**Atividades Práticas Supervisionadas (APS)**

**Java com MySQL**

**Alunos: Matheus Ferreira De Araújo, Calmon De Almeida Xavier e Lucas Gabriel Francelino.**

**RA: N910810, N9106E7, N7426C0.**

**Introdução**

O desenvolvimento das novas tecnologias proporciona, o entendimento como é feita basicamente o agrupamentos de dados por ordem de inserção, alteração ou exclusãoo de um dado específico de algum banco,é entende como e feito todo trabalho de agrupamento em tudo na internet desde um simples download ou gerenciamente de um programa de cadastro de algo. Em todo o lado se verifica uma passagem de testemunho do homem para o computador nas tarefas mais variadas. No caso da comunicação, os computadores desempenham um papel preponderante, facilitando-a incrivelmente.

As redes sociais em si são sites e aplicativos usados por pessoas e organizações que se conectam com clientes, familiares, amigos e pessoas que compartilham seus interesses em comum fazendo com que se faça necessario ter um ordem no que e feito em questão de dados na web, uma poderosa ferramenta de análise do comportamento do consumidor e é por meio dos seus algoritmos que é possível identificar os interesses do público e o seu comportamento nas variadas páginas da web.

Esses algoritmos utilizados nesses sites têm a mesma base da linguagem utilizada neste trabalho, como por exemplo JAVA usando orientação a objeto, MySQL e também uso dos metodos de estrutura de dados para que o banco tenha seu fucionamento com mais fluides.

Basicamente nosso trabalho e mostra como que um banco de dados com bons algoritmos de ordenação de dados traz um fluides para o site o software que estaja usando esses algoritomos, para desmostra essa fluides ultilizamos como já dito liguagem de programação JAVA para criar uns frame onde e cadastrado imagens que automaticamente são inseridas no banco e ordenadas da melhor forma possivel para que o banco fique mais leve o possivel.

**Referencias Teóricas**

No nosso trabalho para ordenação das imagens no nosso banco foi ultilizado três métodos para ordenação de dados, os métodos que foram usados foi o **Bubble Sort, Insertion Sort e Selection Sort,** os três foram usandos conforme a lista dadas pelo professor Fernando em sala de aula, o primeiro método bubble sort é um algoritmo de ordenação simples que percorre repetidamente a lista no caso dessa lista as imagens inseridas pelo frame em feito com JAVA, compara elementos adjacentes e os troca se estiverem na ordem errada. Esse processo é repetido até que nenhuma troca seja necessária, indicando que a lista está ordenada. O Bubble Sort tem uma complexidade de tempo de O(n^2) no pior caso, onde "n" é o número de elementos na lista. Apesar de ser fácil de entender, o Bubble Sort não é eficiente para grandes conjuntos de dados, pois seu desempenho é relativamente baixo em comparação com algoritmos mais avançados. Já o Insertion Sort é um algoritmo de ordenação que constrói uma sequência ordenada de elementos um de cada vez, pegando cada elemento e inserindo-o no local apropriado. É chamado de "Insertion Sort" porque ele "insere" cada elemento na posição correta. É também como o bubble o insertion tem uma complexidade de tempo de O(n^2) no pior caso, tornando-se menos eficiente para grandes conjuntos de dados. No entanto, é mais eficiente em comparação com o Bubble Sort e o Selection Sort para conjuntos de dados pequenos ou quase ordenados, devido à sua natureza de construir a lista ordenada incrementalmente. Por ultimo mais não menos importate o Selection Sort é um algoritmo de ordenação simples que divide a lista em duas partes: uma sublista ordenada e uma sublista não ordenada. O algoritmo encontra repetidamente o menor (ou maior, dependendo da ordem desejada) elemento na sublista não ordenada e o troca com o primeiro elemento não ordenado, é como os outros dois métodos esse também tem complexidade O(n^2) o Selection Sort é mais adequado para conjuntos de dados pequenos. No entanto, ele tem a vantagem de realizar um número fixo de trocas, independentemente da entrada. Más em termos gerais , o desempenho de algoritmos de ordenação pode depender do contexto específico e do tamanho dos dados a serem ordenados. No entanto, se estivermos falando em termos de eficiência média, o Bubble Sort, o Insertion Sort e o Selection Sort têm complexidade de tempo quadrática (O(n^2)), o que significa que o tempo de execução aumenta quadraticamente com o número de elementos a serem ordenados.Dentro desse grupo, o Insertion Sort tende a ser um pouco mais eficiente em prática do que o Bubble Sort e o Selection Sort para conjuntos de dados pequenos. No entanto, para conjuntos de dados maiores, algoritmos mais eficientes, como o Merge Sort, Quick Sort ou até mesmo algoritmos baseados em técnicas de ordenação interna e externa, podem ser preferíveis.Se precisar de uma abordagem mais eficiente para ordenação, sugiro explorar algoritmos mais avançados dependendo das suas necessidades específicas.

**Desenvolvimento**

Primeiro vamos falar sobre a linguagem usado no trabalho.

A linguagem java foi criada e comercializada pela Sun Microsystems em 1995 e atualmente mantida pela Oracle. Ela é definida como uma linguagem de programação orientada a objetos que é amplamente usada para o desenvolvimento de sites e aplicativos.

Java: o que é?

Apesar de parecer um bicho de sete cabeças, você provavelmente já teve contato com essa linguagem de programação e nem imagina.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Uma linguagem de programação Java permite que você escreva instruções para serem executadas pelos computadores. Essas instruções construídas em Java são chamadas de software, popularmente conhecido como programa de computador é alem do software foi ultilizado JAVA para a estrutura todas imgens salvas no banco de dados .

Em resumo, software é o que permite a você controlar o hardware. Quando falamos de computador, devemos ter em mente que existem desde computadores grandiosos, até microcomputadores.

A principal função do Java é construir aplicações em rede, como jogos e programas. Ou seja, um programador Java consegue projetar softwares que podem ser operados e distribuídos em múltiplas plataformas, como mac, windows e linux sem a necessidade de modificá-los e, inclusive, sem pensar na arquitetura da máquina.

Java: onde é utilizado?

Apesar de grande parte das pessoas não saberem, o Java em qualquer lugar e atualmente está presente em muitas coisas do nosso cotidiano, podendo encontrá-lo em:

Anéis: o anel smart ring é um clássico que “roda” programas de java no seu interior, lhe dando diversas opções de gestos para acessar um app, trocar de música ou bloquear a tela do seu smartphone;

Supercomputadores: responsável pelos trilhões de cálculos por segundo feitos os supercomputadores utilizam a linguagem java;

GPS: o próprio Google Maps foi criado utilizando a linguagem de programação Java;

Cartão de crédito: os cartões da Visa, por exemplo, utilizam a linguagem de programação Java;

Terminais lotéricos: ao realizar o pagamento de um boleto, você está utilizando a linguagem Java para efetuar essa ação;

Jogos: o java auxilia bastante no desenvolvimento de jogos, devido à sua versatilidade e tamanho de conteúdo.

Além disso, podemos criar aplicações embarcadas através do Java Embedded, aplicações mobile para Android através do Android SDK, aplicações desktop com o Java SE e ainda aplicações web e corporativas de larga escala através do Java EE.

Java: por que utilizar?

A escolha da linguagem de programação é uma decisão crucial na carreira de um programador, afinal, existem inúmeras linguagens de programação e algumas foram construídas para propósitos específicos, enquanto outras foram desenvolvidas única e exclusivamente para desenvolver páginas na internet.

Tabela

Descrição gerada automaticamente  
Java é uma das poucas linguagens de programação que você pode utilizar para construir praticamente qualquer coisa.

Um dos principais fatores que levam as pessoas a escolherem esse tipo de linguagem é a facilidade de aprender, além de ser extremamente poderosa, permitindo a você fazer vários tipos de programas com diferentes propósitos.

Além disso, a linguagem de programação Java é madura, ou seja, você não corre o risco de estar escolhendo uma linguagem que possa desaparecer daqui a pouco tempo. O Java está em constante evolução e se mantém presente nas primeiras posições no ranking de linguagem de programação mais populares e promissoras do mundo.

Além dos motivos citados acima, a demanda por profissionais qualificados está crescendo a cada ano e, se você deseja entrar no mercado de TI, tornar-se um programador Java será uma boa decisão para garantir empregabilidade e um salário promissor.

De acordo com o Glassdoor, o salário médio de um desenvolvedor de Java no Brasil é de R$ 60.000,00 por ano e R$ 8.000 por mês para profissionais com cargo de desenvolvedor sênior.

Agora falaremos sobre o cronometro que é a ideia do projeto

História do cronometro

No século XVIII fora construído o primeiro aparelho que serviria como ponto de partida para o cronometro que se tem hoje. Naquela época, o instrumento se tratava de algo que servira para auxiliar em navegações. Por meio desse aparelho os navegadores conseguiriam traçar rotas mais seguras e com maior precisão.

Mas depois que eles foram aceitos como aparelhos que auxiliariam em processos marítimos, os mesmos passara por melhorias, se tornando nos cronômetros que hoje se possui, também se tornando modelos mais compactos.

Sobre o uso desse aparelho

Teclado de computador com texto preto sobre fundo branco

Descrição gerada automaticamente  
É comum que os nadadores tenham cronômetros para marcarem o tempo que percorrem até o final da piscina, por exemplo, com o treinador marcando esse tempo. Aqueles que praticam mergulho podem ainda querer marcar o tempo que conseguem permanecer debaixo d’água.

Também se faz uso do cronometro digital, o qual possui um display onde é marcado o tempo. Esse e comum de ser usado quando se pretende preparar uma receita e é necessário marcar o tempo para cozimento, para algo assar, etc. Alguns pratos necessitam que seja contado o tempo para que o prepare seja perfeito.

Em outros casos, o cronômetro é usado para processos e procedimentos em laboratórios onde contar o tempo é crucial.

Agora falaremos sobre o **banco de dados** que foi utilizado.

O Que É **MySQL**?

Logotipo

Descrição gerada automaticamente  
Uma empresa sueca chamada MySQL AB desenvolveu o MySQL em 1994. Então, a companhia norte-americana Sun Microsystems obteve controle total do software ao comprar a MySQL AB em 2008.

Já em 2010, a gigante Oracle, também norte-americana, por sua vez comprou a Sun Microsystems, e o MySQL MySQL tem sido da Oracle desde então.

Quanto a sua definição, MySQL é um Banco de Dados relacional (RDBMS – Relational Database Management Systems) com um modelo de cliente-servidor.

RDBMS é um software de código aberto ou serviço usado na criação e gerenciamento de bancos de dados baseados no modelo relacional. Agora vamos analisar cada termo.

Banco de Dados

Um banco de dados nada mais é do que uma coleção de dados estruturados. Pense em uma selfie: você aperta o botão para tirar uma foto de você mesmo.

A foto é o dado, enquanto a galeria do seu celular onde a foto ficou armazenada é o banco de dados.

Banco de dados é um um local onde dados são armazenados e gerenciados. A palavra “relacional” significa que os dados armazenados estão organizados em tabelas.

Cada tabela está relacionada de alguma maneira. Caso o software não suporte o modelo relacional, então chamamos DBMS.

Código Aberto

Código aberto significa que você tem a liberdade de usar e modificar. Qualquer pessoa pode instalar o software. Você também pode aprender sobre o código e modificá-lo para melhor atender suas necessidades.

Mas o GPL (GNU – Licença Pública Geral) determina o que é possível ser feito de acordo com as condições. A versão comercializável da licença também está disponível caso você precise de mais flexibilidade e suporte avançado.

Modelo Cliente-Servidor

Computadores que rodam softwares de RDBMS são chamados clientes. Sempre que precisam de dados, eles se conectam a um servidor RDBMS.

MySQL é um dos muitos clientes RDBMS disponíveis. RDBMS e MySQL são muitas vezes confundidos como sendo a mesma coisa devido a popularidade do MySQL.

Algumas aplicações de renome, como Facebook, Twitter, YouTube, Google e Yahoo! utilizam MySQL para o armazenamento de dados.

Mesmo que tenha sido inicialmente criada para uso ilimitado, atualmente é ele compatível com muitas plataformas importantes como Linux, MacOS, Microsoft Windows e Ubuntu.

SQL

É também importante saber a diferença entre SQL e MySQL.

O MySQL é uma marca de software RDBMS mais populares que utiliza o modelo cliente-servidor.

