

 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA Campus de Vitória da Conquista</p>	DISCIPLINA	COMPLEXIDADE DE ALGORITMOS
	Nota máxima	10 Pontos
	Professor	Luis Paulo da Silva Carvalho
	Data de entrega	01/11/2024

1. Introdução

Este trabalho destina-se a avaliar os alunos da disciplina de **COMPLEXIDADE DE ALGORITMOS** em relação ao conteúdo e práticas apreendidos em sala de aula. Especificamente, deve ser criado um PROGRAMA e avaliar a sua **eficácia** e **eficiência** considerando as diferentes classes (ou níveis) de complexidade.

2. Escopo do trabalho

O trabalho deve ser realizado INDIVIDUALMENTE. Ele consiste na concretização de um PROGRAMA. Para tanto, inicialmente, deve ser proposto um mini-mundo, ou um tema, no qual o programa será baseado. É obrigatório que o mini-mundo obedeça aos seguintes critérios:

1. O mini-mundo seja na área de sensoriamento. Exemplo: um programa que realiza o processamento de dados sobre a temperatura corporal de pacientes sendo monitorados ou que consiga realizar cálculos sobre o número de pessoas entrando em um recinto através de várias catracas;
2. O sensoriamento deve ocorrer sobre algum dispositivo claramente definido. Exemplos: sensores de temperatura, sensores de batimentos cardíacos, catracas, etc.;
3. O mini-mundo deve realizar alguma computação sobre valores numéricos. Considerando os exemplos citados, respectivamente, valores de temperatura em *celsius* e total de pessoas que passaram pelas catracas;
4. O programa do mini-mundo deve gerar automaticamente seus próprios dados para testes. Exemplo: o programa de monitoramento de pacientes deve gerar uma coleção de dados de leituras de temperaturas de cada um dos pacientes, enquanto o programa das catracas pode gerar uma lista de pessoas que entraram pelas catracas.

3. Instruções gerais

As seguintes instruções devem ser atentamente observadas e cumpridas:

1. O aluno deve escolher o tema do seu trabalho e enviar para o meu e-mail (luispscarvalho@gmail.com) sua decisão entre os dias **10/10/2024** e **17/10/2024**. A partir daí deve acompanhar o andamento dos trabalhos. O acompanhamento visa identificar e resolver possíveis problemas e dificuldades de desenvolvimento;
2. Ainda aceitarei a escolha de um tema após o dia **17/10/2024**, todavia, já no dia **18/10/2024** (um dia após o prazo final para definição do tema) será descontado **1.0** (um ponto) da nota total e, para cada dia após **18/10/2024**, será descontado **0.5** pontos da nota final;
3. Se o total de descontos acumulados for igual ou superior a **5.0** (cinco pontos) não aceitarei mais um tema e não corrigirei a avaliação. Nesta situação o aluno terá sua nota **ZERADA**;
4. De forma alguma o programa desenvolvido por mim em sala de aula poderá ser reutilizado para realizar o seu trabalho. Ou seja, o mini-mundo escolhido **NÃO** pode conter a descrição ou visar o desenvolvimento de um programa que controle a temperatura/biometria de dados vitais de pacientes ou de algum(a) animal/coisa;
5. Não será permitida a repetição de um tema. Ou seja, a avaliação consistirá da escolha de um tema por cada trabalho a ser apresentado. À medida que os temas forem enviados para mim eu realizarei o controle para garantir a não repetição e exigirei um tema novo, caso o escolhido já tenha sido selecionado anteriormente por algum outro desenvolvedor ou que ele não corresponda ao que é esperado para avaliar os conteúdos da disciplina. Desta forma, quanto mais cedo um tema for definido e comunicado para mim através do meu e-mail menor deve ser a probabilidade de redefinição ou escolha de outro (tema);
6. É responsabilidade do aluno acompanhar a aprovação do seu tema através da adição do mesmo na planilha de controle de temas da avaliação, cujo link se encontra disponibilizado no classroom. A não entrada do tema na planilha significa que a avaliação não será corrigida e o aluno terá sua nota **ZERADA**;
7. O programa pode ser construído utilizando os recursos apresentados e exemplificados durante as aulas, na linguagem de programação JAVA. Todavia, se quiser e achar necessário, o desenvolvedor pode escolher outras tecnologias de codificação/desenvolvimento desde que tenham suporte nativo à criação e uso de THREADS. **ATENÇÃO:** Isso exclui, por exemplo, as linguagens JAVASCRIPT e TYPESCRIPT. Desta forma, programas escritos em tais linguagens serão desconsiderados e o aluno terá sua nota **ZERADA**;
8. Depois de criar o programa, deverá ser realizada uma captura de vídeo (máximo de 10 minutos) para explicar: (a) o

