

<div>Universidade Federal do Pará</div> <div>Pró-Reitoria de Ensino de Graduação – Diretoria de Ensino</div> <div>Instituto de Ciências Exatas e Naturais</div>				
CURSO:	Sistemas de Informação			
DISCIPLINA:	Estrutura de dados I			
DOCENTE:	Denís Lima do Rosário			NOTA:
ALUNO 1:	David Pinheiro Tavares	MATRÍCULA:	202311140028	
ALUNO 2:	Guilherme da Silva Sandim	MATRÍCULA:	202311140021	
TURMA:		DATA:	24/10/2024	

### **Trabalho sobre Estrutura de Dados (Vale 6 pontos)**

#### **Descrição da atividade**

A atividade deve ser realizada em dupla e entregue via SIGAA até o dia 22/10/2024. Apenas um dos alunos deve enviar o trabalho escrito pelo SIGAA, o qual deverá conter o link do trabalho no git com os nomes dos autores como comentário no início do código (se possível com código bem indentado e comentado). Trabalhos sem os nomes não serão considerados

#### **Questão (6.0 pts):**

O objetivo deste trabalho é entender o desempenho dos algoritmos de ordenação. O aluno deverá implementar um TAD para ordenar valores, onde a função principal deve ter as seguintes opções:

- uma opção de escolher a quantidade de valores inteiros de entrada dentre as opções: 10, 100, 1.000 e 10.000 elementos.
- uma opção de escolher como os valores de entrada estão dispostos antes da ordenação, ou seja, ordenados descendentes e/ou aleatórios.
- uma opção para apresentar os valores do vetor de entrada, ou seja, o vetor antes da ordenação.
- O programa deverá ter uma opção para a escolha do algoritmo de ordenação entre **três** algoritmos de ordenação, escolhendo conforme a regra:
  - Escolher um dos algoritmos: Bubblesort, Selectionsort, Insertionsort
  - Escolher um dos algoritmos: Mergesort, Quicksort, Heapsort
  - Escolher um dos algoritmos: radixsort, bucket sort e counting sort
- Durante a ordenação, o programa deverá contabilizar o número de comparações, número de trocas que realizam na ordenação de sequências, e o tempo de execução.
- Após a execução do algoritmo de ordenação, o programa deverá apresentar o vetor ordenado, bem como o tempo de execução, número de comparações, e número de trocas.

OBS: Para contabilizar o tempo de execução deve-se colocar contadores em seu código e instrumentá-lo de forma a obter o tempo de execução, ou seja, o temporizador deve ser configurado assim que iniciar o programa e concluído após inserir todos os elementos na estrutura.

Um dos alunos da dupla deverá gravar explicando o funcionamento do código, o qual deverá ser disponibilizado via google drive ou similar.