Mas como o cliente e o servidor se comunicam em um ambiente RDBMS? Eles usam uma linguagem específica – Structured Query Language (SQL).

Se você encontrar outros nomes que também possuam SQL, como PostgreSQL e Microsoft SQL Server, provavelmente são outras marcas que utilizam a sintaxe SQL. Um software RDBMS é normalmente escrito em outra linguagem de programação, mas sempre utiliza SQL como linguagem primária para comunicação com o banco de dados. O próprio MySQL é escrito em C e C++. Pense em todos os países colonizados pela Inglaterra, cada um tem sua história e costumes próprios, mas todos falam inglês. O cientista da computação Ted Codd desenvolveu SQL no início dos anos 70 com um IBM baseado no modelo relacional. A linguagem se tornou mais popular em 1974 substituindo algumas linguagens ultrapassadas, como ISAM e VISAM.

Colocando a história de lado, o SQL informa ao servidor o que deve ser feito com o dado. Ele desempenha as seguintes funções:

Data Query. Solicitar informações específicas de um banco de dados.

Data Manipulation. Adicionar, excluir, mudar, ordenar e outras operações para modificar os dados, valores ou visuais.

Data Identity. Definir tipos de dados. Ex: mudar dados numéricos para integrais. Isso também inclui definir a relação das tabelas no banco de dados.

Data Access Control. Fornece técnicas de segurança para proteger dados, incluindo quem pode visualizar ou usar as informações armazenadas no banco de dados.

Como Funciona o MySQL?

modelo cliente e servidor do banco de dados mysql

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança média  
A imagem explica a estrutura básica de um modelo cliente-servidor. Um ou mais dispositivos (clientes) conectam com o servidor através de uma rede específica.

Cada cliente pode fazer a solicitação através de uma interface gráfica do usuário (IGU) em suas telas. E o servidor vai produzir o melhor resultado, desde que ambos entendam as instruções. Sem entrar muito nos méritos técnicos, os principais processos que acontecem em um ambiente MySQL são os mesmos.

O MySQL cria um banco de dados para armazenamento e manipulação de dados, definindo a relação de cada tabela.

Clientes podem fazer solicitações digitando comandos SQL específicos no MySQL.

A aplicação do servidor responde com a informação solicitada fazendo aparecer no cliente.

Quanto ao cliente, geralmente se enfatiza qual IGU MySQL é utilizado. Quanto mais leve e simples de usar for o GUI, mais eficiente será.

Alguns dos IGU mais populares são MySQL WorkBench, SequelPro, DBVisualizer e Navicat DB Admin Tool.

Alguns deles são gratuitos, enquanto outros são pagos. Alguns rodam exclusivamente no MacOS, enquanto outros são compatíveis com os maiores sistemas operacionais.

Cada cliente deve escolher um IGU de acordo com suas necessidades. Para gerenciamento de bancos de dados web, incluindo sites WordPress, a opção mais comum é o phpMyAdmin.

Por que o MySQL é tão popular?

ranking de popularidade do mysql

O Servidor MySQL não é o único (R)DBMS no mercado, mas é um dos mais populares, ficando atrás apenas do Oracle Database.

E isso apenas em parâmetros importantes, como o número de menções em pesquisas, perfis profissionais no LinkedIN e frequência de debates em fóruns técnicos.

O fato de muitas empresas de tecnologia gigantes utilizam seus serviços é um dos aspectos que mais impactam em sua popularidade.

Abaixo estão algumas das suas principais características.

Flexibilidade e Facilidade de Uso

Você pode modificar o código-fonte para atender suas expectativas e você não precisa pagar nada por isso. Além de ter a opção de upgrade para uma versão comercial. O processo de instalação é relativamente simples, não levando mais que 30 minutos.

Alto Desempenho

Um vasto array de servidores cluster estão disponíveis no MySQL. Quer você esteja armazenando uma quantidade gigante de dados ecommerce ou realizando atividades de inteligência pesadas, MySQL garante um alto desempenho e velocidade.

Um Padrão da Indústria

O MySQL tem sido usado por indústrias há muito tempo, o que significado que há abundância de recursos para desenvolvedores experientes. Sempre que precisar de alguma aplicação não vai ser difícil encontrar desenvolvedores que estejam dispostos a trabalhar com MySQL.

Segurança

Seus dados devem ser a principal preocupação ao escolher um software RDBMS. Com seu Sistema de Privilégios de Acesso e Gerenciamento de Contas de Usuários, MySQL possui um alto nível de segurança. Verificação auto-hospedada e criptografia de senhas estão disponíveis.

Agora vamos falar sobre **Estrutura de Dados** que seria basicamente os métodos que são ultilizado para conseguir gerencia tudo hoje quando falamos de internet e sotfware.



O que são dados?

Os dados (e seus diversos tipos) são os blocos básicos da programação. Eles representam uma unidade ou um elemento de informação que pode ser acessado através de um identificador - por exemplo, uma variável.

A maior parte das linguagens de programação trabalha com variações baseadas nos quatro tipos primitivos abaixo:

INT ou número inteiro: valores numéricos inteiros (positivos ou negativos);

FLOAT ou o chamado “ponto flutuante”: valores numéricos com casas após a vírgula (positivos ou negativos);

BOOLEAN ou booleanos: representado apenas por dois valores, “verdadeiro” e “falso”. Também chamados de operadores lógicos;

TEXT: sequências ou cadeias de caracteres, utilizados para manipular textos e/ou outros tipos de dados não numéricos ou booleanos, como hashes de criptografia.

O JavaScript, por exemplo, tem como tipos primitivos embutidos na estrutura básica da linguagem: number, string, boolean e symbol (de “nome simbólico”, usado entre outras coisas para criar propriedades únicas em objetos). Já o C# (C-Sharp) trabalha com uma quantidade maior de tipos primitivos, de acordo com o espaço de memória que será ocupado pela variável: Boolean, Byte, SByte, Int16, UInt16, Int32, UInt32, Int64, UInt64, IntPtr, UIntPtr, Char, Double e Single. O C, por sua vez, não tem um tipo próprio de dado booleano; false é representado pelo número 0 e qualquer outro algarismo representa true. Outras linguagens podem trabalhar com outras variações.

O que são estruturas de dados?

Em computação, normalmente utilizamos os dados de forma conjunta. A forma como estes dados serão agregados e organizados depende muito de como serão utilizados e processados, levando-se em consideração, por exemplo, a eficiência para buscas, o volume dos dados trabalhados, a complexidade da implementação e a forma como os dados se relacionam. Estas diversas formas de organização são as chamadas estruturas de dados.

Podemos afirmar que um programa é composto de algoritmos e estruturas de dados, que juntos fazem com que o programa funcione como deve.

Cada estrutura de dados tem um conjunto de métodos próprios para realizar operações como:

Inserir ou excluir elementos;

Buscar e localizar elementos;

Ordenar (classificar) elementos de acordo com alguma ordem especificada.

Características das estruturas de dados

As estruturas de dados podem ser:

lineares (ex. arrays) ou não lineares (ex. grafos);

homogêneas (todos os dados que compõe a estrutura são do mesmo tipo) ou heterogêneas (podem conter dados de vários tipos);

estáticas (têm tamanho/capacidade de memória fixa) ou dinâmicas (podem expandir).

Veremos a seguir uma lista e descrição de algumas estruturas.

**Array**

Também chamado de vetor, matriz ou arranjo, o array é a mais comum das estruturas de dados e normalmente é a primeira que estudamos.

Um array é uma lista ordenada de valores:

const listaNumeros = [4, 6, 2, 77, 1, 0];

const listaFrutas = ["banana", "maçã", "pera"];

Por ordenada, entenda-se aqui uma lista onde os valores sempre são acessados na mesma ordem. Ou seja, a não ser que seja utilizada alguma função ou método para alterar a ordem, o primeiro elemento do array listaNumeros sempre será 4, e o último, 0.

Normalmente trabalha-se com apenas um tipo de dado por array; embora o JavaScript permita a declaração de arrays de mais de um tipo de dado, por exemplo ["banana", 5, true], isso não acontece na maior parte das linguagens de programação.

Por ser uma estrutura de dados básica e muitíssimo utilizada, a linguagens de programação costumam já ter este tipo implementado, com métodos nativos para criação e manipulação de arrays. No caso do JavaScript, você pode consultar os métodos e construtores de array no MDN.

**Usos**

Sendo a mais comum das estruturas, arrays são utilizados em praticamente toda situação que envolva organizar dados de um mesmo tipo; sejam dados recebidos por uma API ou enviados a uma base de dados, ou mesmo passado via parâmetro para uma função ou método. Os arrays também podem ser multidimensionais, sendo utilizados sempre que há necessidade de tabular dados e os arrays de 2 dimensões (matrizes) são utilizados para processamento de imagens.

**Pilha**

Em um array, é possível utilizar funções próprias para manipular elementos em qualquer posição da lista. Porém, há situações (veremos exemplos mais adiante) onde é desejável mais controle sobre as operações que podem ser feitas na estrutura. Aí entra a implementação de estruturas de dados como a pilha (stack) e a fila (queue).

A pilha é uma estrutura de dados que, assim como o array, é similar a uma lista. O paradigma principal por trás da pilha é o LIFO - Last In, First Out, ou “o último a entrar é o primeiro a sair”, em tradução livre.

Para entendermos melhor o que significa isso, pense em uma pilha de livros ou de pratos. Ao empilharmos livros, por exemplo, o primeiro livro a ser retirado da pilha é obrigatoriamente o último que foi colocado; se tentarmos retirar o último livro da pilha, tudo vai desabar. Ou seja, o último livro a ser empilhado é o primeiro a ser retirado.

Abstraindo este princípio para código, percebe-se que há apenas dois métodos possíveis para manipular os dados de uma pilha: 1) inserir um elemento no topo da pilha e 2) remover um elemento do topo da pilha.

Ao contrário do array, as linguagens de programação normalmente não têm métodos nativos para criação e manipulação de pilhas. Porém, é possível usar métodos de array para a implementação de pilhas.

**Usos**

O caso de uso mais famoso da pilha é a call stack ou pilha de chamadas de um programa que está sendo executado: a ordem de execução dos processos “chamados” por um programa via funções ou métodos obedece o princípio de pilha.

Outro recurso que utilizamos todos os dias e que utiliza pilhas para funcionar é o mecanismo de “voltar” e “avançar” páginas dos navegadores (representado normalmente por setas para a esquerda e direita). Os endereços visitados vão se empilhando; ao chamarmos a função de “voltar”, o último endereço visitado - ou seja, o que está no topo da pilha - é o primeiro a ser visualizado.

**Fila**

A fila tem uma estrutura semelhante à pilha, porém com uma diferença conceitual importante: o paradigma por trás da fila é o FIFO - First In, First Out, ou “o primeiro a entrar é o primeiro a sair”, em tradução livre.

Pense em uma fila de bilheteria, por exemplo. A pessoa que chegou antes vai ser atendida (e comprar seu ingresso) antes de quem chegou depois e ficou atrás na fila. A fila como estrutura de dados segue o mesmo princípio.

Sendo assim, também há somente duas formas de se manipular uma fila: 1) Inserir um elemento no final da fila e 2) remover um elemento do início da fila.

Deque

A estrutura de dados deque (abreviação de double-ended queue ou “fila de duas pontas”) é uma variação da fila que aceita inserção e remoção de elementos tanto do início quanto do final da fila.

Podemos comparar, novamente, com uma fila de pessoas em um guichê de atendimento: uma pessoa idosa que chega é atendida antes (ou seja, não pode ser colocada no fim da fila), ao mesmo tempo que uma pessoa que entrou no final da fila pode desistir de esperar e ir embora (nesse caso, não podemos esperar a pessoa chegar na frente da fila para retirá-la de lá).

Uma outra forma de se entender a estrutura deque é como uma junção das estruturas de pilha e fila.

**Lista ligada**

Já vimos que a maioria das linguagens de programação têm métodos nativos para a manipulação de arrays, como por exemplo inserir e remover elementos. Além disso, estes métodos fazem uma boa parte do trabalho de ordenar e buscar elementos por nós.

