Java, de um programa para controle de estoque possuindo cadastro de usuário, cadastro de produtos com código de barras, controlar estoque mínimo de produto, cadastro de fornecedor, entrada do produto de acordo com o fornecedor, controlar saída de produto e categorizar os produtos.

Nós da LVW² Soluções, apartir dessa ideia desenvolvemos um programa para controle de estoque nomeado como Afrodite, o tem a finalidade de que qualquer pessoa possa utilizá-lo, pois foi construido visando ser intuitivo, de fácil manuseio, e que possa abrangir qualquer empresa ou negócio do ramo de vendas.

**Introdução**

Para o desenvolvimento desse programa de controle de estoque, utilizamos a linguagem de programação Java.

Que é o Java:

“Java é uma linguagem de programação e plataforma computacional lançada pela primeira vez pela Sun Microsystems em 1995. Existem muitas aplicações e sites que não funcionarão, a menos que você tenha o Java instalado, e mais desses são criados todos os dias. O Java é rápido, seguro e confiável. De laptops a datacenters, consoles de games a supercomputadores científicos, telefones celulares à Internet, o Java está em todos os lugares!”

Fonte: (https://www.java.com/pt\_BR/download/faq/whatis\_java.xml)

Java é uma linguagem de programação de alto nível e orientada a objeto, que possui uma alta compatibilidade qualquer sistema operacional que suporte a JVM (Java Virtual Machine) como Windows, Mac, Linux, Android, televisores, relógios, entre muitos outros.

Sua história:

O início do desenvolvimento do Java se da em 1991, pelo grupo Sun Microsystems, com o objetivo de possibilitar a criação de programas que pudessem ser executados em qualquer dispositivo, Mas para tal, precisariam desenvolver um sistema que se possibilitasse essa compatibilidade.

Assim nasceu linguagem Oak e sistema intitulado GreenOS que anos depois se tornariam Java e JVM(Java Virtual Machine),tal mudança do nome de Oak para Java deu-se de uma homenagem para uma ilha norte-Americana que importava o café consumido pelos desenvolvedores.

Em 1996, a Sun Microsystem disponibilizou gratuitamente para sua comunidade o JDK(Java Developer`s Kit) que é um kit para desenvolvimento de software, e ao longo dos anos foi disponibilizado diversas atualizações e ferramentas gratuitas pela Sun, tornando o Java uma das linguagens de programação mais utilizadas durante muitos anos.

Fonte: (https://tableless.com.br/java-origem/)

O que é uma JVM:

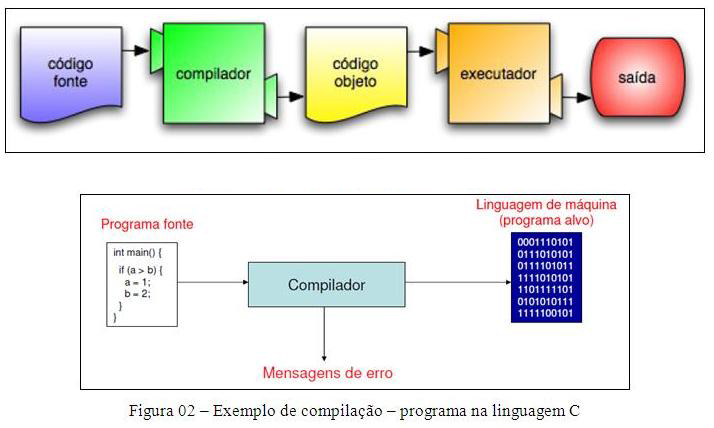
Pra quem já esta um tempo no ambiente de programação, já deve ter ouvido a frase “Java roda em qualquer lugar” e por mais espantoso que possa parecer, isso realmente acontece devido a sua Maquina Virtual.

“Antes de falarmos sobre a JVM, vamos pensar no conceito de máquina virtual. Uma máquina virtual é um software que simula uma máquina física e consegue executar vários programas, gerenciar processos, memória e arquivos. Resumindo, ele constitui de uma plataforma, onde a memória, o processador e seus outros recursos, são totalmente virtuais, não dependendo de hardwares.”

Fonte: (https://tableless.com.br/java-origem/)

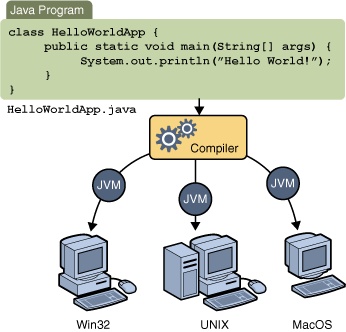
Linguagem de programação como C, C++ são linguagens compiladas, o código é escrito em uma linguagem alto nível, ou seja, compreensível ao programador e é traduzido por um compilador para uma linguagem de maquina (baixo nível).

Códigos escritos em linguagem compilada só podem ser executados no sistema operacional o qual foi projetado, caso contraria o código deve ser reajustado e recompilado para esse outro sistema.



Fonte: (https://www.computersciencemaster.com.br/2016/08/aula-02-configurando-o-ambiente-java.html?m=1)

Já a interpretada, o código fonte é executado por um interpretador que logo a pós é executado pelo sistema operacional. No caso do Java, o sistema operacional é a maquina virtual, recebe o código e excuta em uma linguagem de maquina.



Fonte: (https://www.devmedia.com.br/processo-de-interpretacao-e-compilacao-entendendo-o-java-de-uma-forma-diferente/24257)

Programação Orientada a Objeto (POO):

Orientação a objeto tem como conceito classificar, organizar e abstrair coisas, tendo como objetivo de tentar aproximar o mundo real com o mundo virtual, classificando todos os elementos inseridos como objetos, ordenando suas funções e quais objetos se comunicaram um com o outro.

Os quatro pilares da orientação a objeto:

Encapsulamento: Esconder partes da implementação do programa ao usuário, tornando essas partes independentes das demais ao sistema.

Herança: Permite que classes herdem métodos e atributos de outras classes, desse que elas possuam as mesmas restrições, exemplo: public ou protected.

Abstração: É focalizar nos aspectos essenciais de uma identidade, de modo filtrar informações relevantes ao que se esta desenvolvendo.

Polimorfismo: Permite que classes derivadas de uma classe base possam mesmo possuindo a mesma assinatura, utilizar métodos iguais para assoeis diferentes.

**Referencial Teórico**

“Tecnologia Java nasceu com um objetivo em mente, foi lançado com outro, mas, no final, decolou mesmo no desenvolvimento de aplicações do lado do servidor. Sorte? Há hoje o Java FX, tentando dar força para o Java não só no desktop, mas como aplicações ricas na web, mas muitos não acreditam que haja espaço para tal, considerando o destino de tecnologias como Adobe Flex e Microsoft Silverlight.”

(Java e Orientação a Objetos, Caelum, pg.17).

A ideia primordial da Sun Microsystems para o que posteriormente viria a se tornar o Java, seria de um interpretador que possibilitaria a reescrita de software para aparelhos eletrônicos, à partir disso surgiu-se um projeto para desenvolvimento de um dispositivo que faria o controle desses aparelhos remotamente, mas não foi bem aceito pelas empresas na época, entretanto com o avanço da web, a Sun viu uma oportunidade de utilizar a ideia de seu projeto para rodar aplicações em browser.

“Já o Java utiliza do conceito de máquina virtual, onde existe, entre o sistema operacional e a aplicação, uma camada extra responsável por "traduzir" - mas não apenas isso - o que sua aplicação deseja fazer para as respectivas chamadas do sistema operacional onde ela está rodando no momento:

Já o Java utiliza do conceito de máquina virtual, onde existe, entre o sistema operacional e a aplicação, uma camada extra responsável por "traduzir" - mas não apenas isso - o que sua aplicação deseja fazer para as respectivas chamadas do sistema operacional onde ela está rodando no momento:”.

(Java e Orientação a Objetos, Caelum, pg.18).

Graças a Java Virtual Machine (JVM) que uma de sua varias funções é tradução do código gerado pelo programador para um código compreensível para a maquina o qual está instalado, possibilitando as aplicações feitas em Java “rodar” em qualquer sistema que tenho uma JVM.

“O Java vem com centenas de classes predefinidas, Você não terá que reinventar a roda se souber como encontrar o que precisa na biblioteca Java, normalmente conhecida como API Java. Há coisas melhores a fazer, Se você pretende escrever códigos, pode escrever somente as partes que forem exclusivas de seu aplicativo. Abiblioteca Java consiste em uma pilha gigante de classes apenas esperando para serem usadas como blocos de construção.”

(Use a cabeça Java, Kathy Sierra e Bert Bates, pg 11).

Java possui muitos acessórios que facilita o desenvolvimento de programas como a Java Development Kit (JDK) que é uma ferramenta que tem afinal idade de distribuir os recursos necessários para desenvolver uma aplicação Java, como a JVM, bibliotecas e o compilador.

“Orientação a objetos vai te ajudar em muito em se organizar e escrever menos, além de concentrar as responsabilidades nos pontos certos, flexibilizando sua aplicação, encapsulando a lógica de negócios. Outra enorme vantagem, onde você realmente vai economizar montanhas de código, é o polimorfismo das referências, que veremos em um posterior capítulo.”

(Java e Orientação a Objetos, Caelum, pg.50).

A orientação a objeto é extremamente relevante pelo seu conceito de possibilitar projetar e programar um software utilizando representação de algo da vida real por meio de objetos utilizando seus quatro pilares: encapsulamento, herança, polimorfismo e abstração. Em Java a orientação a objeto é vital, pois tudo é considerado um objeto.

“Sempre que estamos trabalhando com o desenvolvimento de uma aplicação queremos ser o mais produtivos possível, e não é diferente com uma aplicação web. Uma das formas de aumentar a produtividade do desenvolvedor é utilizar uma ferramenta que auxilie no desenvolvimento e o torne mais ágil, no nosso caso, uma IDE.” (Java para desenvolvimento web, Caelum, pg.49).

Um de seus pontos fortes na linguagem Java está na facilidade de se encontrar erros no programa desenvolvido, justamente por causa das IDEs que o auxiliam na identificação de erros, capacidade de compilar bibliotecas, debuggers, entre outros.

“Dentro de um bloco, podemos declarar variáveis e usá-las. Em Java, toda variável tem um tipo que não pode ser mudado.”

(Java para desenvolvimento web, Caelum, pg.30).

Variáveis são um dos recursos mais básicos e importantes das linguagens de programação, utilizadas para armazenar valores em memória, elas nos permitem gravar e ler esses dados com facilidade a partir de um nome definido por nós.

“As estruturas de repetição também são conhecidas como laços (loops) e são utilizados para executar, repetidamente, uma instrução ou bloco de instrução enquanto determinada condição estiver sendo satisfeita.”

Fonte:(<https://tableless.com.br/java-estruturas-de-repeticao/>).

