Universidade Estadual de Santa Cruz – UESC

Departamento de Ciências Exatas e Tecnológicas – DCET

Curso de Ciência da Computação – CIC

**Proj1a**

Raphael Antônio Dalto Viana – 201911006

Professor: César Alberto Bravo Pariente

Matéria: CET087 – Conceitos de Linguagem de Programação

Sumário

[1 – Introdução 3](#_Toc113447519)

[2 – Baixando os arquivos 4](#_Toc113447520)

[3 – Compilando e executando os arquivos 5](#_Toc113447521)

[4 – Discussão dos resultados 6](#_Toc113447522)

[5 – Conclusão 7](#_Toc113447523)

# 1 – Introdução

O projeto consiste em uma análise de tempo de processamento do algoritmo Insertion Sort em dois paradigmas de programação distintos, um imperativo em C e um OO em JAVA.

A proposta é de gerar em ambos modelos, 1024 vetores de tamanho 100 contendo inteiros aleatórios, ordená-los usando o Insertion Sort e registrar o tempo de processamento. O processo deve ser repetido 1024 vezes e por fim deve-se gerar um gráfico com os tempos de processamento.

Afim de deixar ciente, os vetores foram randomizados em valores de 0 à 100 para facilitar a compreensão e também por conta do tamanho do vetor.

# 2 – Baixando os arquivos

Os arquivos se encontram no repositório do github abaixo, contendo os arquivos em C, Java e os .txt utilizados na montagem dos gráficos resultantes.

<https://github.com/Radviana/CET087-CLP>

# 3 – Compilando e executando os arquivos

Para compilar os programas, deveremos abrir o cmd na pasta contendo os arquivos e executar as seguintes linhas de comando:

* **gcc insertion.c** para o arquivo em C
* **javac insertion.java** para o arquivo em JAVA

Para executar os programas, deveremos abrir o cmd na pasta contendo os arquivos e executar as seguintes linhas de comando:

* **insertion.exe** para o arquivo em C
* **java insertion.java** para o arquivo em JAVA

# 4 – Discussão dos resultados

Após analisar os dados contidos nos arquivos de texto contidos na pasta “Teste para relatório”, podemos notar que para a quantia de dados e repetições propostas, o paradigma em OO se mostrou mais efetivo que o imperativo vide gráficos abaixo.

# 5 – Conclusão

Devido aos computadores atuais terem processadores cada vez melhores, o tempo de execução de ambos no final é irrisório para a quantia de dados dada. Pode se testar para quantias maiores ou vetores mais compridos para ver se essa diferença se torna mais expressiva.