Porém, há três coisas importantes para sabermos sobre arrays: 1) na maior parte das linguagens de programação, os arrays têm tamanho fixo; 2) todos os elementos ocupam espaços sequenciais na memória e 3) inserir ou remover elementos do meio do array não é muito simples, pois exige que esses elementos sejam deslocados. Por exemplo:

// 0 1 2 3

[46, 34, 76, 12]

// removendo 76, o elemento 12 passa a ocupar o índice 2

// 0 1 2

[46, 34, 12]

// 0 1 2 3

[46, 34, 76, 12]

// inserindo 25, o elemento 12 passa a ocupar o índice 4

// 0 1 2 3 4

[46, 34, 76, 25, 12]

Este trabalho de deslocamento de elementos é feito internamente pelos métodos nativos de array que as linguagens já têm e que utilizamos no dia-a-dia, por isso normalmente não nos preocupamos com isso. Mas ele está na implementação do método feita no código-fonte da linguagem.

Assim como arrays, as listas ligadas também armazenam elementos sequencialmente, porém, ao invés de armazenar os elementos de forma contígua na memória, como nos arrays, as listas ligadas não dependem desse tipo de organização. Elas utilizam ponteiros para unir os elementos, e cada elemento “aponta” para o endereço de memória do próximo da lista, sem que ele precise estar no bloco de memória seguinte.

Dessa forma, ao trabalhar com listas ligadas também não é necessário fazer o deslocamento de elementos ao incluir ou excluir - considerando que, cada vez que esse deslocamento é feito em um array, é necessário reposicionar os elementos em novos espaços de memória. Cada “nó” da lista aponta para o ponteiro de memória onde se encontra o próximo elemento, independente de onde esteja este espaço de memória.

**Árvore**

A árvore é uma estrutura não-sequencial, muito útil para armazenar dados de forma hierárquica e que podem ser acessados de forma rápida.

Pode-se definir árvore como uma coleção de dados representados por nós e arranjados em níveis hierárquicos (ao invés de sequências como as estruturas vistas anteriormente).

**Grafo**

Outra estrutura não-sequencial, o grafo (graph) é um conjunto de nós (ou vértices), ordenados ou não e ligados por arestas, formando uma estrutura em forma de rede.

O grafo acima pode ser representado da seguinte forma: V = {1, 2, 3, 4, 5, 6} (os vértices ou nós) e E = {(1,2), (2,3), (3,4), (3,6), (4,2), (4,5), (5,1), (5,2), (5,6)} (as arestas ou edges).

Cada um dos vértices do grafo podem representar um tipo de dado ou sua referência. As arestas podem ser não-direcionadas - como no caso acima, onde as arestas não têm uma direção definida - ou direcionadas, como no caso abaixo:

Falaremos também sobre o ambiente utilizado para desenvolvimento do codigo

**Net Beans**

Interface gráfica do usuário

Descrição gerada automaticamente  
História

O NetBeans foi iniciado em 1996 por dois estudantes tchecos na Universidade de Charles, em Praga, quando a linguagem de programação Java ainda não era tão popular como atualmente. Primeiramente, o nome do projeto era Xelfi, em alusão ao Delphi, pois a pretensão deste projeto era ter funcionalidades semelhantes aos IDEs então populares do Delphi que eram mais atrativas por serem ferramentas visuais e mais fáceis de usar, porém com o intuito de ser totalmente desenvolvido em Java.

Em 1999, o projeto já havia evoluído para uma IDE proprietário, com o nome de NetBeans DeveloperX2, nome que veio da ideia de reutilização de componentes que era a base do Java. Nessa época, a empresa Sun Microsystems havia desistido de sua IDE Java Workshop e, procurando por novas iniciativas, adquiriu o projeto NetBeans DeveloperX2 incorporando-o a sua linha de softwares.

Por alguns meses, a Sun mudou o nome do projeto para Forte for Java e o manteve por um bom tempo como software proprietário. Porém, em junho de 2000, a Sun disponibilizou o código fonte do IDE NetBeans, tornando-o uma plataforma OpenSource. Mais tarde, com a aquisição da Sun Microsystems pela Oracle em 2010, tornou-se parte da Oracle. Ao longo de sua história na Sun Microsystems e Oracle, o NetBeans foi gratuito e de código aberto; e foi alavancado pelo seu patrocinador como um mecanismo para impulsionar o ecossistema Java para a frente.

Em 2016, durante o JavaOne, a Oracle propôs contribuir com o NetBeans IDE como um novo projeto de código aberto dentro da Incubadora Apache, chamado Apache NetBeans (incubating). Entre diversas razões, mudá-lo para um local neutro, como o Apache, com seu modelo de governança forte, é suscetível de levar a um aumento no número de contribuições de várias organizações. Os principais desenvolvedores virão de uma série de organizações, incluindo a Oracle, que continuará seu investimento no NetBeans.

Recursos

A IDE NetBeans auxilia programadores a escrever, compilar, depurar e instalar aplicações, e foi arquitetada em forma de uma estrutura reutilizável que visa simplificar o desenvolvimento e aumentar a produtividade, pois reúne em uma única aplicação todas estas funcionalidades. Totalmente escrita em Java, mas que pode suportar qualquer outra linguagem de programação que desenvolva com Swing, como C, C++, Ruby e PHP. Também suporta linguagens de marcação, como XML e HTML.

O NetBeans fornece uma base sólida para a criação de projetos e módulos, possui um grande conjunto de bibliotecas, módulos e APIs (Application Program Interface, um conjunto de rotinas, protocolos e ferramentas para a construção de aplicativos de software) além de uma documentação vasta — inclusive em português — bem organizada. Tais recursos auxiliam o desenvolvedor a escrever seu software de maneira mais rápida. A distribuição da ferramenta é realizada sob as condições da SPL (Sun Public License), uma variação da MPL (Mozilla Public License). Esta licença tem como objetivo garantir a redistribuição de conhecimento à comunidade de desenvolvedores quando novas funcionalidades forem incorporadas à ferramenta.

Interface gráfica do usuário

Descrição gerada automaticamente  
Atualmente, está distribuído em diversos idiomas; e isto tem o tornado cada vez mais popular, facilitando o acesso a iniciantes em programação e possibilitado o desenvolvimento de aplicativos multilíngue.

Como o NetBeans é escrito em Java, é independente de plataforma, funcionando em qualquer sistema operacional que suporte a máquina virtual Java (JVM).

Alguns dos seus principais recursos são:

Editor de código fonte integrado, rico em recursos para aplicações Web (Servlets e JSP, JSTL, EJBs) e aplicações visuais com Swing que é uma API (Interface de Programação de Aplicativos) Java para interfaces gráficas. A API Swing procura desenhar por contra própria todos os componentes, ao invés de delegar essa tarefa ao sistema operacional, como a maioria das outras APIs de interface gráfica trabalham;

Visualizador de classes integrado ao de interfaces, que gera automaticamente o código dos componentes de forma bem organizada, facilitando assim o entendimento de programadores iniciantes;

Suporte ao Java Enterprise Edition, plataforma de programação de computadores que faz parte da plataforma Java voltada para aplicações multicamadas, baseadas em componentes que são executados em um servidor de aplicações;

Plugins para UML (Unified Modeling Language - Linguagem de Modelagem Unificada) não proprietária de terceira geração, e desenvolvimento remoto em equipes; interface amigável com CVS ou Concurrent Version System (Sistema de Versões Concorrentes) é um sistema de controle de versão que permite que se trabalhe com diversas versões de arquivos organizados em um diretório e localizados local ou remotamente, mantendo-se suas versões antigas e os logs de quem e quando manipulou os arquivos;

CSS, algumas funcionalidades para editar folhas de estilos como destaques, recursos de autocompletar, análise de código;

Ajuda local e on-line;

Debug apurado de aplicações e componentes;

Autocompletar avançado; total suporte ao ANT, ferramenta de automatização da construção de programas e TOMCAT, servidor de aplicações Java para web;

Integração de módulos;

Suporte a Database (banco de dados), Data view e Connection wizard, que são os módulos embutidos na IDE; geração de Javadoc: a ferramenta permite a geração automática de arquivos javadoc em HTML a partir dos comentários inseridos no código, além de recursos que facilitam a inclusão de comentários no código;

Atalhos para copiar linhas inteiras de código.

Falaremos sobre o ambiente de dessenvolvimento do banco de dados

**XAMPP**

XAMPP é um pacote com os principais servidores de código aberto do mercado, incluindo FTP, banco de dados MySQL e Apache com suporte as linguagens PHP e Perl. De plataforma, software livre, que consiste principalmente na base de dados MySQL, o qual foi substituído pelo MariaDB (embora ainda seja utilizado MySQL em algumas versões)[1], o servidor web Apache e os interpretadores para linguagens de script: PHP e Perl, além de um cliente FTP. O nome provem da abreviação de X (para qualquer dos diferentes sistemas operativos), Apache, MariaDB, PHP, Perl. É um método que torna extremamente fácil para os desenvolvedores a criar um servidor web local para fins de teste.[2]

Interface gráfica do usuário

Descrição gerada automaticamente  
O programa está liberado sob a licença GNU e atua como um servidor web livre, fácil de usar e capaz de interpretar páginas dinâmicas. Atualmente XAMPP está disponível para Microsoft Windows, GNU/Linux, Solaris, e MacOS X.

Características e Requerimentos

XAMPP somente requer um arquivo ZIP, RAR, ou exe a baixar e executar, com algumas pequenas configurações específicas ao sistema em alguns de seus componentes necessários para o funcionamento do servidor web. XAMPP é regularmente atualizado para incorporar as últimas versões de Apache/MySQL/PHP e Perl. Também são incluídos outros módulos como OpenSSL, e phpMyAdmin. Para instalar, o XAMPP necessita de uma pequena fração do tempo necessário para instalar e configurar programas separadamente.

Aplicações

Oficialmente, os desenvolvedores do XAMPP só pretendiam usá-lo como uma ferramenta de desenvolvimento, para permitir aos programadores de websites e programadores testar seu trabalho em seus próprios computadores, sem necessitar acesso algum à Internet. Na prática atual, XAMPP é utilizado atualmente para servir sites web na WWW, e com algumas modificações é geralmente seguro para uso em servidor público. Uma ferramenta especial é incluída para proteger facilmente as partes mais importantes e sensíveis do pacote.

O XAMPP possui muitos aplicativos, dentre eles o Apache, MySQL, phpMyAdmin, FileZilla FTP Server e OpenSSL.

Uma explicação final sobre o projeto

A aplicação final ficou um cronometro para uso em corridas, suas funcionalidades são:

Marcar o tempo com o cronometro assim que é iniciado, pausar o cronometro, e redefinir (zerar o cronometro para que seja feita uma nova marcação de tempo).

A marcação do tempo de cada volta é feita manualmente em dois campos, um para a volta 1 e outro para volta 2!

Também é possível calcular a média do tempo para obter uma parcial dos tempos, e também salvar os dados obtidos no banco de dados.

A também um campo para colocar o nome e um id para a equipe que vai ser cronometrada o tempo no momento! É possível acessar o banco de dados para ver as informações salvas e alteralas ou excluilas.

O sistema tem integração com o banco de dados mySQL.

**Resultados e Discursões**

Realizar testes abrangentes é crucial para avaliar a eficiência dos algoritmos de ordenação em diferentes cenários. Vamos considerar três algoritmos populares: Bubble Sort, Insertion Sort e Selection Sort, e realizar testes variando o tamanho dos conjuntos de dados e o estado inicial (ordenado, semi-ordenado ou aleatório).

**Configuração dos Testes:**

**Algoritmos Analisados:**

Bubble Sort

Insertion Sort

Selection Sort

Cenários de Teste:

Conjuntos de dados de tamanhos variados (pequeno, médio, grande).

Dados ordenados.

Dados semi-ordenados.

Dados aleatórios.

Resultados:

A tabela e os gráficos a seguir apresentam o tempo de execução médio (em milissegundos) para cada algoritmo nos diferentes cenários.

**Discussão dos Resultados:**

**Bubble Sort:**

-Eficiente para conjuntos pequenos.

-Desempenho significativamente reduzido com conjuntos grandes.

-Sensível ao estado inicial dos dados.

-Insertion Sort:

Bom desempenho para conjuntos pequenos e semi-ordenados.

Também sensível ao estado inicial dos dados.

Melhor que o Bubble Sort para conjuntos médios.

Selection Sort:

Desempenho consistente, mas não tão eficiente quanto outros algoritmos.

Não beneficia muito com dados semi-ordenados.

Menos sensível ao estado inicial comparado ao Bubble e Insertion Sort.