Utilização de laços permite executar o mesmo bloco de código enquanto uma condição é atendida. Sendo os laços For e While usados para otimizar e facilitar o desenvolvimento de um programa, por continuar a executar um código de acordo com a condição estabelecida, como While que executa uma linha de código enquanto é considerado verdadeiro. Enquanto o laço For nos permite ler os itens de um código, e para cada um deles, executar a linha de código referente ao laço.

“A declaração de uma classe em Java é bastante simples. Utilizamos a palavra reservada class seguida pelo nome da classe. Logo após, entre chaves, definimos os elementos a ela relacionados: atributos, construtores e métodos.”

Fonte:(<https://www.devmedia.com.br/java-declaracao-e-utilizacao-de-classes/38374>)

Classes em Java são elementos que são utilizados para representar objetos que remetem ao mundo real, possuindo características e comportamentos que são os atributos e métodos. Para que seja possível construir uma classe é necessário utilizar um dos pilares da orientação a objeto (OO), que é o da abstração.

“Muitos sistemas precisam manter as informações com as quais eles trabalham para permitir consultas futuras, geração de relatórios ou possíveis alterações nas informações. Para que esses dados sejam mantidos para sempre, esses sistemas geralmente guardam essas informações em um banco de dados, que as mantém de forma organizada e prontas para consultas.”

(Java para desenvolvimento web, Caelum, pg.14).

Aplicações Java para que tenha acesso a um banco de dados precisam ter o controle sobre o driver JDBC que será utilizado, criar uma conexão com a aplicação estável entre cliente e servidor, modelagem e execução do SQL.

“A conexão a um banco de dados é feita de maneira elegante com Java. Para evitar que cada banco tenha a sua própria API e um conjunto de classes e métodos, temos um único conjunto de interfaces muito bem definidas que devem ser implementadas. Esse conjunto de interfaces fica dentro do pacote java.sql e nos referiremos a ele como JDBC.”

(Java para desenvolvimento web, Caelum, pg.15).

JDBC (Java Database Connectivity) é uma API que reúne grupos de classes e interfaces em Java possibilitando se conectar através de um driver específico do banco de dados, Possibilitando executar instruções SQL de qualquer tipo de banco de dados relaciona através desse Driver.

**Plano de desenvolvimento e aplicação**

O programa foi produzido utilizando os recursos do Java traves do Neatbeans juntamente com um banco feito no PostgreSQL em SQL, os quais vão ser detalhados abaixo.

Para o planejamento de como será desenvolvido o programa, criamos o modelo entidade e relacionamento (MER) e um diagrama entidade e relacionamento (DER).

Diagrama entidade e relacionamento (DER) é um modelo de como será desenvolvida a relação entre as entidades do sistema.

Entidades: são considerados como os objetos físicos ou lógicos de um domínio, são ligadas uma a outra por relacionamentos e possuem atributos.

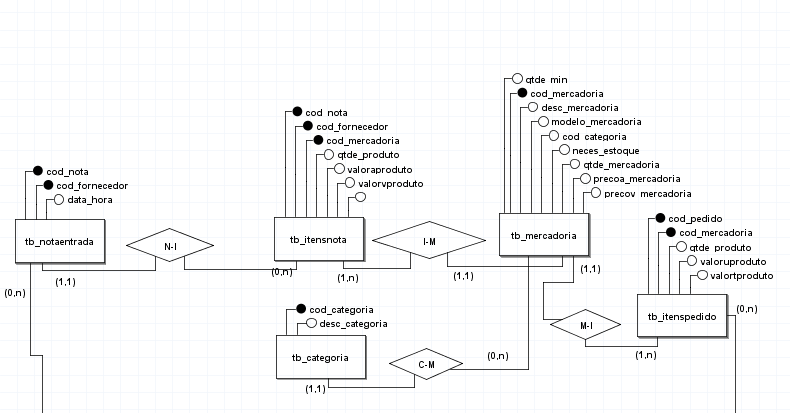
Relacionamentos: ao se identificar uma entidade deve se atribuir a ela um relacionamento de acordo com a quantidade de objetos envolvidos, podem possuir relacionamentos 1 para 1, 1 para “n”(muitos) ou n para “n”(muitos para muitos).

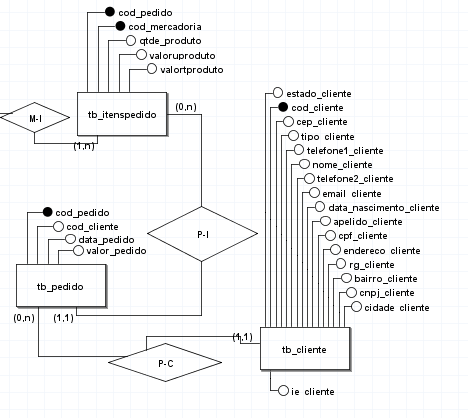
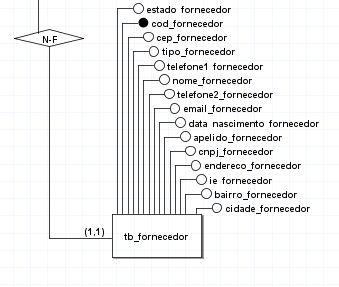
Atributos: são características que descrevem as características das entidades.

Chave estrangeira (FK): chave de uma tabela que cria um relacionamento uma chave primária de outra, possui um valor que pode ser repetido.

Chave primaria (PK): Chave exclusiva dessa tabela, seu valor não pode ser repetido ou ser nulo.

Para a representação do nosso Diagrama utilizamos a ferramenta Case: BrModelo





Modelo entidade e relacionamento (MER) é um modelo que descreve os objetos (entidades) envolvidos em um domínio, com seus atributos e como elas se relacionam.

“Cada tipo de dado possui uma representação externa determinada pelas suas funções de entrada e de saída. Muitos tipos nativos possuem formato externo óbvio. Entretanto, muitos tipos existem apenas no PostgreSQL, como os caminhos geométricos, ou possuem várias possibilidades para o formato, como os tipos de data e hora. Algumas das funções de entrada e saída não são inversíveis, ou seja, o resultado da função de saída pode perder precisão quando comparado com a entrada original.”

Fonte: (<http://pgdocptbr.sourceforge.net/pg80/datatype.html>)

Um Sistema de Gerenciamento de Banco de Dados (SGBD) é um software responsável pelo gerenciamento de uma base de dados. Seu objetivo é retirar da aplicação a responsabilidade de gerenciar o acesso, a manipulação e a organização dos dados.

Datasource é uma configuração de conexão para um banco de dados de um servidor.

“Assim, o principal objetivo do padrão Data Access Object (DAO) é encapsular o acesso ao data source fornecendo uma interface para que as diversas outras camadas da aplicação possam se comunicar com o data source.”

Fonte:(<https://www.devmedia.com.br/implementando-o-data-access-object-no-java-ee/33339>)

Tipos de dados da cadeia de caracteres:

CHAR: Sequência de caracteres de comprimento fixo, capaz de suporta até 8000 caracteres.

VACHAR: Sequência caracteres de comprimento variável, capaz de suporta até 8000 caracteres.

INTEGER: é referenciado como tipos numéricos exatos mais usual por suportar até 4 bytes.

DATE: é referenciada com data de calendário como dia, mês e ano.

INSERT: permite inserir valores numa tabela.

“O comando CREATE TABLE se utiliza para criar uma tabela chamada nom\_sql com os campos especificados ao passar uma ou mais definição\_coluna e/ou restrição\_tabela. Se passada a restrição IF NOT EXISTS, a tabela só é criada quando não existe uma tabela com o mesmo nome na base. Caso contrário, não se é criado e não é gerado nenhum erro.”

Fonte: (<https://doc.4d.com/4Dv15/4D/15/CREATE-TABLE.300-2288142.pt.html>)

CREATE TABLE tb\_categoria

(

cod\_categoria serial NOT NULL,

desc\_categoria character varying(40) NOT NULL,

CONSTRAINT pk\_tb\_categoria\_cod\_categoria PRIMARY KEY (cod\_categoria),

CONSTRAINT tb\_categoria\_desc\_categoria\_key UNIQUE (desc\_categoria)

);

Constraint podem colocar restrições para limitar o tipo de dados a introduzir numa tabela:

NOT NULL: garante que uma coluna não pode ter o valor NULL

DEFAULT: Da um valor padrão para uma coluna quando nenhum é especificado

PRIMARY KEY: Utiliza para identificar de forma única uma linha na tabela

FOREIGN KEY: Utiliza para garantir a integridade referencial dos dados

CREATE TABLE tb\_mercadoria

(

cod\_mercadoria serial NOT NULL,

desc\_mercadoria character varying(60) NOT NULL,

modelo\_mercadoria character varying(60),

cod\_categoria integer NOT NULL,

neces\_estoque character varying(20),

qtde\_mercadoria integer,

precoa\_mercadoria double precision,

precov\_mercadoria double precision,

qtde\_min integer,

CONSTRAINT pk\_tb\_mercadoria\_cod\_mercadoria PRIMARY KEY (cod\_mercadoria),

CONSTRAINT fk\_tb\_mercaoria\_cod\_categoria FOREIGN KEY (cod\_categoria)

REFERENCES tb\_categoria (cod\_categoria) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION

);

Modelo Entidade e relacionamento referente ao projeto:

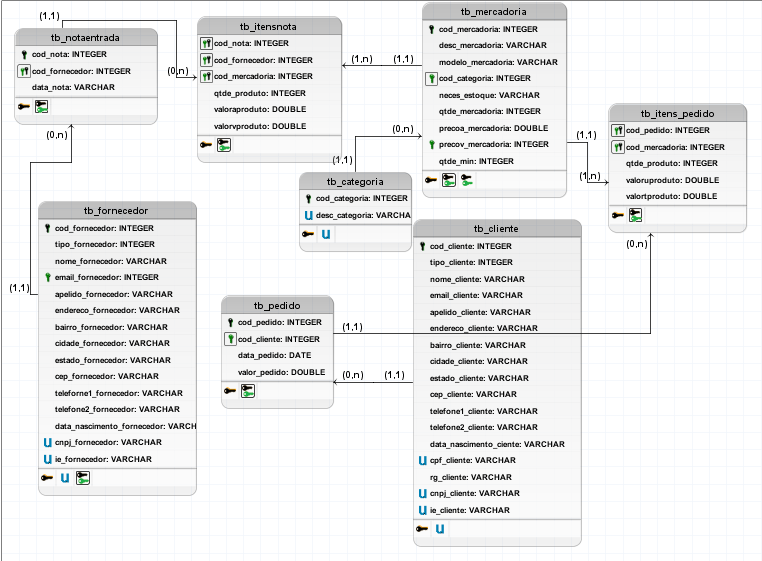
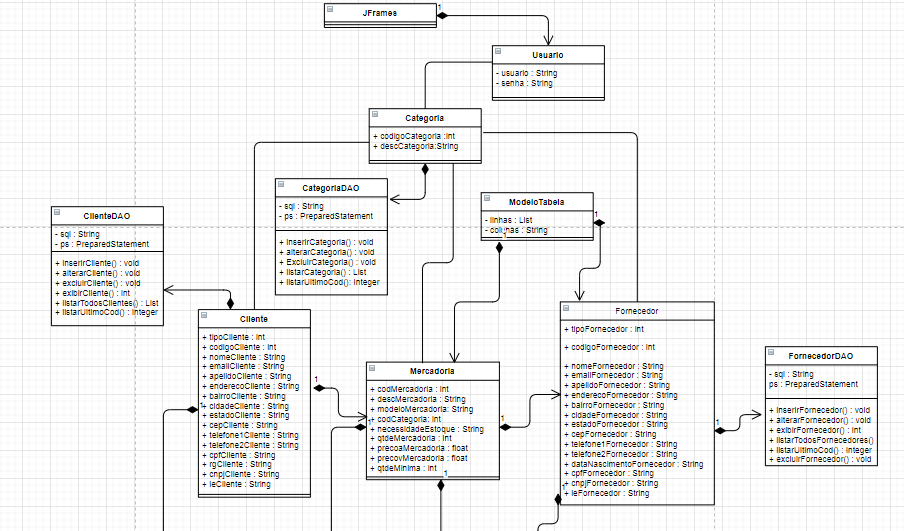
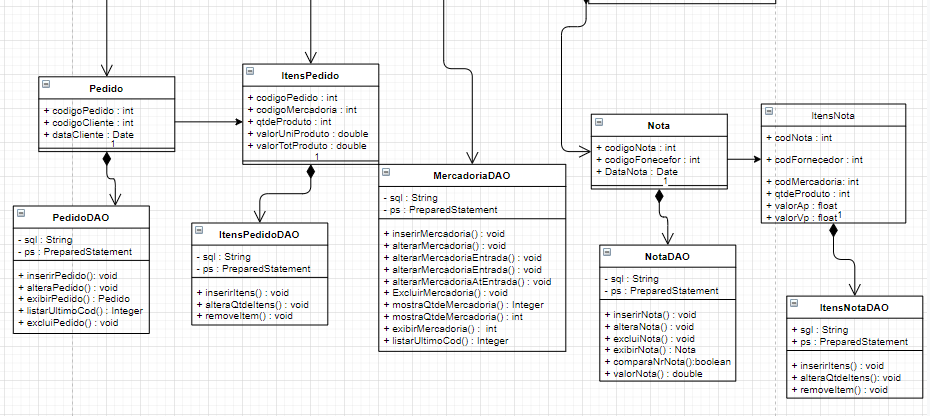


Diagrama de Classes (UML):





Desenvolvimento do programa na linguagem Java:

Variáveis: é um método de armazenar um determinado valorou para receber um valor, e podem ser utilizadas com durante o código. As Variáveis têm vários tipos como: Int, string, float, boolean entre outros.

- Int: é um tipo de dado que recebe valores numéricos inteiros.

- String: é um tipo de dado que recebe conjunto de caracteres que são True ou False colocados em uma determinada ordem.

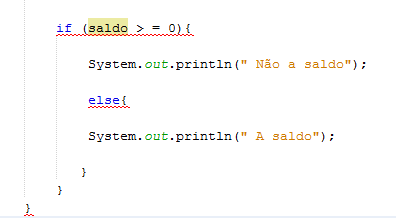
- Boolean: é um tipo de dado que recebe valor de Verdadeiro ou Falso.

- Float: é um tipo de dado que recebe valores numéricos reais.

As estruturas condicionais em Java têm por finalidade tomar uma decisão de acordo com o resultado de uma condição especificada:

IF: é um método de criar condições para que possam retornar verdadeiro ou falso, baseado na condição citada no código.

Else: será ativado caso não IF retorne falso.

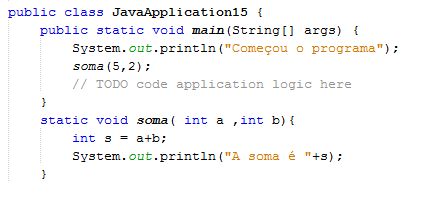


Classe é um elemento do código Java que utilizamos para representar objetos do mundo real. Podemos declarar atributos e métodos, que representam as características e comportamentos desse objeto.

Os objetos são características definidas pelas classes. Neles é permitido instanciar objetos da classe para inicializar os atributos e invocar os métodos.

Atributos são as características de um objeto, essas características também são conhecidas como variáveis.

Métodos são as ações que os objetos podem exercer quando solicitados, onde podem interagir e se comunicarem com outros objetos.



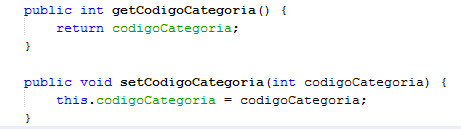
Um Array é uma estrutura de dados que armazena uma coleção sequencial de tamanho fixo de elementos do mesmo tipo, Em Java, todos os arrays são alocados dinamicamente.

Exemplo:

double[] salarios;

double salarios[];

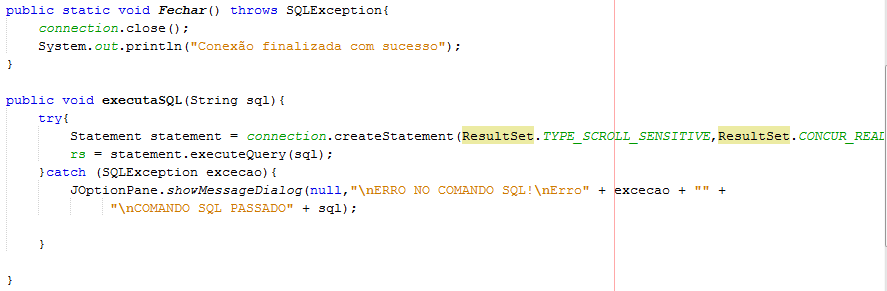
Get e Set são que métodos que servem para pegarmos informações de variáveis da classe que são definidas como private, porém esses método são definidos como 'public'.



Uma exceção (Exception) em Java é um objeto da classe java.lang.Exception, que capaz de armazenar dados nas suas variáveis de instância. Quando um erro ou situação anormal é encontrado durante a execução de um método.

Para tratar as exceções em Java são utilizados os comandos try e catch:

Em “try” são introduzidas todas as linhas de código que podem vir a lançar uma exceção, para que caso ocorra algum problema com os comandos dentro do bloco, a execução acionara os blocos “catch” onde é esta escrita à ação que será tomada.



O código vai tentar(try) certas linhas de comando, se não funcionar ele irá usar uma exceção (catch).

**Projeto do programa**

Nosso projeto tem uma tela de login de usuário simples para acessar as funcionalidades do projeto. No momento existe um único cadastro do usuário com todas as permissões de manutenção do software.

E nossa tela de login colocamos o logo de nossa empresa representada e os campos de login de usuário e a senha que respectivamente são “admin” e “123” por padrão;

Fig.(1).

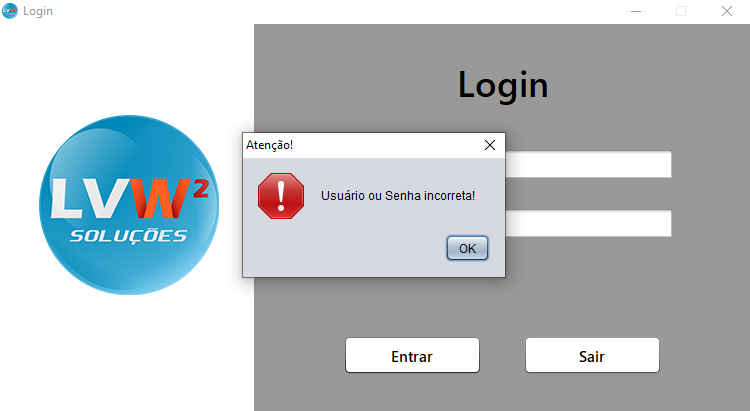
Caso essas informações sejam inseridas erradas o sistema informará com a seguinte mensagem representada na figura abaixo na Figura 2:

Fig.(2).

Com as informações inseridas corretamente, o login é executado e aparecerá uma tela de login como representado na figura 3;

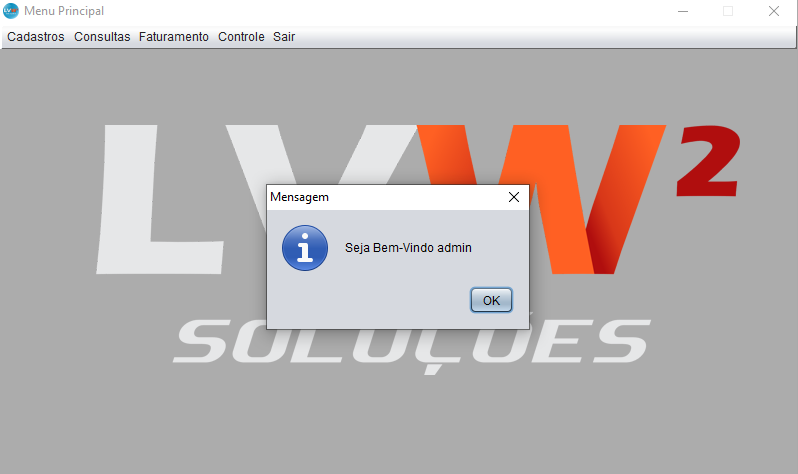


Fig.(3).

Na janela principal temos uma barra de menu separando por categorias as funcionalidades de nosso software, essas categorias são: cadastros, consultas, faturamento, controle e a funcionalidade de “Sair” expressamente representada por uma categoria para facilitar ao usuário finalizar o programa como representado na figura 4;



Fig.(4).

Mostraremos as funcionalidades de cada categoria que estão em submenus dos menus de categorias principais, começando pela categoria de cadastro.

Podemos cadastrar categorias, clientes, fornecedor e produtos como representado na figura 5.

