**ATIVIDADE CONTEXTUALIZADA – PROGRAMAÇÃO E ESTRTURA DE DADOS**

Nome Completo: Luan Emerson Soares de Lima

Matrícula: 01462198

Curso: Análise e Desenvolvimento de Sistemas

**INTRODUÇÃO**

Segundo Markenzon (2010) estrutura de dados é uma forma de organização de informações, mais especificamente na memória, para a aplicação mais eficiente de algoritmos, exemplos de estruturas de dados são lista, árvore, grafo e Tabela Hash.

Cormen (2012) destaca que estruturas de dados buscam representar o mundo real, para uma posterior aplicação de algoritmo e resolução de problemas.

Como já citado anteriormente, uma das principais estruturas de dados para a resolução de problemas é a Tabela Hash. Essa Tabela é uma estrutura de dados especial, que armazena as informações desejadas as associando a chaves (GRONER, 2019).

No presente trabalho será utilizado a estrutura HashSet da API Collection para resolução de problemas.

**CONTEXTUALIZAÇÃO E PROBLEMATIZAÇÃO**

O problema s ser resolvido é o desenvolvimento de um programa que utilize a estrutura HashSet da API Collection e que adicione 10 mil elementos e remova 5 mil elementos.

**DESENVOLVIMENTO**

Para a resolução do problema foi utilizado o Apache Netbeans IDE. No Netbens o algoritmo foi escrito em java. O código do algoritmo pode ser visto abaixo.

**INÍCIO DO CÓDIGO**

package aplicacaohashset;

import java.util.HashSet;

public class AplicacaoHashSet {

public static void main(String[] args) {

int colocar = 10000;

int retirar = 5000;

HashSet<Integer> vetor = new HashSet<Integer>();

long pontoinicial = System.currentTimeMillis();

for (int i = 0; i< colocar; i++){

vetor.add(i);

}

for (int i = 0; i< retirar; i++){

vetor.remove(i);

}

long pontofinal = System.currentTimeMillis();

System.out.println( "Tempo em Milesegundos: "+ (pontofinal - pontoinicial));

System.out.println( "Tempo em Segundos: "+ (pontofinal - pontoinicial)/1000);