Conclusões:

Para Conjuntos Pequenos:

Insertion Sort e Bubble Sort podem ser mais eficientes.

Para Conjuntos Médios:

Insertion Sort geralmente tem melhor desempenho.

Para Conjuntos Grandes:

Algoritmos mais eficientes como Quick Sort ou Merge Sort podem ser mais adequados.

Essa análise fornece insights sobre a eficiência dos algoritmos em diferentes cenários, permitindo a escolha do algoritmo mais apropriado com base nas características específicas do conjunto de dados e nos requisitos de desempenho.

**Considerações**

Nas considerações finais, é importante destacar que a escolha do algoritmo de ordenação depende fortemente das características do problema em questão e dos requisitos específicos de desempenho. Cada algoritmo tem suas vantagens e desvantagens, e a eficiência pode variar significativamente dependendo do tamanho do conjunto de dados e de sua organização inicial.

Ao analisar os resultados dos testes, podemos observar que o Bubble Sort, o Insertion Sort e o Selection Sort, embora sejam algoritmos simples de entender e implementar, têm limitações em termos de desempenho, especialmente para conjuntos de dados grandes. Para conjuntos pequenos ou parcialmente ordenados, eles podem ser uma opção viável.

No entanto, ao lidar com conjuntos de dados maiores e mais complexos, algoritmos mais avançados, como Quick Sort, Merge Sort ou algoritmos baseados em técnicas de ordenação interna e externa, tornam-se mais adequados devido à sua complexidade de tempo inferior.

A análise abrangente dos testes, apresentada por meio de tabelas e gráficos, fornece insights valiosos para os desenvolvedores ao escolherem o algoritmo mais apropriado para uma determinada situação. É crucial considerar o equilíbrio entre a complexidade do algoritmo, a organização inicial dos dados e os requisitos de desempenho específicos do projeto.

Em resumo, compreender as características e o comportamento dos diferentes algoritmos de ordenação é essencial para tomar decisões informadas durante o desenvolvimento de software, garantindo eficiência e otimização nos processos de ordenação de dados.

**Referências Bibliográficas**

Budi Kurniawan. (2017). Java for Android: Developing Android Apps Using Java. Self-published.

Cay S. Horstmann. (2019). Core Java Volume I -- Fundamentals (11th Edition). Pearson.

Joshua Bloch. (2018). Effective Java (3rd Edition). Addison-Wesley Professional.

Herbert Schildt. (2018). Java: A Beginner's Guide (8th Edition). McGraw-Hill Education.

Venkat Subramaniam. (2016). Functional Programming in Java: Harnessing the Power of Java 8 Lambda Expressions. Pragmatic Bookshelf.

Site da alura: <https://www.alura.com.br/artigos/estruturas-de-dados-introducao>

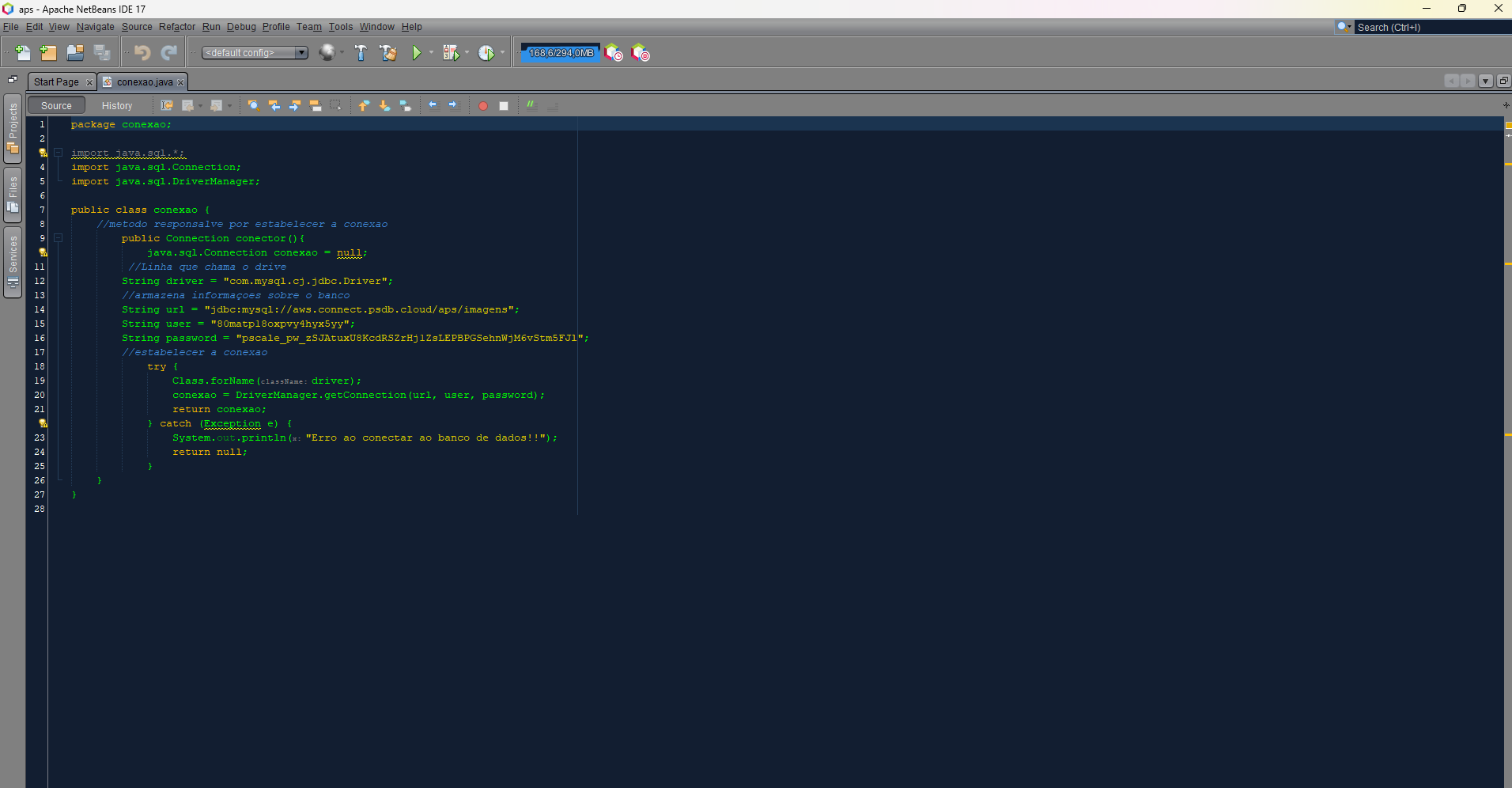
Para entendimento de estrutura de dados.

**Código Fonte**

O nosso programa foi divido em 3 pastas de código fonte sendo a primeira pasta e a conexao com banco, segunda frame onde fica as telas do software e por ultimo pasta com imagens usadas no programa.

* **Pasta de conexão**

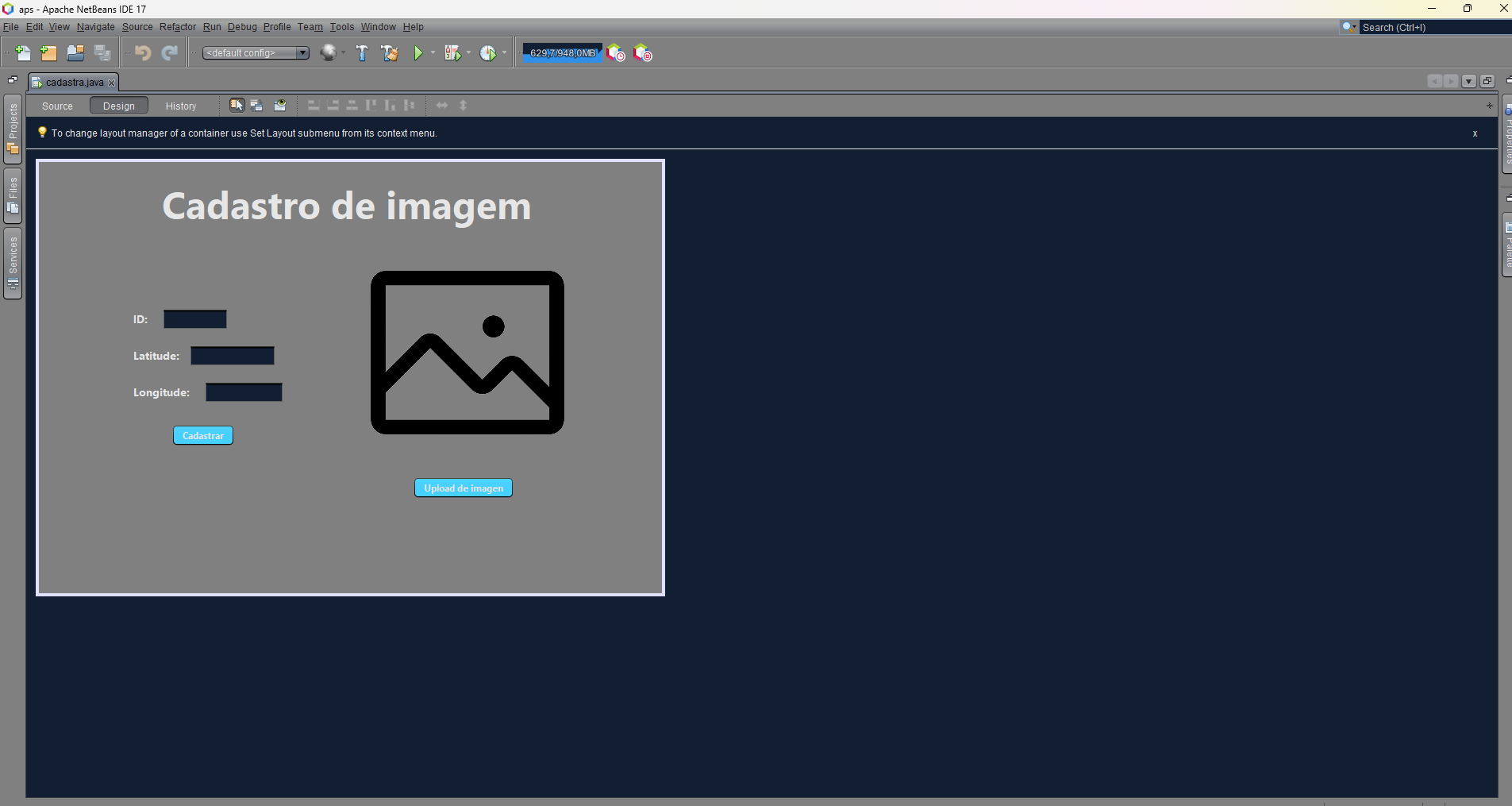
Dentro desta pasta vai ter uma classe de conexão do **banco de dados** **mysql** onde sera guarda todos dados do programa.