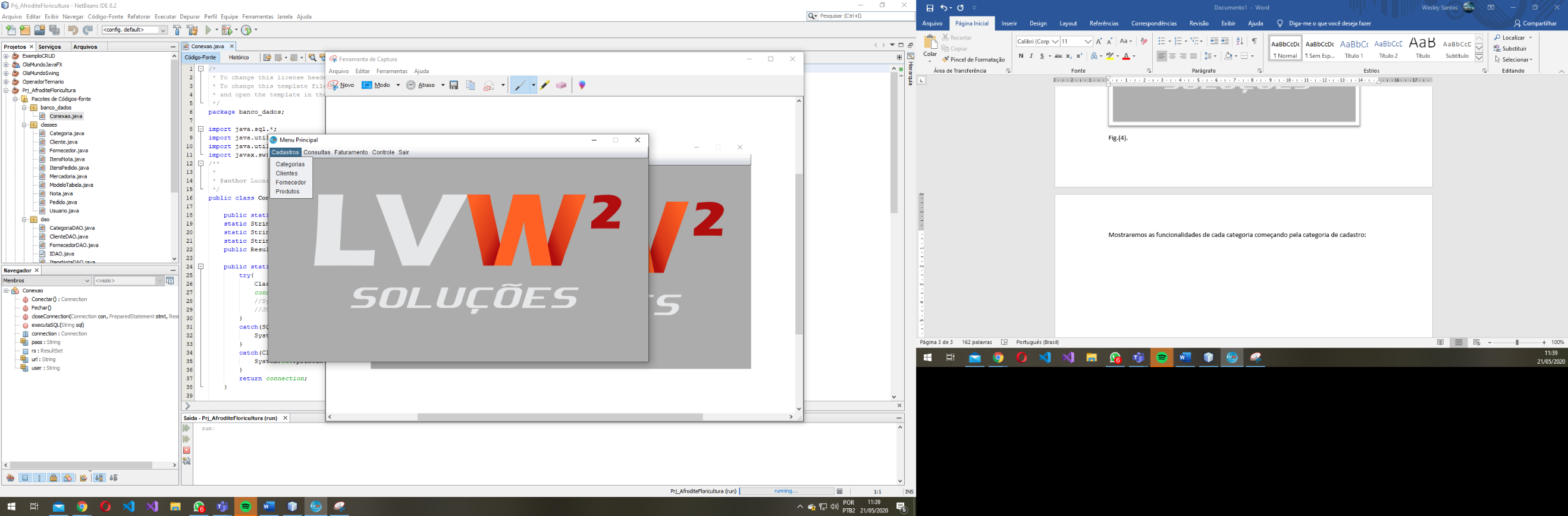


Fig.(5).

Na tela de cadastro de categorias inserimos as categorias dos produtos com o seu respectivo nome. Também podemos acessar novas categorias, editar, excluir e limpar todos os campos para inserir novas categorias ou pesquisar, como representado na figura 6.

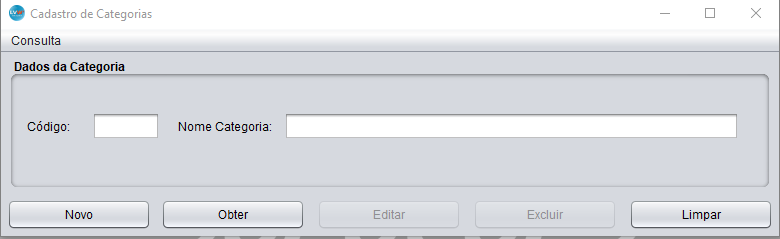


Fig.(6).

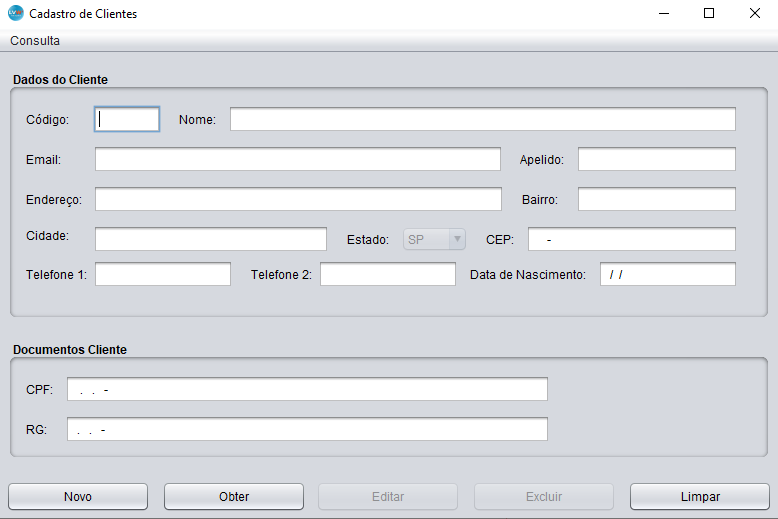
Na tela de cadastro de clientes, além do cadastro, podemos pesquisar clientes já cadastrados, editar cadastros, bem como excluir. Todos os clientes cadastrados tem um código de ID além de suas informações pessoais para facilitar o acesso aos seus dados como representado na figura 7;

Fig.(7).

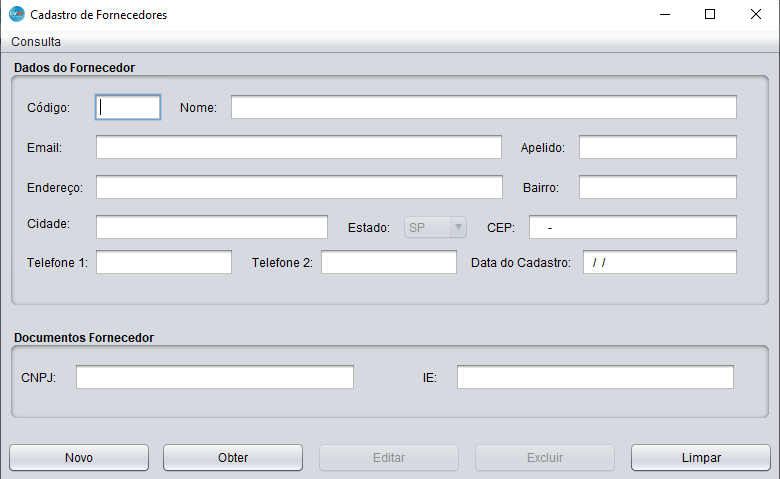
No cadastro de fornecedores temos as mesmas funcionalidades de pesquisa, edição, exclusão e cadastro de clientes. Bem como inserir vários atributos para cada cadastro como representado na figura 8.

Fig.(8).

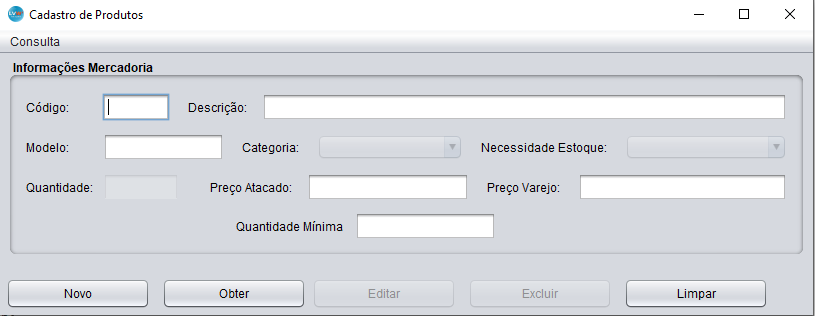
Na janela de cadastro de produtos colocamos as descrições do produto, sua categoria que já cadastramos anteriormente, preço de atacado, preço de venda, quantidade mínima para controle de estoque. Como padrão mantivemos as mesmas funcionalidades de pesquisa de produtos, edição e exclusão como representado na figura 9.

Fig.(9).

Agora iremos navegar no submenu de consultas onde mostraremos tudo o que podemos consultar em nosso software que inclui clientes, fornecedores, notas, produtos e pedidos.

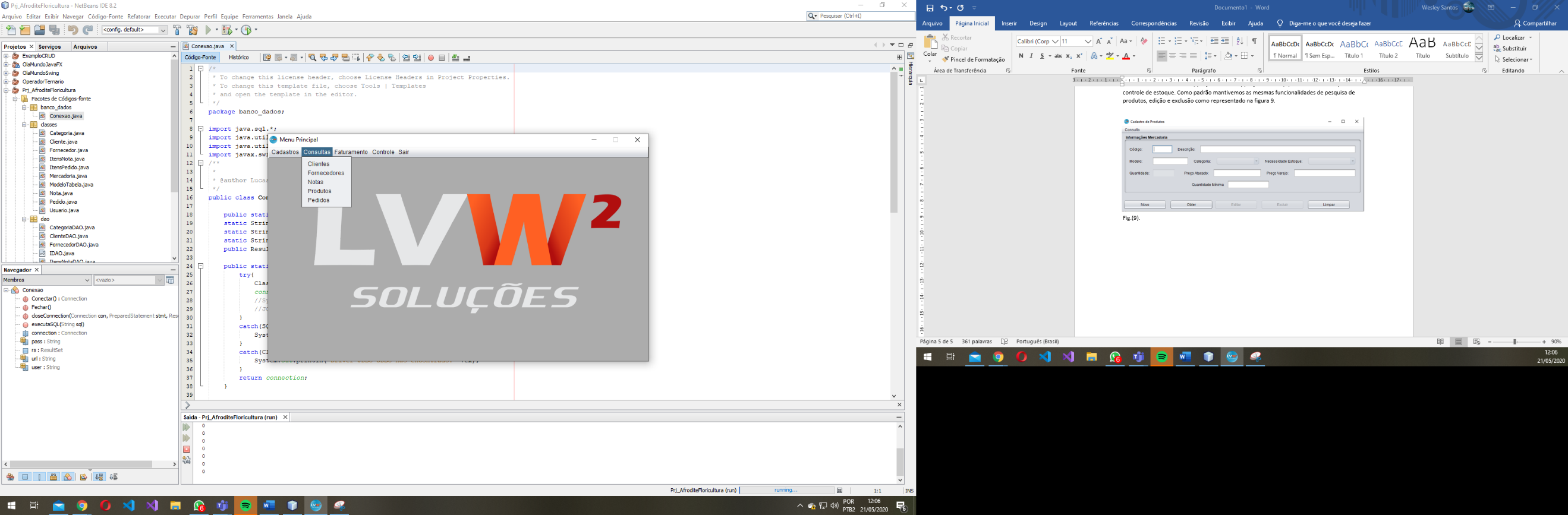
Representaremos na figura 10.

Fig.(10).

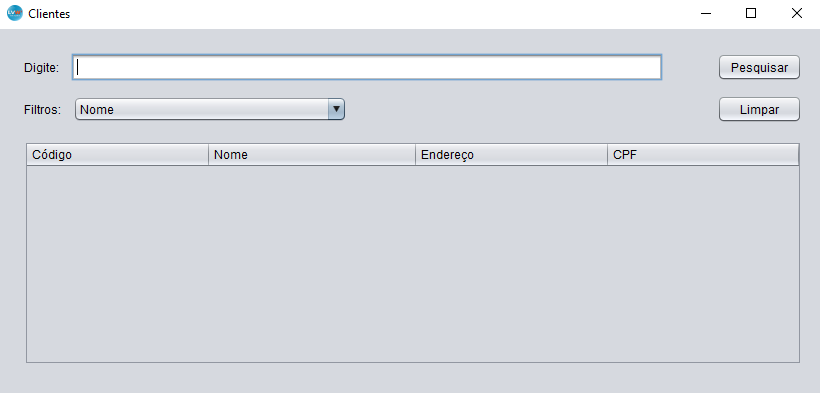
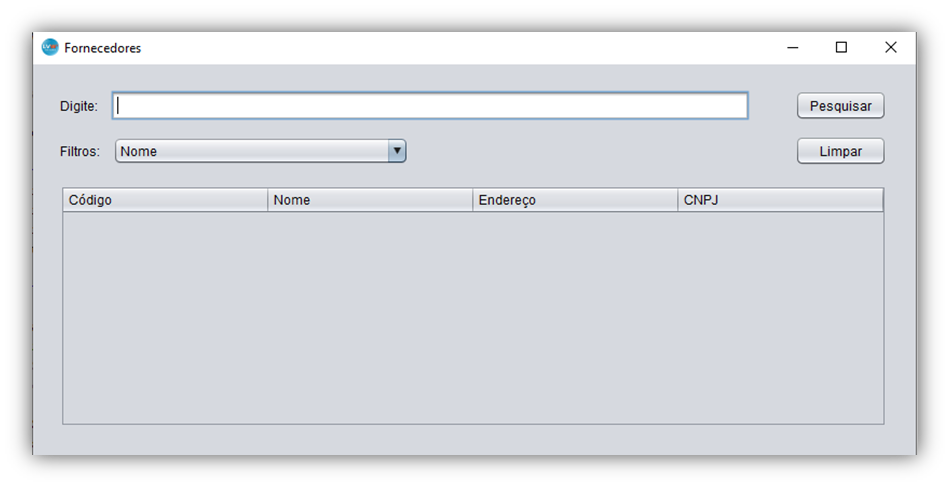
Nas consultar de Clientes e fornecedores utilizamos janelas parecidas pelos cadastros conterem informações parecidas assim como informações que precisaríamos para suas consultas. Representamos consultas de clientes e fornecedores respectivamente nas figuras 11 e 12 abaixo;

Fig.(11).