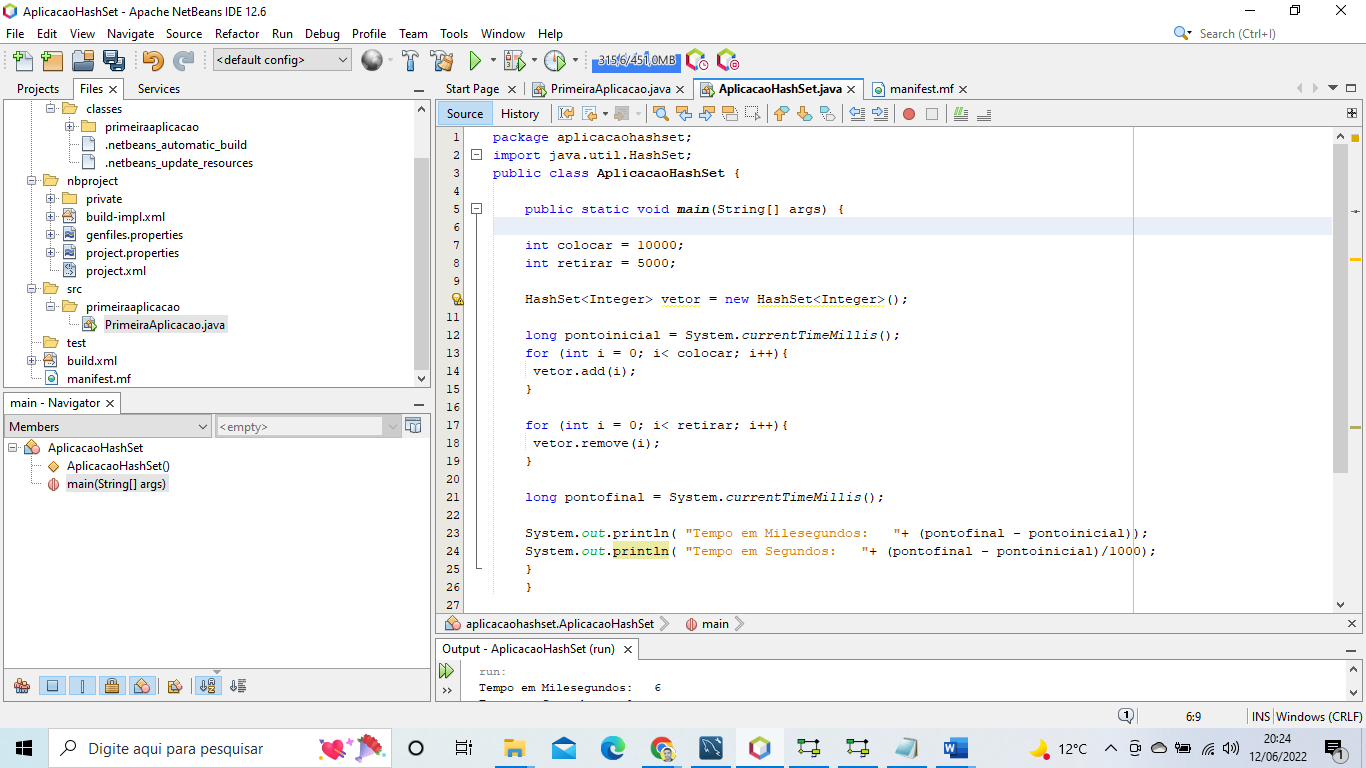
}

}

**FINAL DO CÓDIGO**

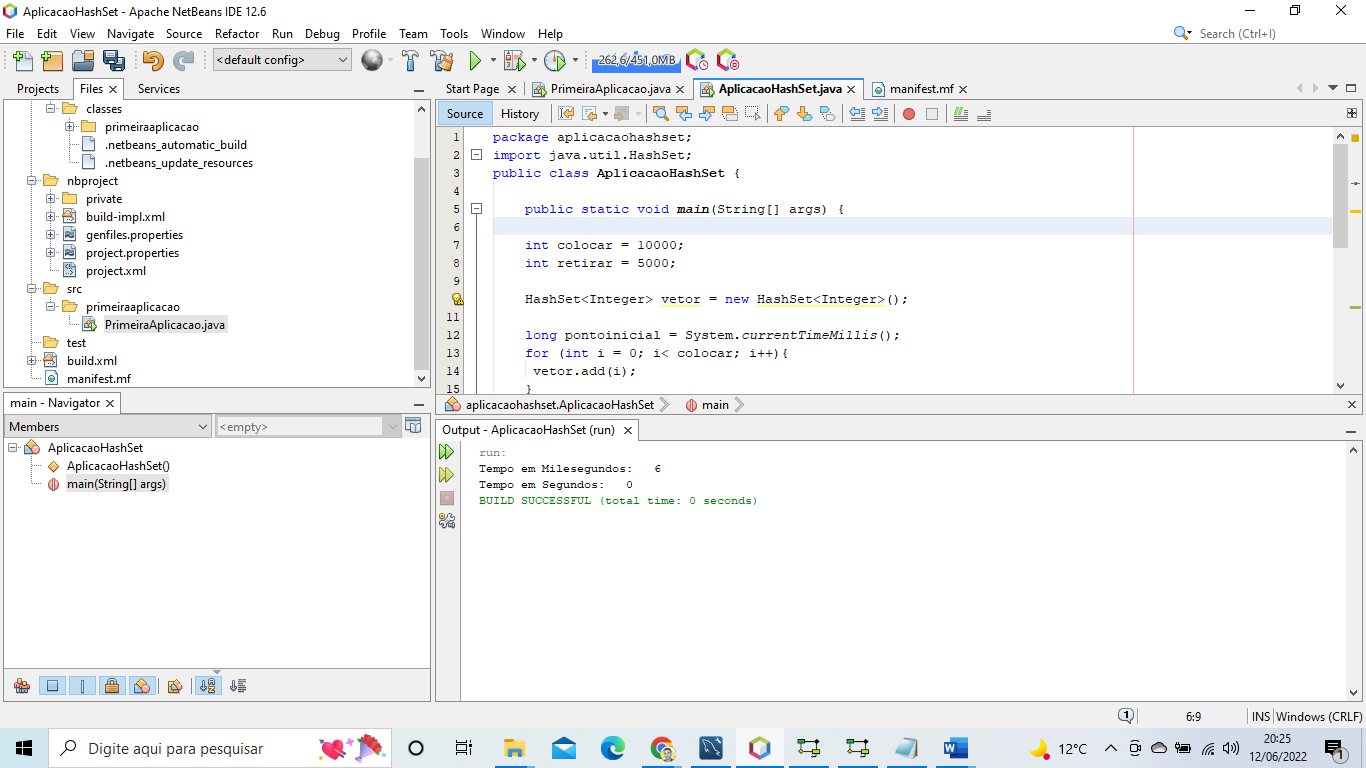
As Figuras 1 mostra a implementação do código no Netbeans e a Figura 2 o tempo de processamento em milissegundo e segundo.

Figura 1 - Aplicação do Algoritmo



Fonte: Autoria Própria (2022)

Figura 2 - Tempo de Processamento do Algoritmo



Fonte: Autoria Própria (2022)

**DESENVOLVIMENTO**

No presente trabalho foi aplicado o algoritmo HashSet da API Collection com um laço que adiciona 10 mil elementos e outro que remove 5 mil elementos. O tempo de processamento da operação foi computado antes e depois da operação. O tempo final foi subtraído do inicial e obteve-se o tempo de 6 millissegundo ou aproximadamente 0 segundo para o processamento da operação.

**REFERÊNCIAS**

CORMEN, T. **Algoritmos**: teoria e prática. 3 ed. São Paulo: LTC, 2012.

MARKENZON, L. Estruturas de Dados e Seus Algoritmos. 3 ed. São Paulo: LTC, 2010.

GRONER, L. **Estruturas de dados e algoritmos com javascript: escreva um código JavaScript complexo e eficaz usando a mais recente ECMAScript**. São Paulo: Novatec Editora, 2019.