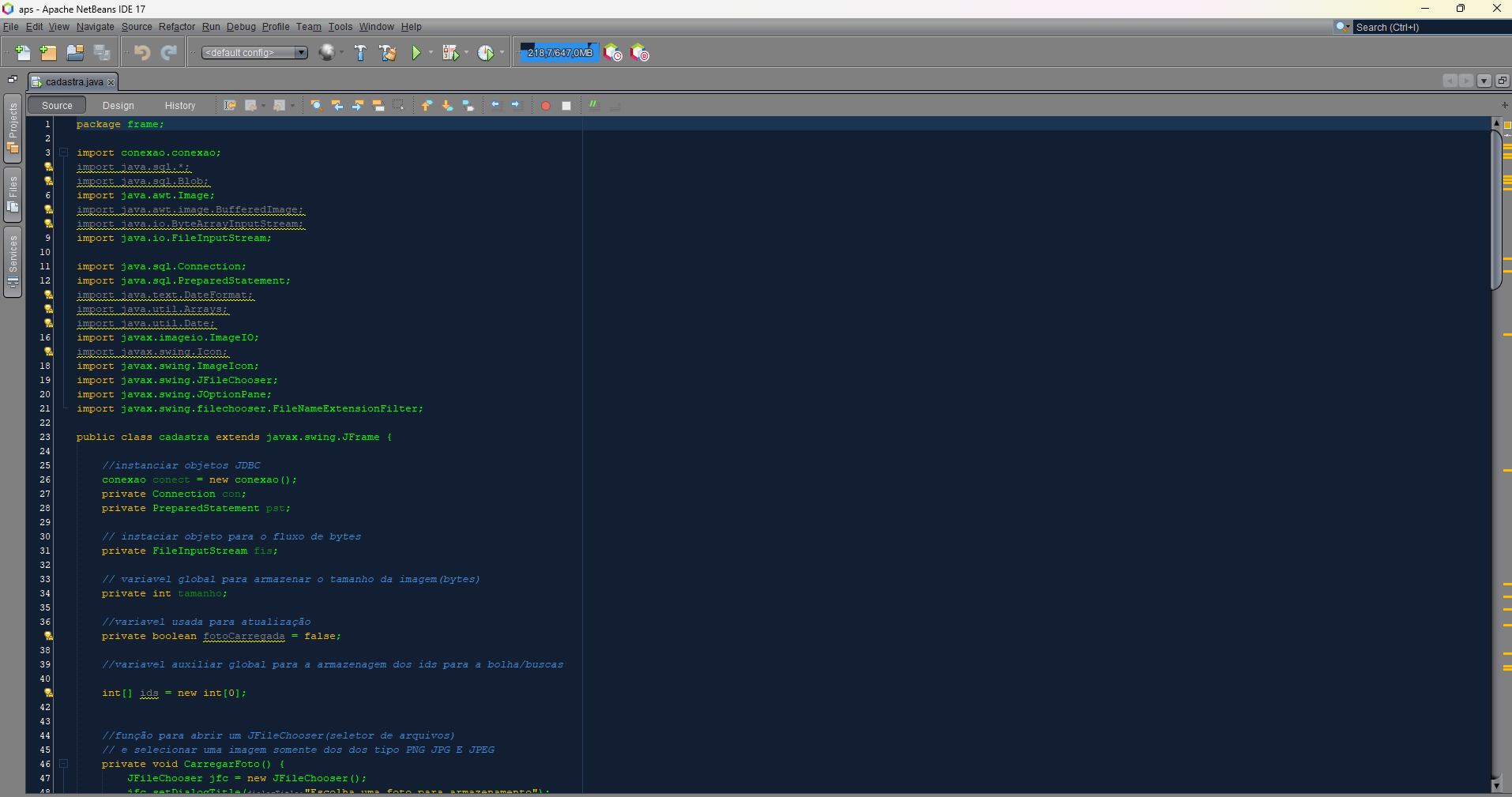
* **Pasta de frames**

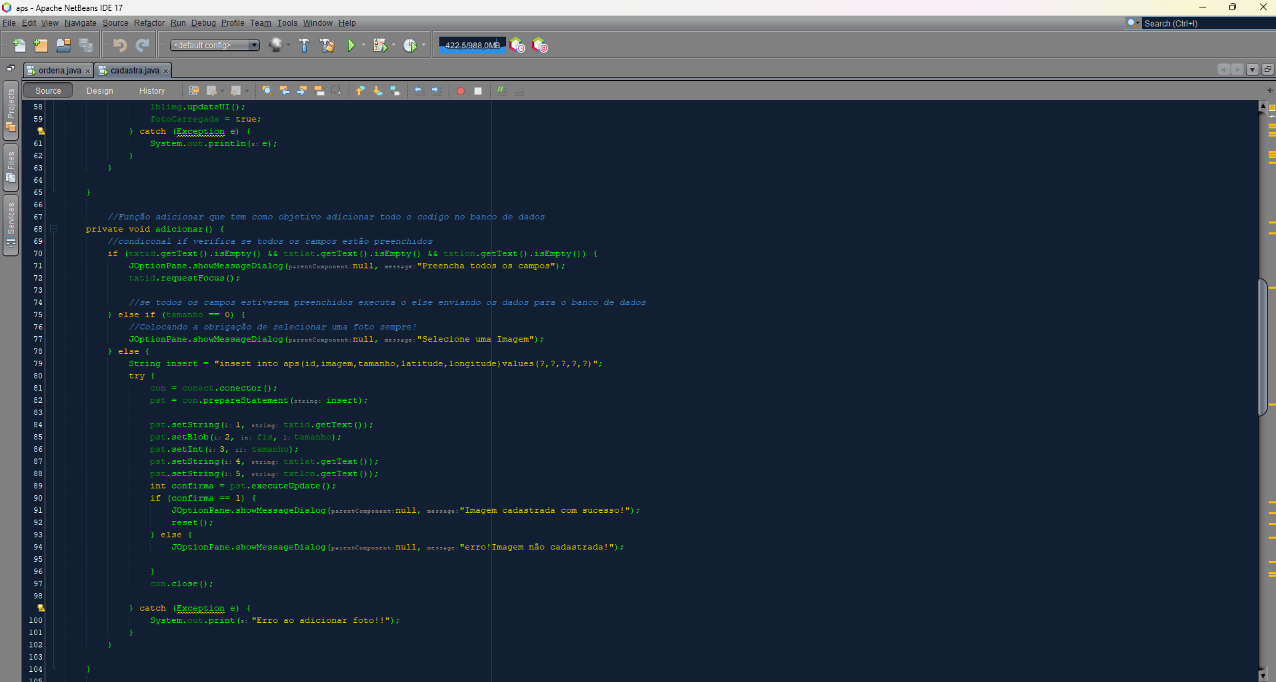
Dentro da pasta frames temos 3 frames sendo **cadastra, ordena, visualizar** que são geridos pelo frame **principal** que ele que o software se inicia.