Fig.(12)



As consultas de notas e pedidos também utilizamos o mesmo layout com a diferença do campo de pesquisa de código de cliente e código do fornecedor. As pesquisas são feitas por intervalo de datas e mostram o valor todas da somatória das notas nesse período.

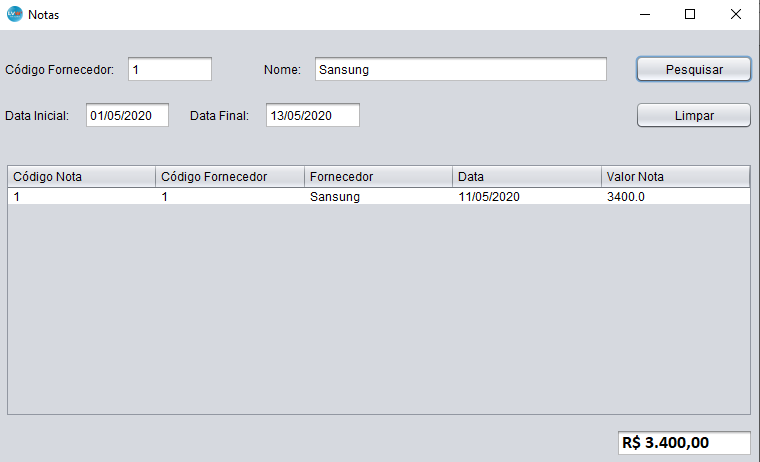
Consulta de notas figura 13.

Fig.(13).

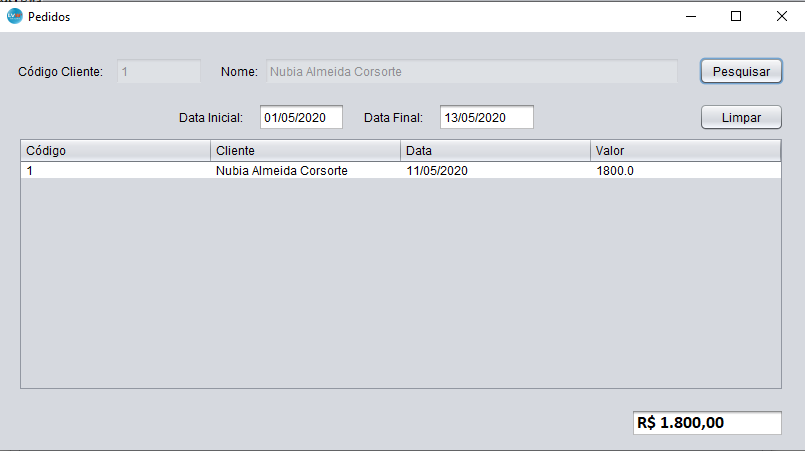
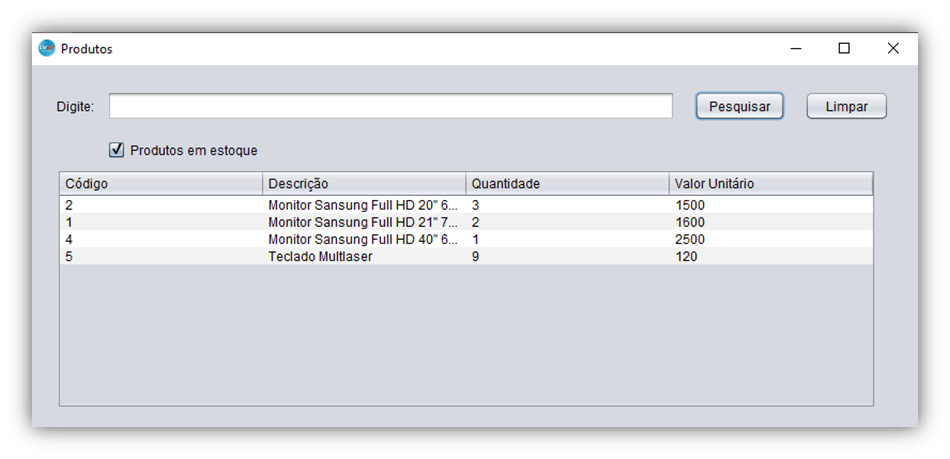
Consulta de pedido, figura 14.

Fig.(14).



Na consulta de produtos colocamos uma caixa de seleção para separar consultar de itens que contém no estoque com itens que ainda não foram adicionados no estoque ou que acabaram para facilitar a consulta e controle, como representado na figura 15.

Fig.(15).

Na categoria faturamento adicionamos dois submenus para futuras implementações em faturamentos de notas, no momento esses submenus separam as entradas e saídas de mercadorias do sistema, sendo entradas de mercadoria dando acesso à notas e saída de mercadorias dando acesso a pedidos como mostrado nas figuras 16 e 17.

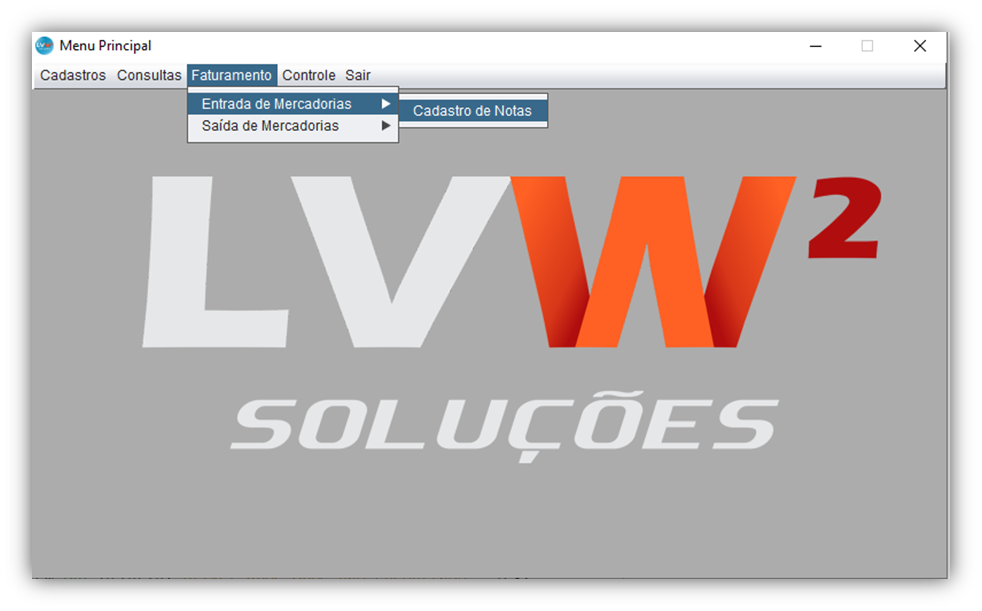


Fig.(16).

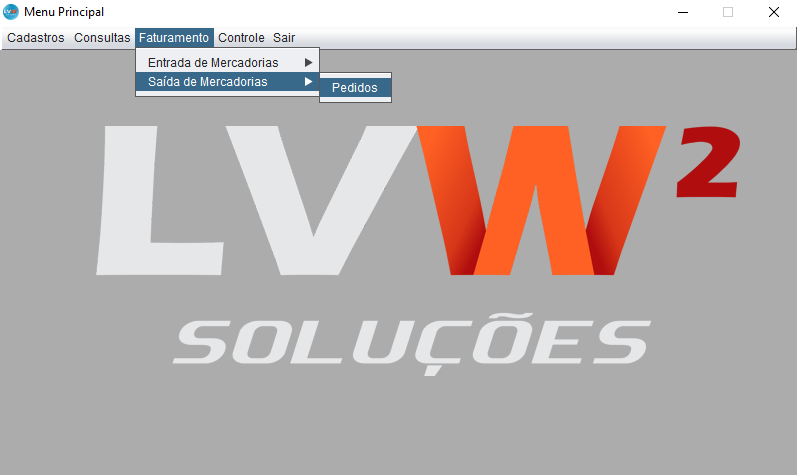


Fig.(17).

Na tela de notas de entrada inserimos mercadorias no sistema, nessa janela também podemos consultar notas já cadastradas para evitarmos erros de usabilidade. Podemos também fazer a exclusões a edições de notas. Na tabela de pesquisa vemos a somatória e os detalhes principais dos itens de cada nota como representado na figura 18.

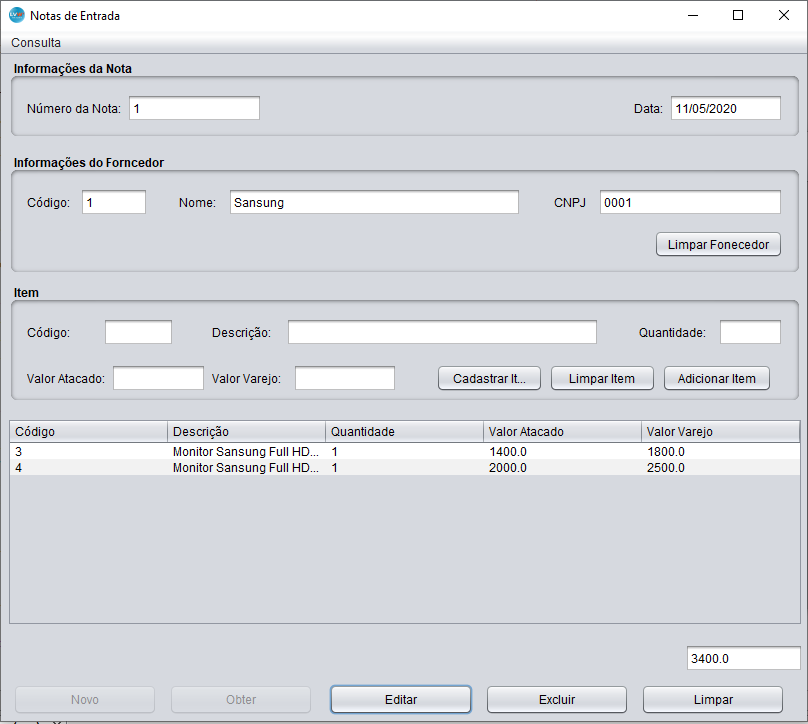


Fig.(18).