* **Frame Cadastra**

**Print da tela visual (Swing)**

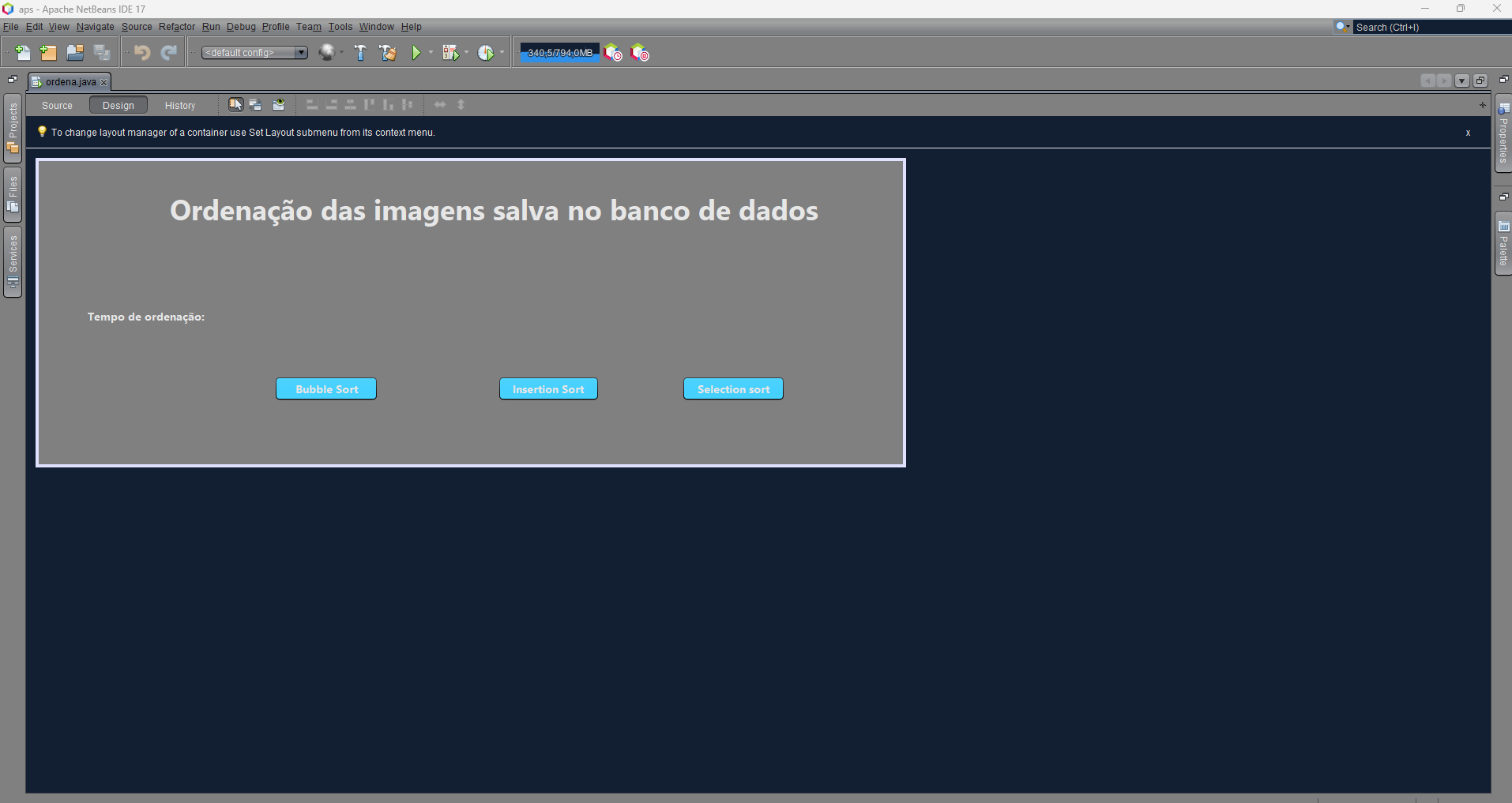
**Print da tela com codigo**

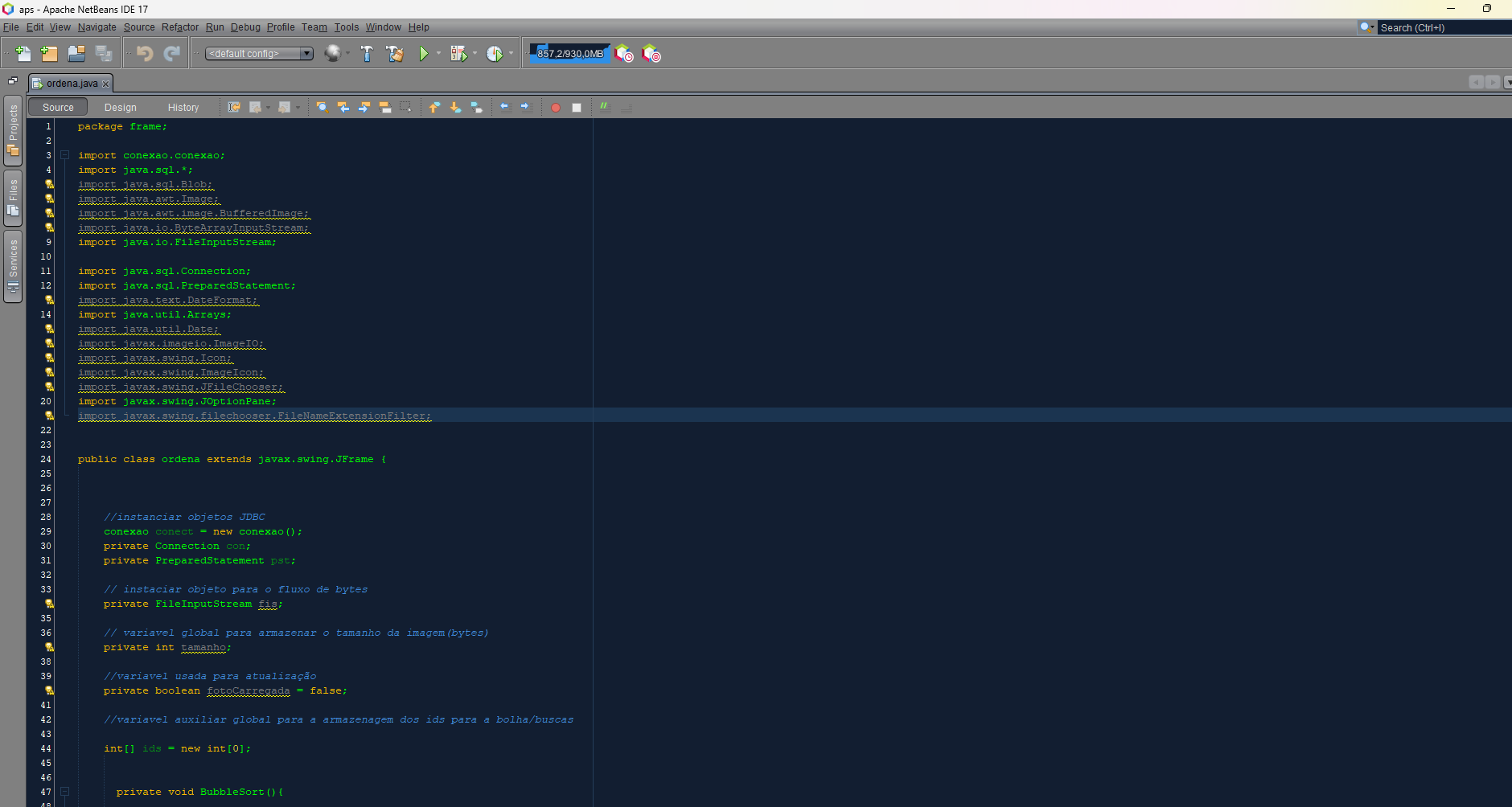
****

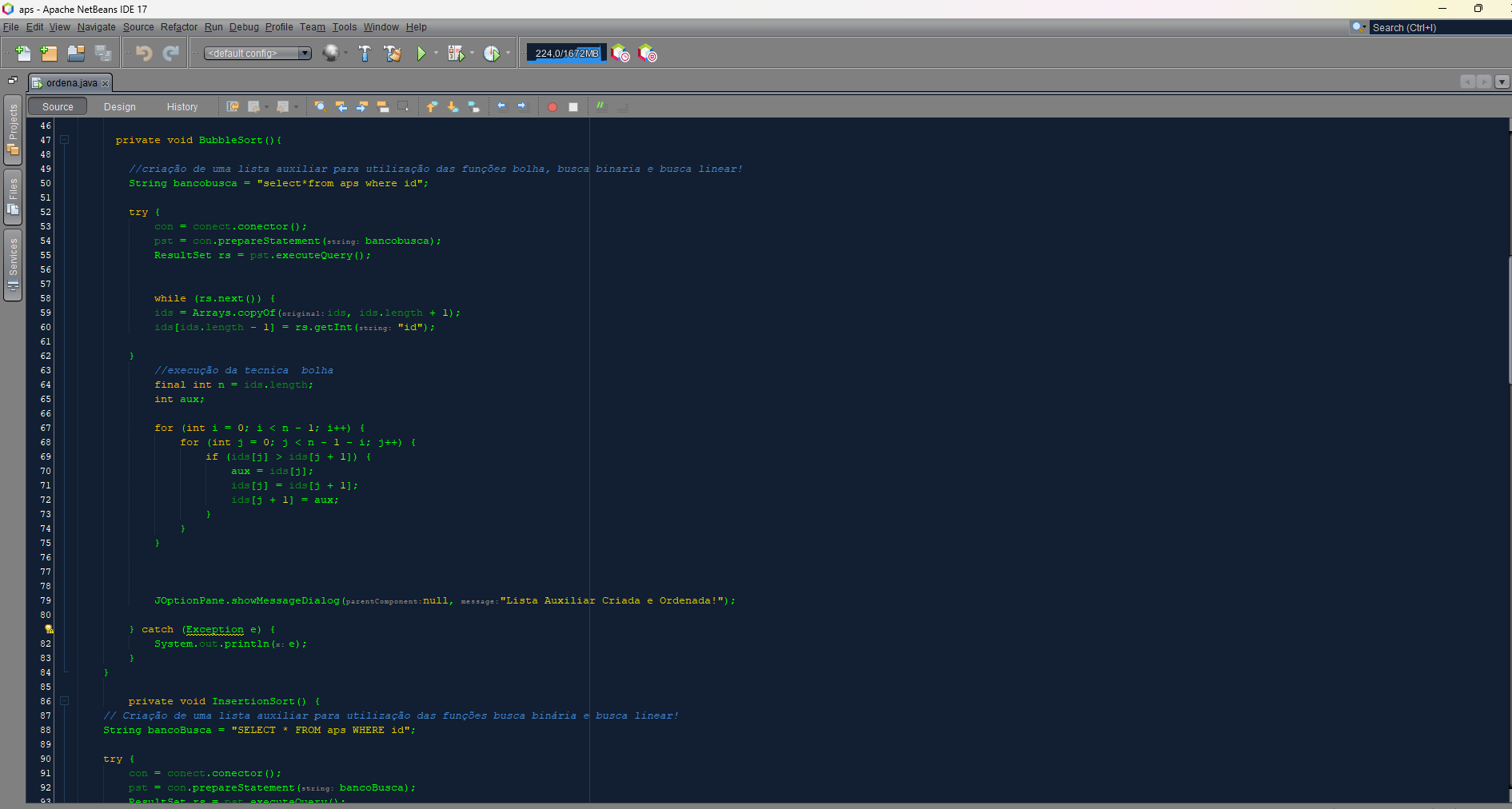
****

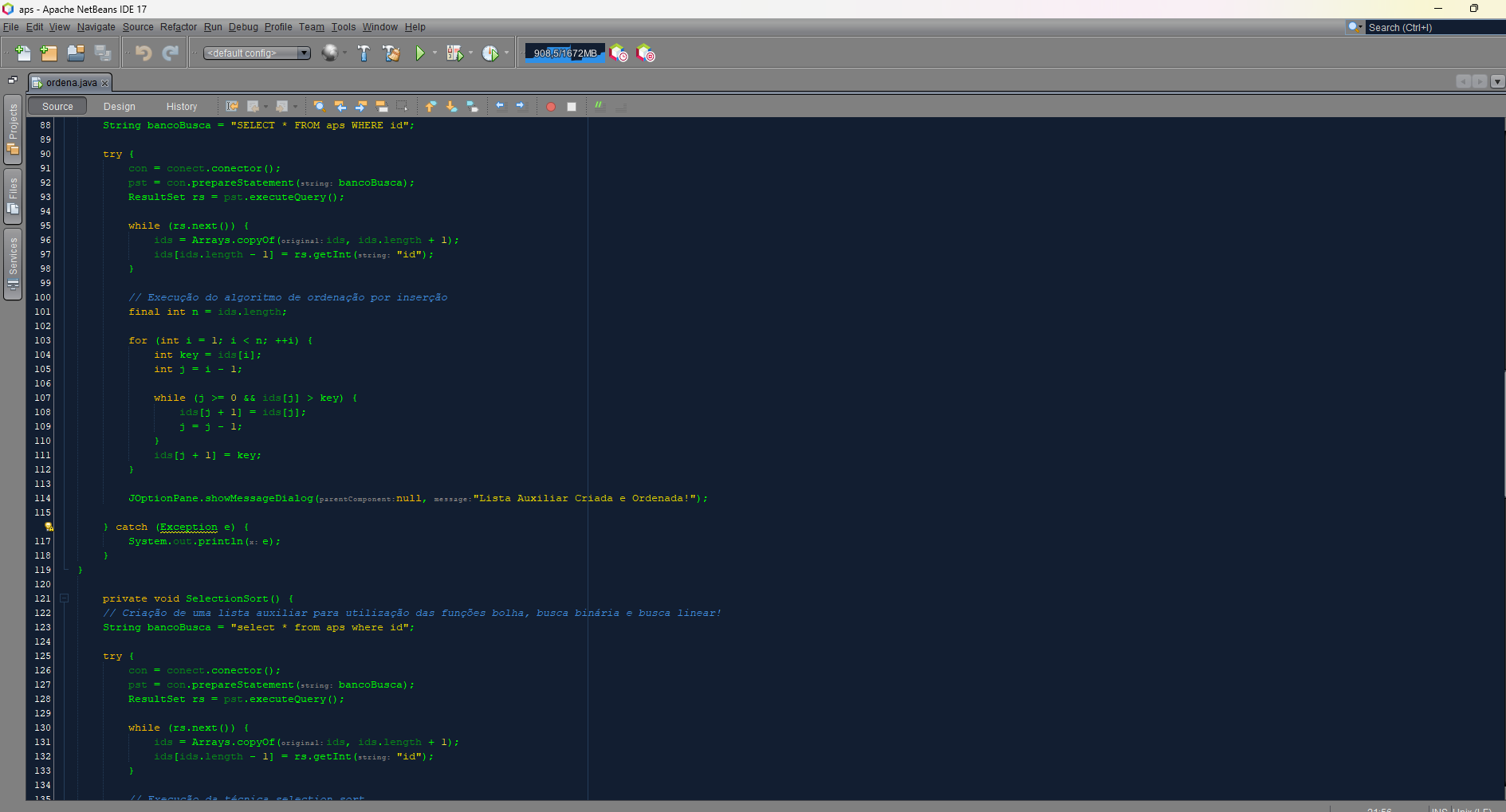
**Frame Ordena**

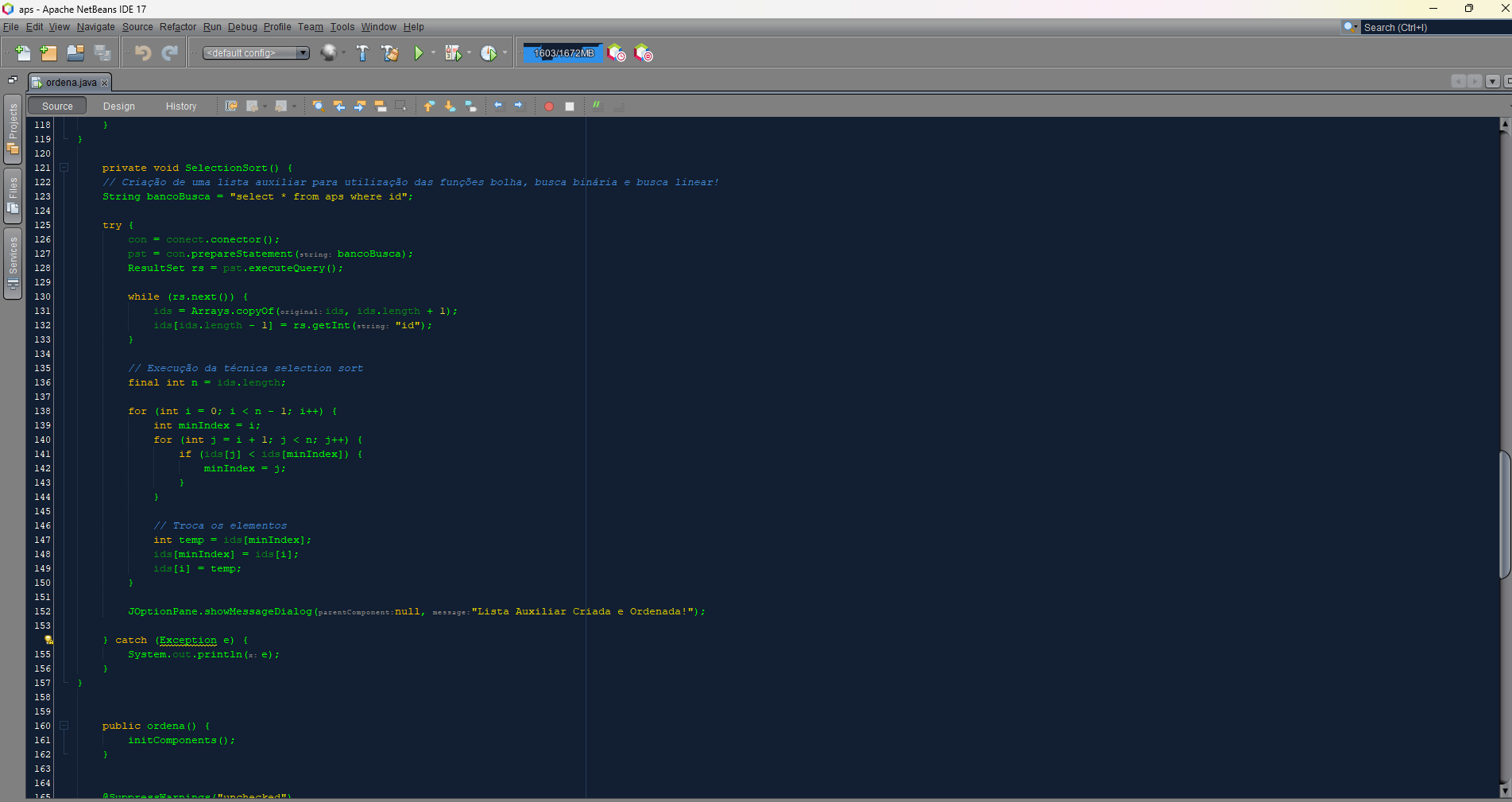
**Print da tela visual (Swing)**

****

**Print da tela com codigo**

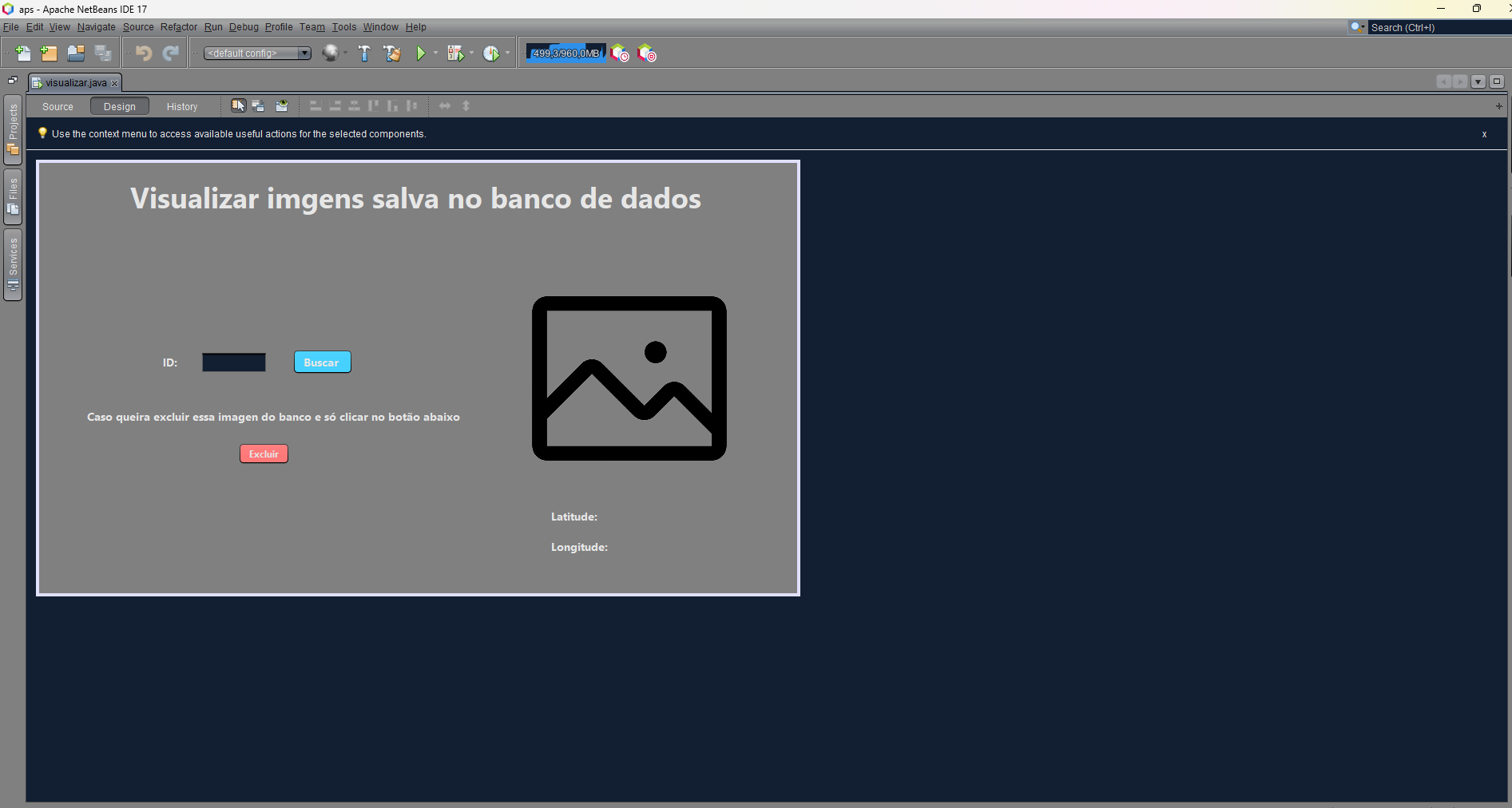
****

****

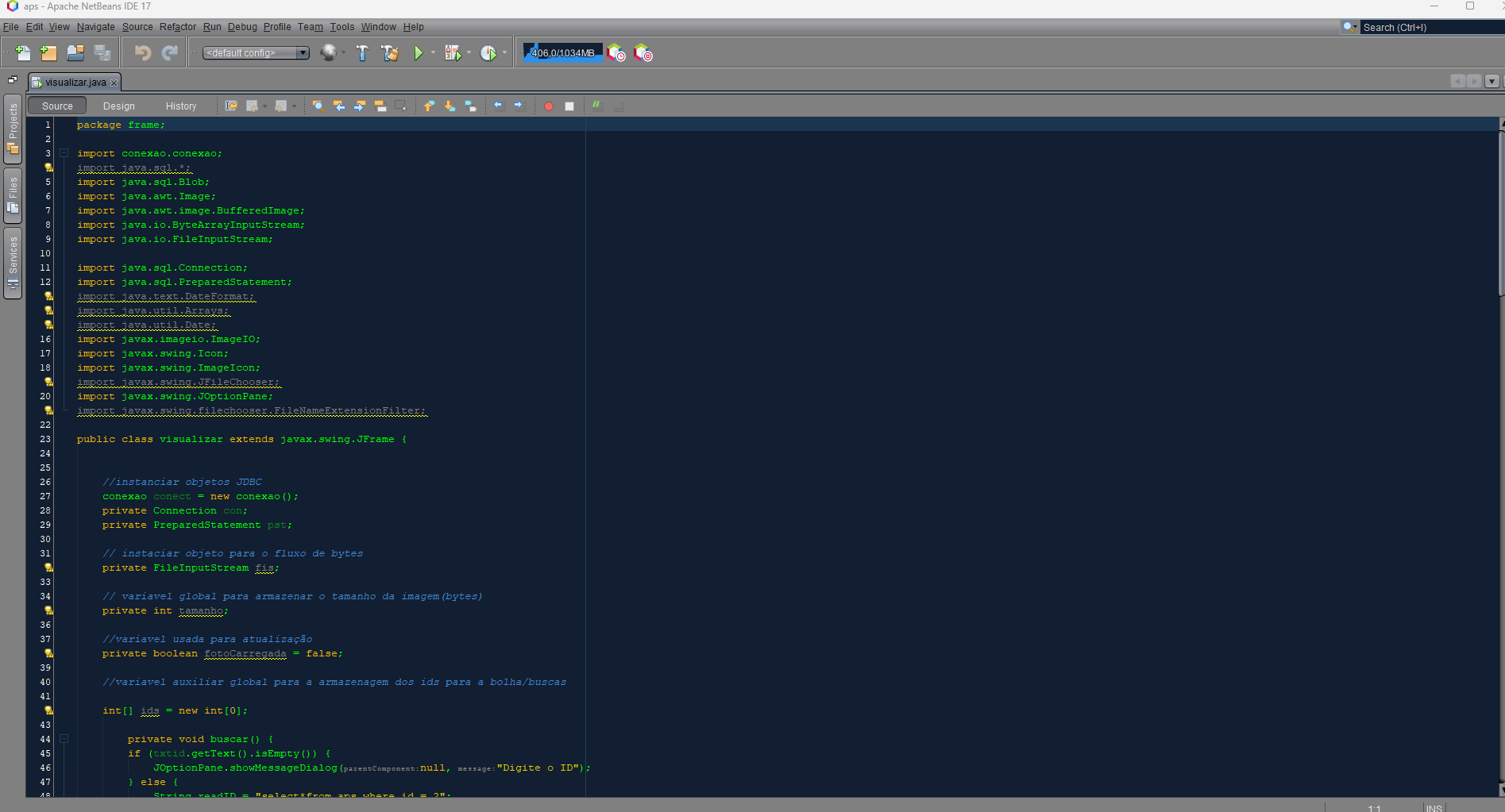
****

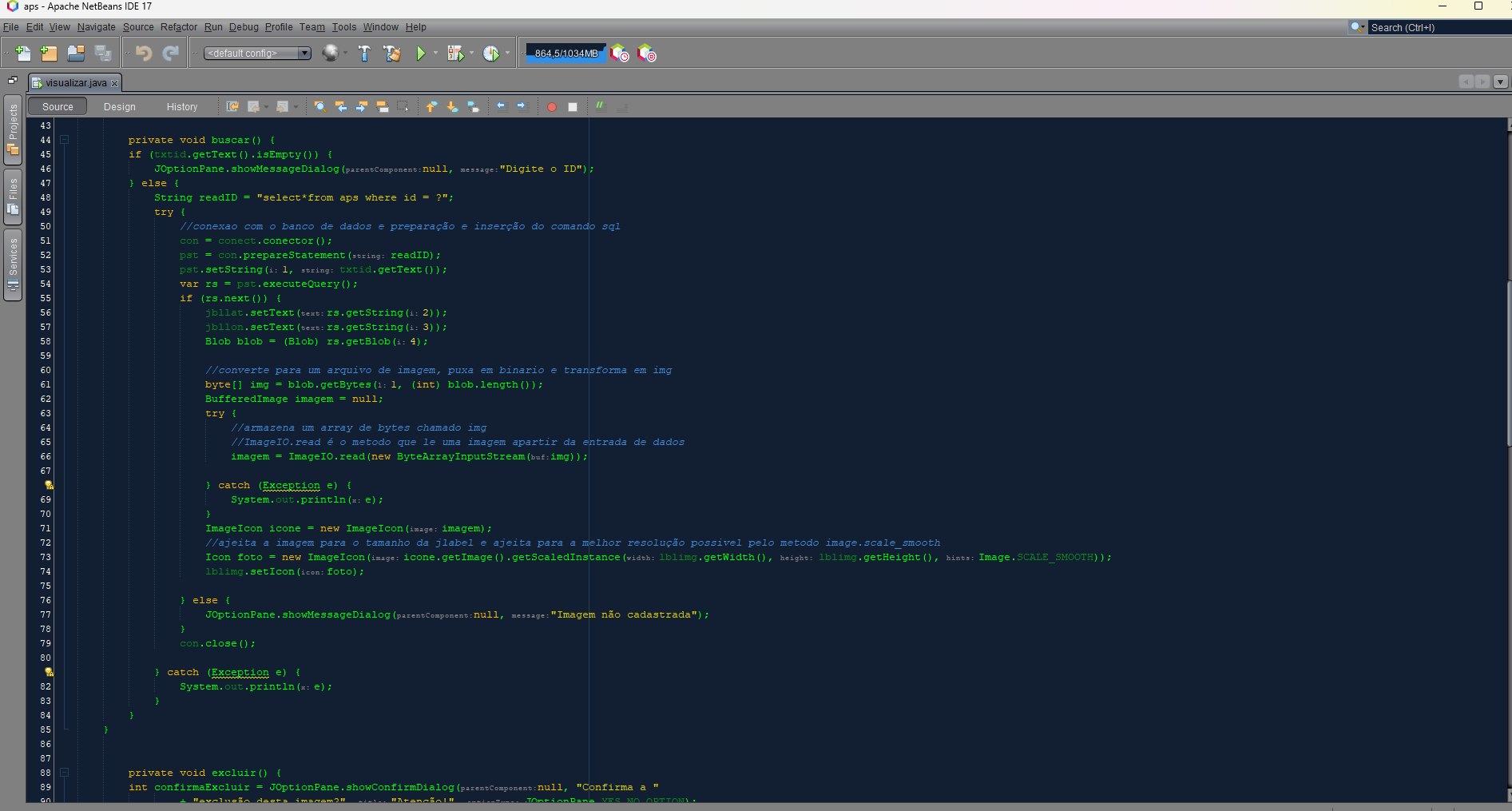
**Frame Visualizar**

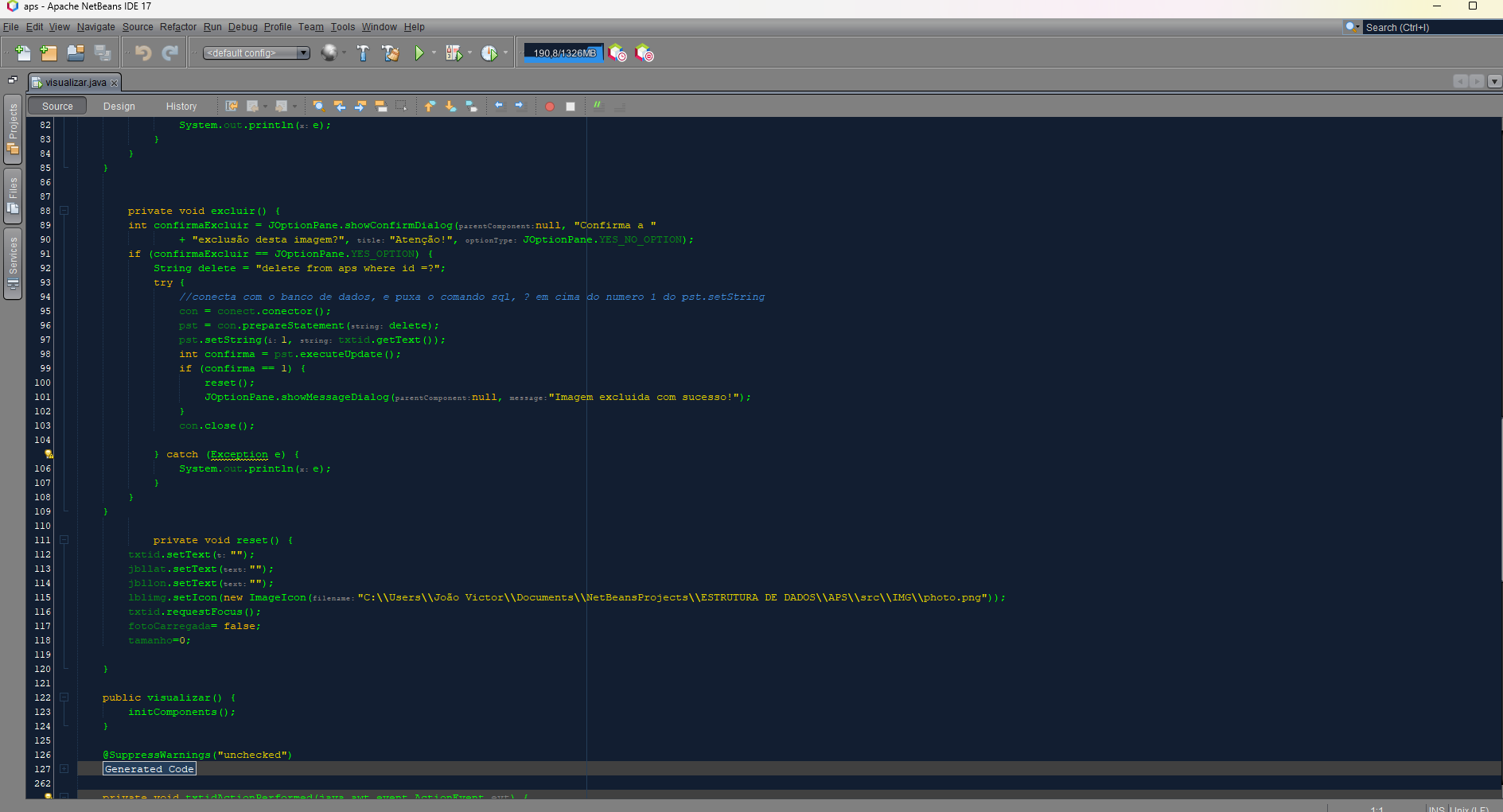
**Print da tela visual (Swing)**

****

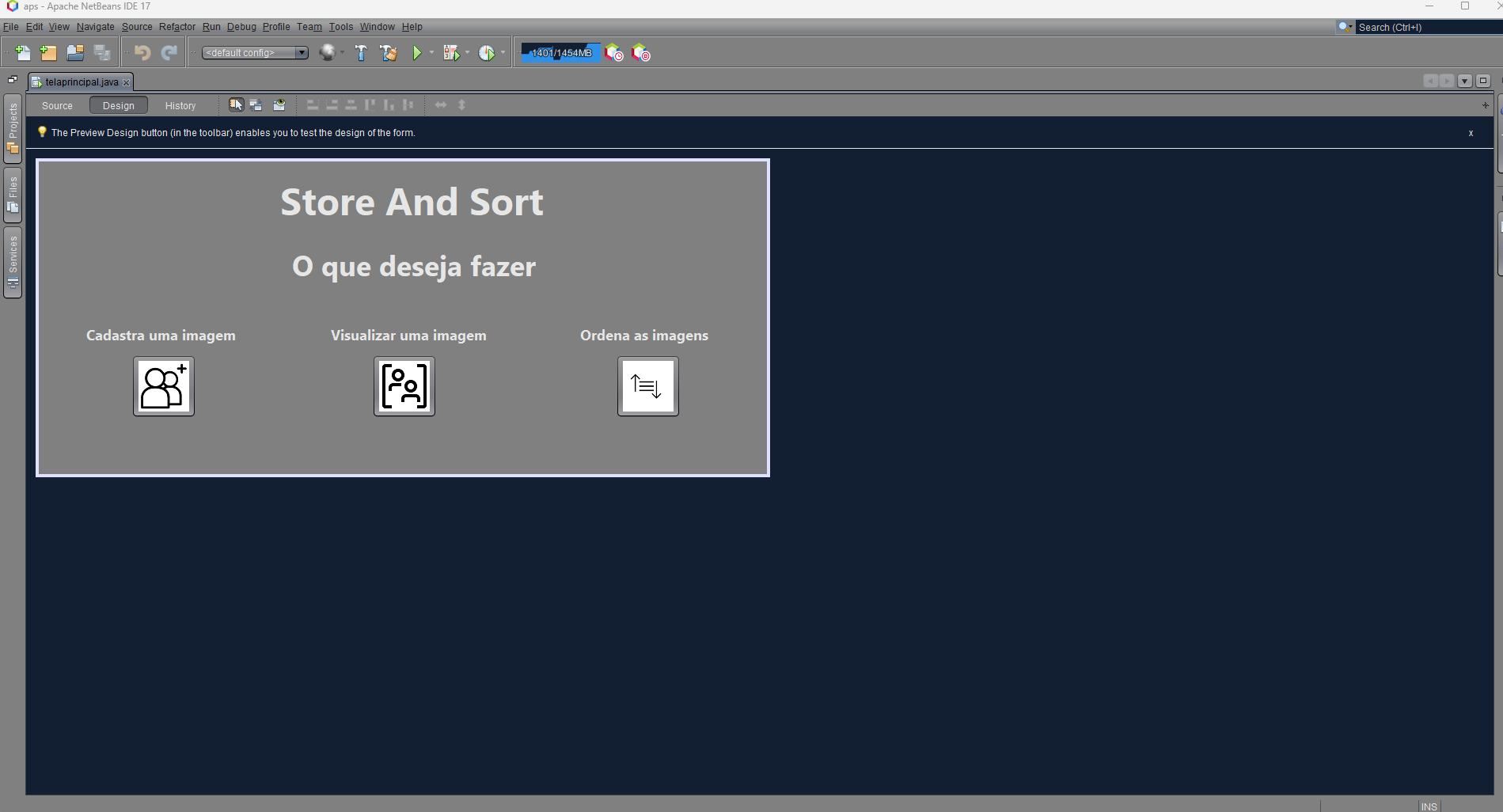
**Print da tela com codigo**

****

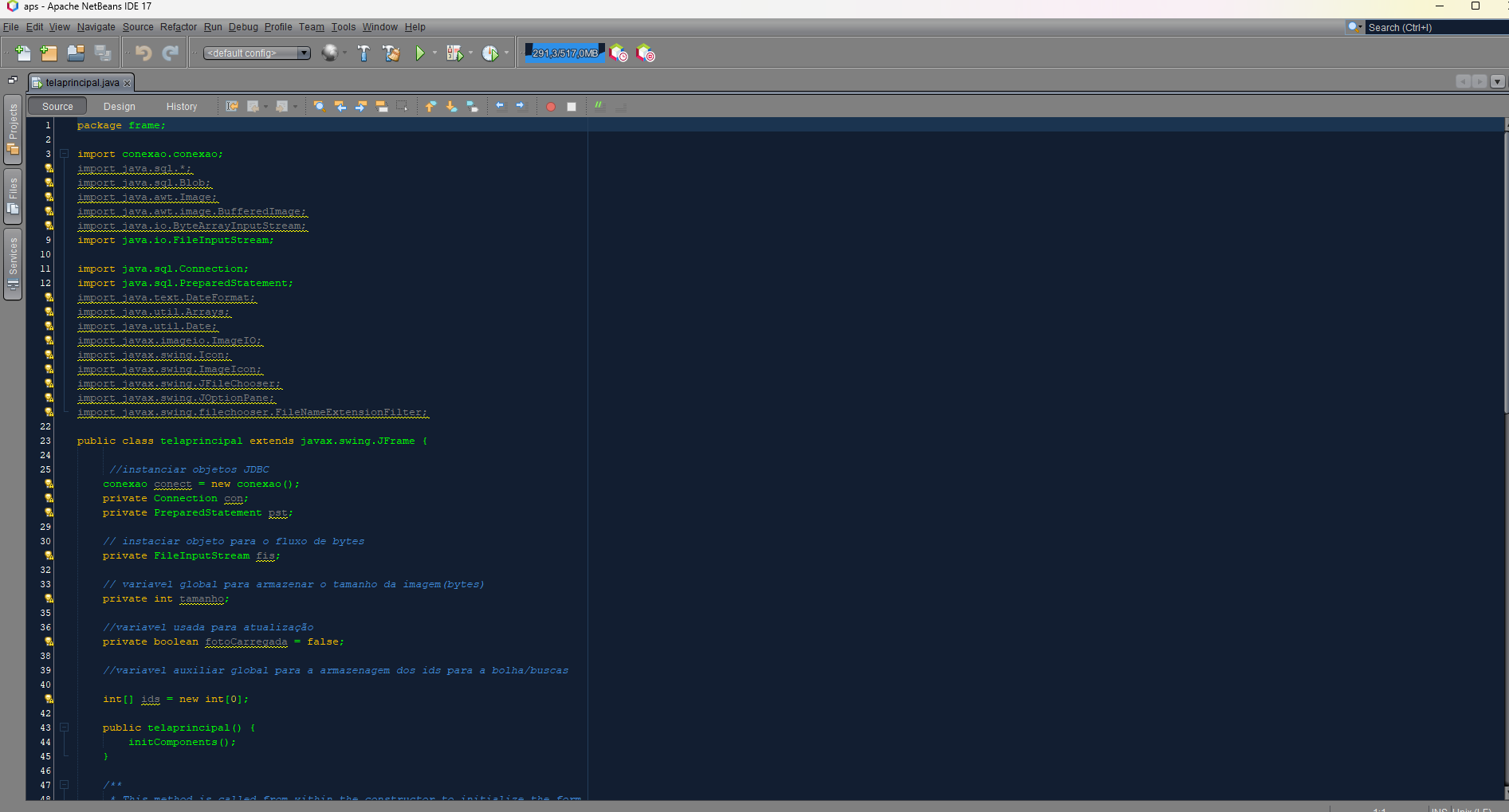