Na tela de cadastro de pedidos temos funcionalidades parecidas à tela de cadastro de notas, porém na barra de menus adicionamos as opções de cadastro de clientes juntamente com a de consulta para ajudar o usuário caso precise cadastrar um usuário no momento de fazer uma nota de saída como representado na figura 19.

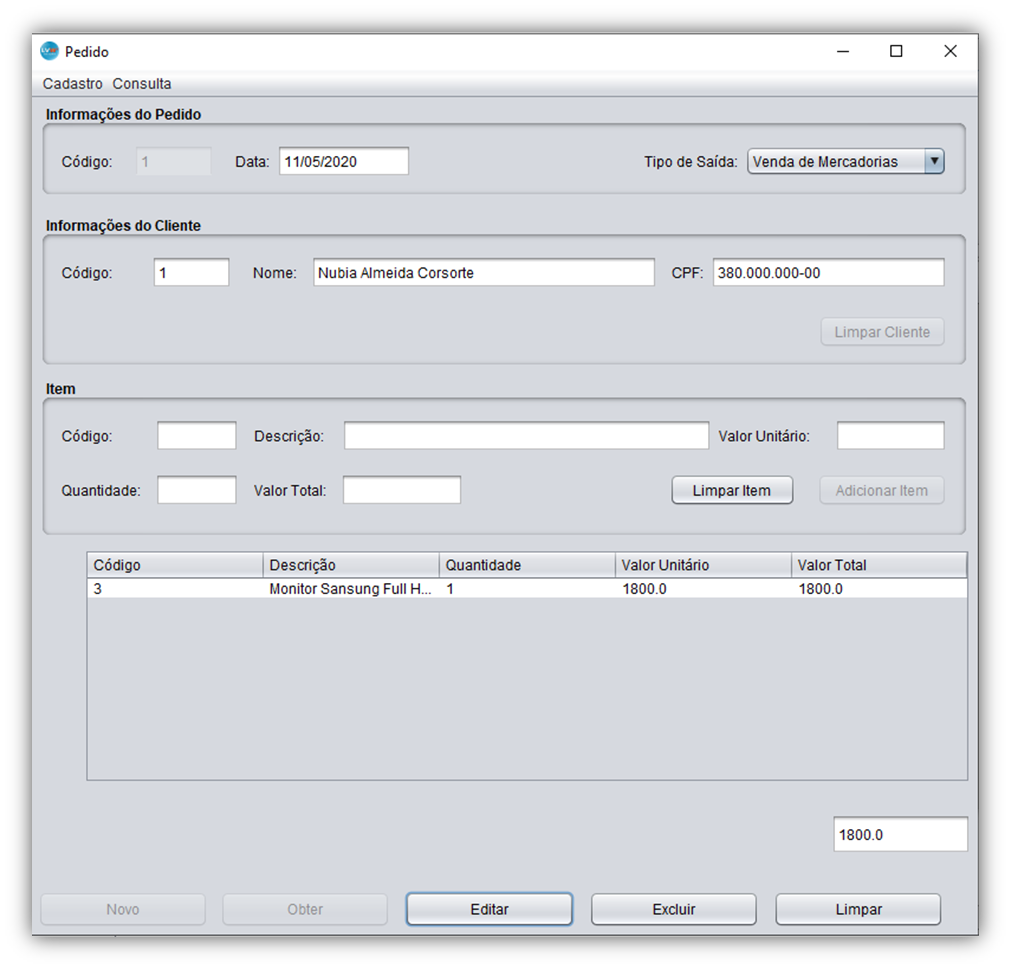


Fig.(19).

Na categoria de controle colocamos funções que auxiliam na administração do estoque, mercadoria e notas para controle interno da empresa.

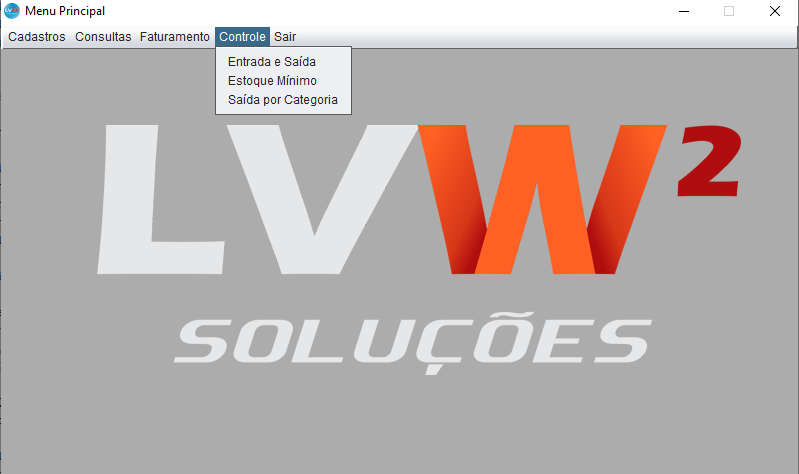


Fig.(20).

Como primeira funcionalidade colocamos consulta de entrada e saída de estoque pelo código do produto utilizando um intervalo de datas como representado na figura 21.

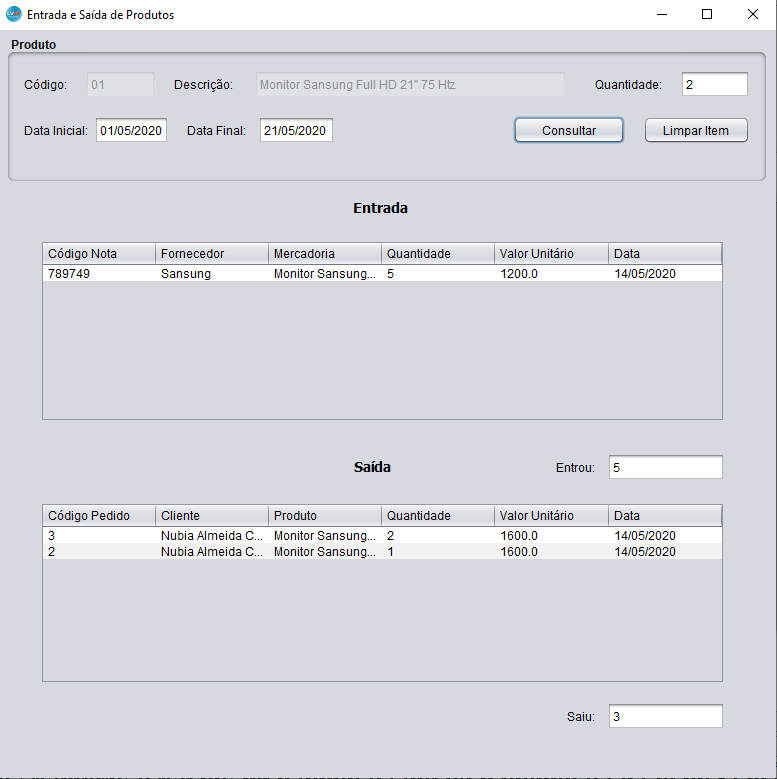


Fig. (21).

Adicionamos a janela de estoque mínimo onde mostramos todas as mercadorias que estão com a quantidade abaixo da quantidade desejada no sistema, nesta janela podemos exportar um documento no formato CSV da data atual dos itens da lista como mostrado na figura 22.

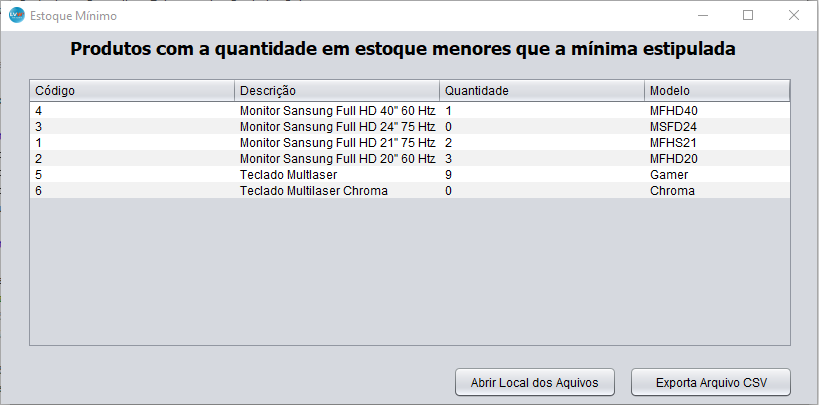


Fig.(22).

Na opção de saída por categoria mostramos os itens de uma determinada categoria por um intervalo de tempo, funcionalidade útil para pesquisar possíveis erros de emissões de notas entre outros problemas de administração do sistema.

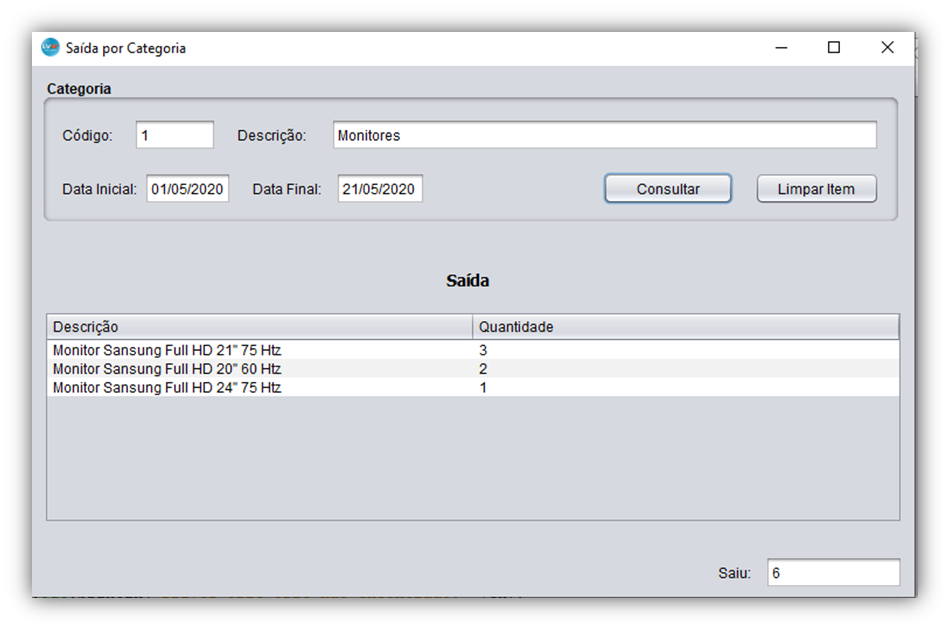


Fig.(23).

Como última função temos a opção de fechar o sistema, aqui previamente haverá uma confirmação como forma evitar do usuário fechar o sistema sem intenção logo depois o sistema executa a ação desejada como mostrado na figura 24.

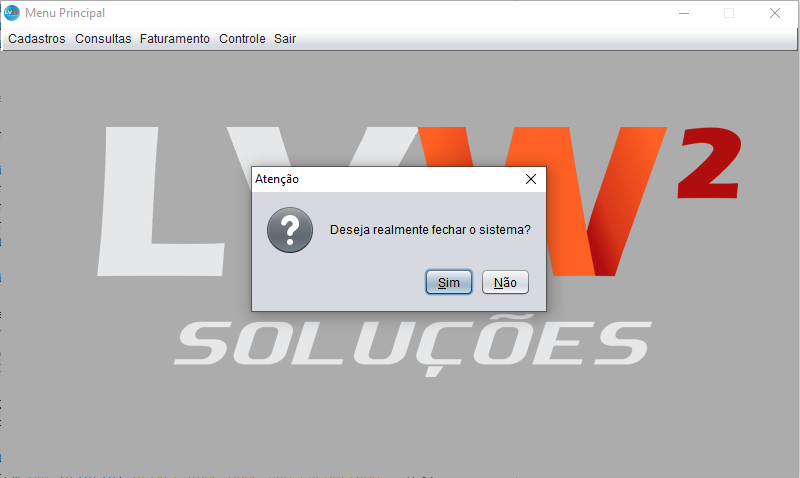
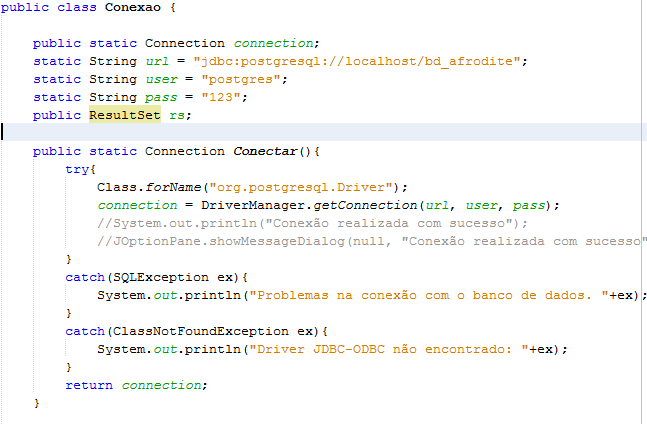
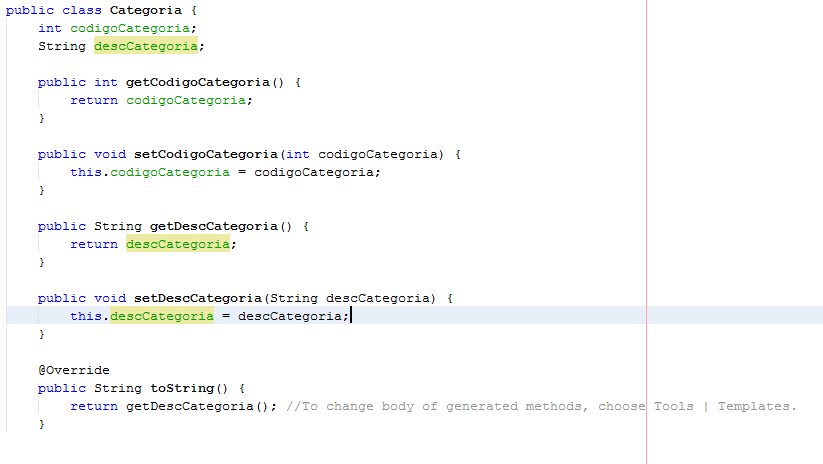
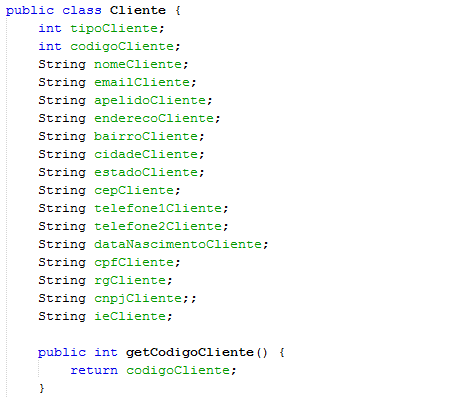


Fig.(24).