****

****

**Frame Telaprincipal**

**Print da tela visual (Swing)**

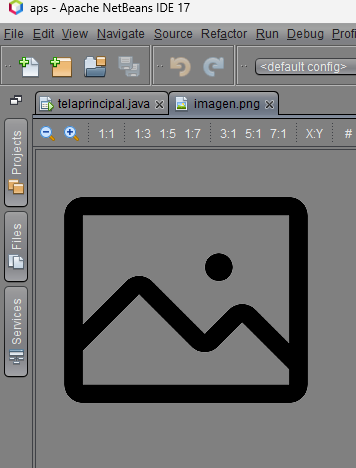
**Print da tela com codigo**

****

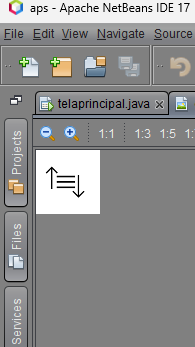
* **Pasta de Imgens**

Nesta pasta tera 4 imagens png que foi ultilizada no programa.

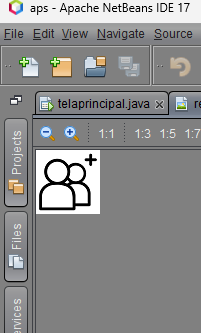
**IMG1**

****

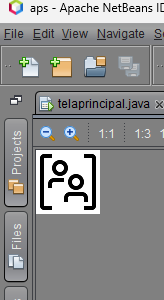
**IMG2**

****

**IMG3**

****

**IMG4**

****