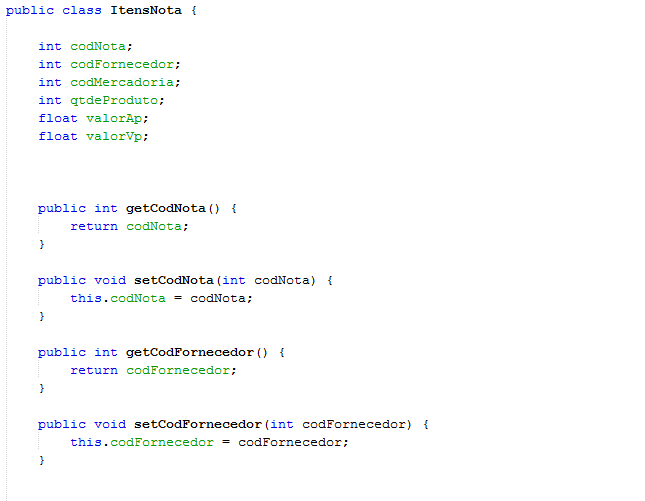
**Relatório com as linhas de código**

****









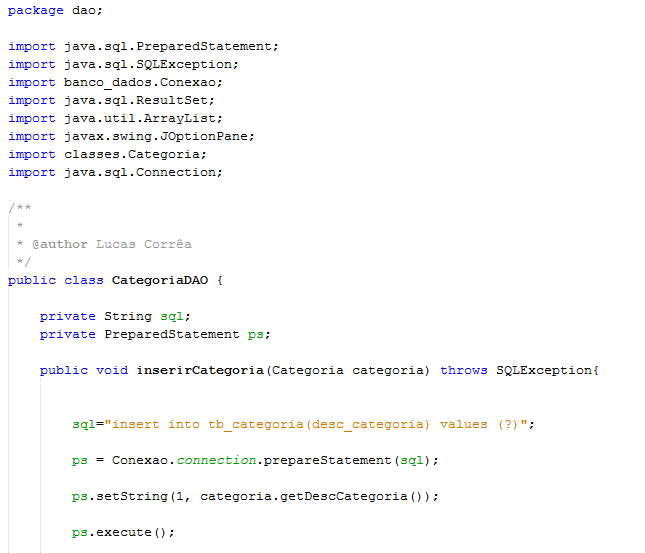


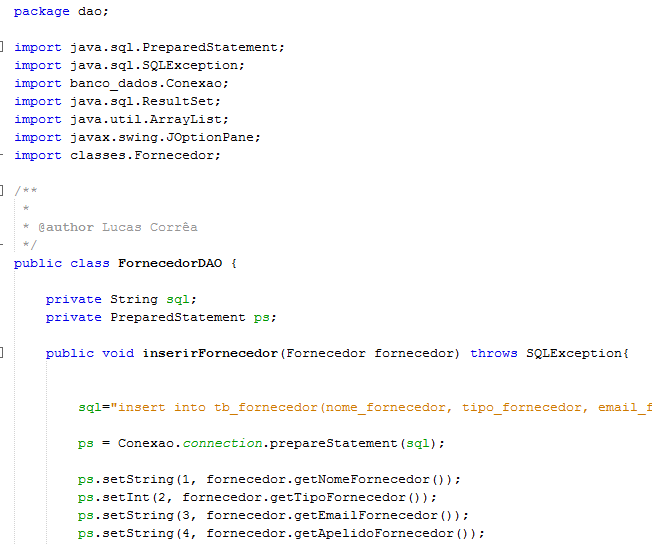


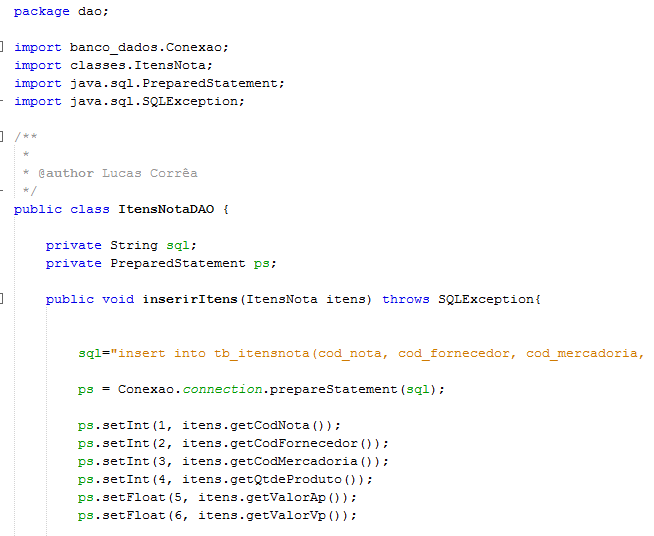




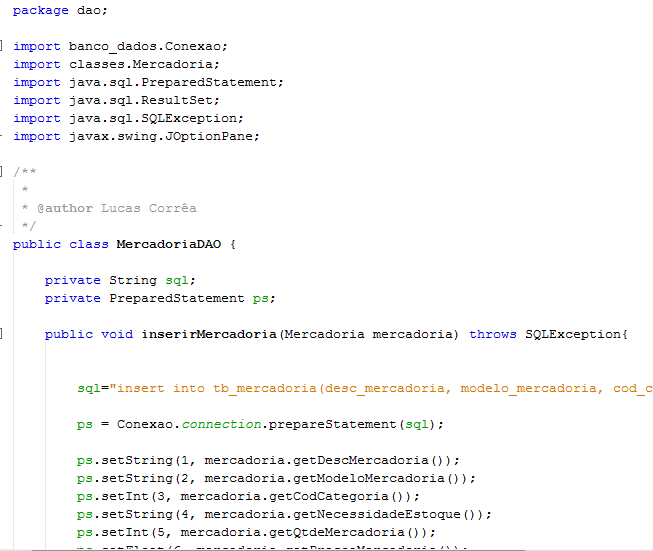


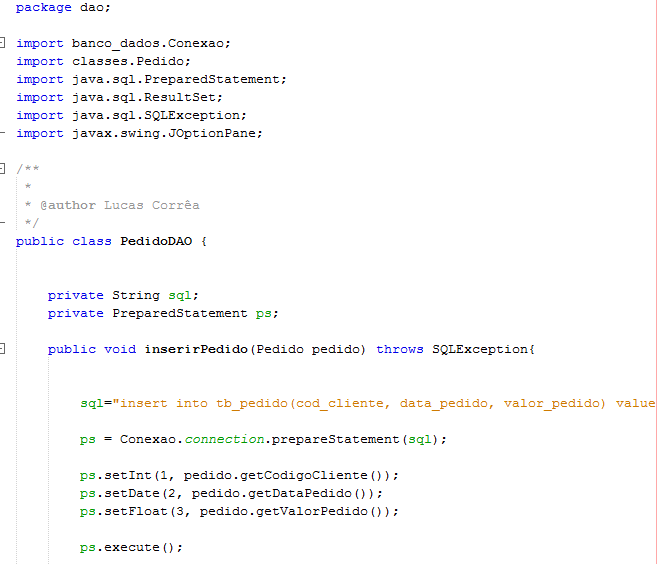




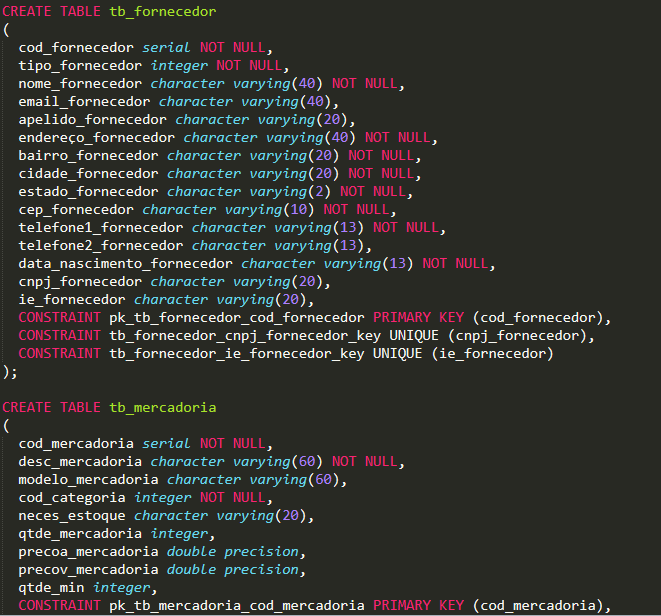


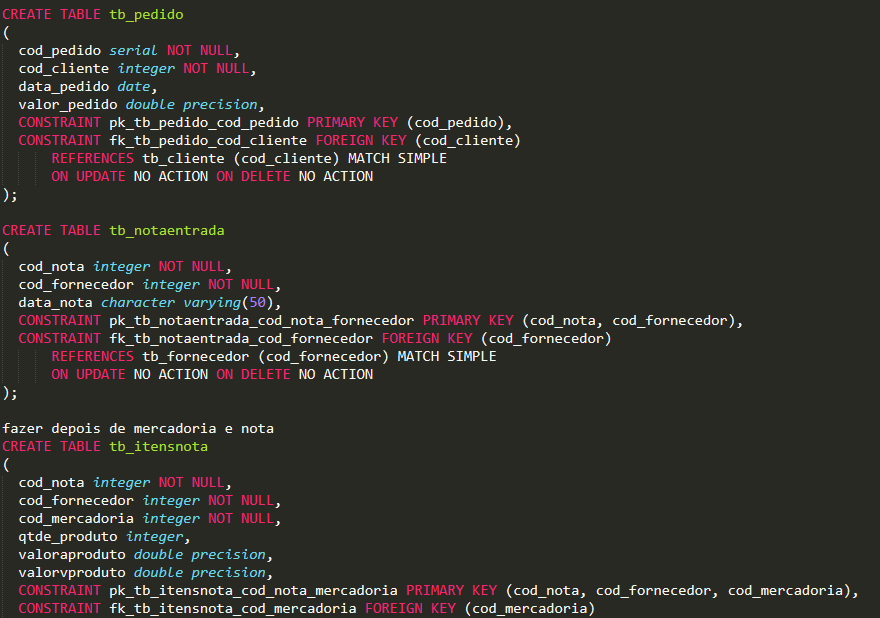


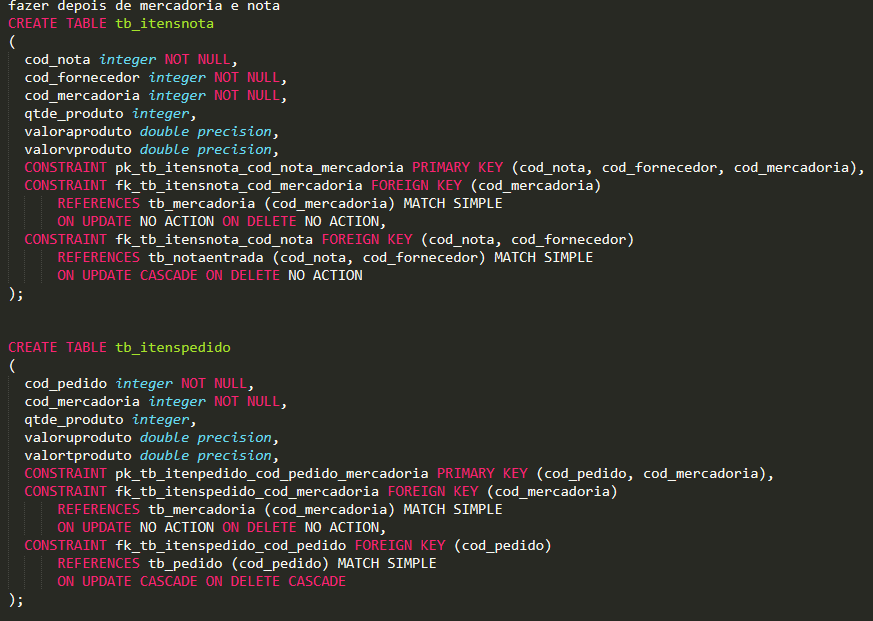










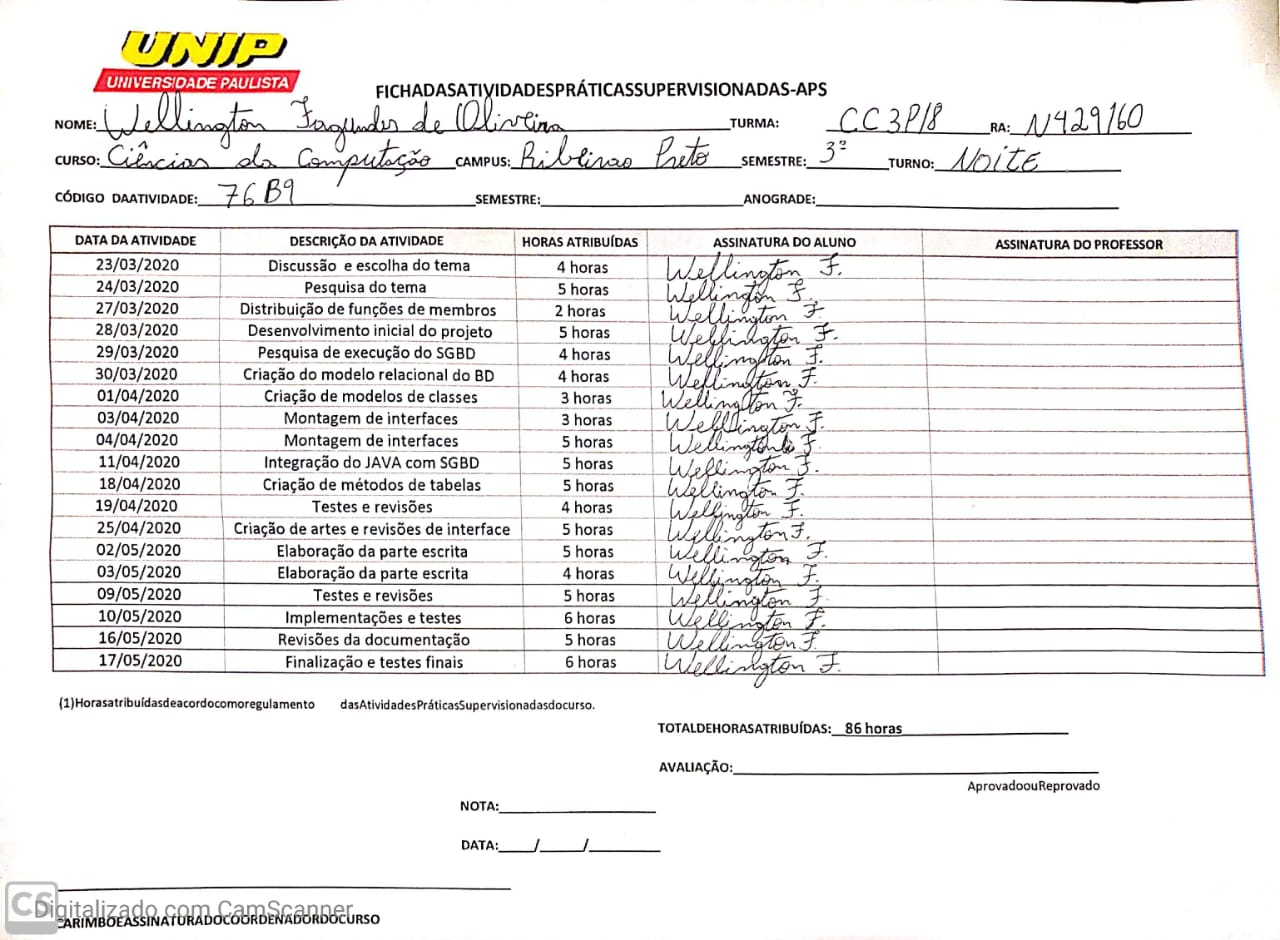


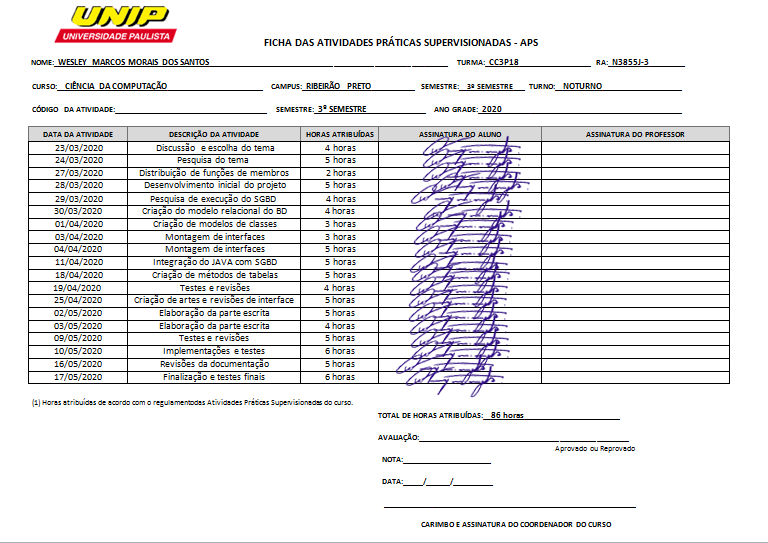
**Bibliografia**

Caelum - Java para desenvolvimento web

Caelum - Java e Orientação a Objetos

Kathy Sierra e Bert Bates - Use a cabeça Java





## C:\Users\Joyc\Downloads\lucas_page-0001.jpgC:\Users\Joyc\Downloads\ficha_aps_vinicius_panobianco_digitalizada_page-0001.jpg