	DISCIPLINA	COMPLEXIDADE DE ALGORITMOS
	Nota máxima	10 Pontos
	Professor	Luis Paulo da Silva Carvalho
	Data de entrega	01/11/2024

tema escolhido e aprovado, (b) como realizar a configuração inicial e a execução do programa e (c) a visualização do resultado da execução. O vídeo deverá ser entregue para mim no dia de encerramento da avaliação. O vídeo pode ser enviado para o *youtube* ou para o *google drive*;

9. Também deverão ser entregues os arquivos-fonte: os *scripts* e/ou arquivos e/ou projetos, arquivos de configuração (caso existam) e quaisquer outros arquivos ou dependências necessárias para executar o programa;
10. Somente oferecerei atendimentos relacionados à avaliação até o dia **25/10/2024**. Isso evitará que eu tenha que dar suporte a desenvolvedores que, de forma indevida, mesmo após tantos avisos durante as aulas, demoraram de iniciar os seus trabalhos.

4. Critérios para a criação do programa

Em relação aos conteúdos da disciplina a serem abordados e ao código do programa a ser criado, devem ser observados os seguintes critérios:

1. Devem ser simulados os monitoramentos de 10 objetos, pessoas ou animais com seus respectivos dispositivos/equipamentos. Exemplo: sensores de batimentos de 10 pacientes ou 10 catracas;
2. As seguintes funcionalidades (funções ou métodos) devem ser implementadas: (a) gerar automaticamente (randomicamente) os dados dos sensores para qualquer quantidade N de dispositivos monitorados, (b) os dados gerados devem conter, obrigatoriamente, um coluna de identificação tanto para cada objeto, pessoa ou animal sendo monitorado, (c) um método *main*, principal, a partir do qual todas as funcionalidades possam ser executadas/testadas, (d) permitir os seguintes processamentos sobre os dados gerados no passo a:
 - (d.1) imprimir a lista de objetos, pessoas ou animais sobre os quais ocorre o monitoramento;
 - (d.2) imprimir a lista das leituras por cada objeto, pessoa ou animal sobre o qual ocorre o monitoramento.
 - ATENÇÃO:** deve ficar claro quais leituras pertencem a cada um;
 - (d.3) ordenação crescente dos dados considerando os valores lidos dos sensores para cada coisa monitorada;
 - (d.4) deve ser adicionada, a critério do desenvolvedor, uma funcionalidade extra, relativa ao mini-mundo escolhido, cuja complexidade seja ou $O(N^3)$ ou $O(2^N)$ ou $O(N!)$. **ATENÇÃO:** a funcionalidade de "busca de padrão" ilustrada no exemplo da sala de aula **NÃO** pode ser repetida no seu programa.
3. Não podem ser usadas bibliotecas externas ou internas da linguagem para automatizar as impressões, filtragens, ordenações e acumulações pedidas. Ou seja, devem ser criados ou pesquisados e adicionados ao programa códigos completos de todos os algoritmos envolvidos na automação das funcionalidades **d.1**, **d.2**, **d.3** e **d.4** acima;
4. Especificamente, para a funcionalidade, **d.4**, deve ser pesquisado ou desenvolvido um algoritmo completamente diferente dos que foram criados para os itens, **d.1**, **d.2** e **d.3**;
5. O código deve estar comentado. Cada comentário sobre cada método ou função deve informar qual é a complexidade do algoritmo envolvido na operação (**ATENÇÃO:** utilize a notação Big-O). Além de informar qual é a complexidade, deve dizer a razão, ou justificativa, de ser aquela complexidade e informar também quais são as consequências da complexidade para o código caso a entrada de dados seja muito grande;
6. Ao comentar o código da funcionalidade **d.4**, deve ser adicionada uma observação sobre se é possível que o algoritmo gere alguma situação de necessidade de processamento via *brute force* (força bruta). O comentário deve descrever qual(is) é(são) a(s) situação(ões);
7. Durante a captura do vídeo de demonstração, o desenvolvedor deve narrar e demonstrar o uso de todas as funcionalidades, com seus resultados impressos, e explicar qual é a classe de complexidade de cada uma.

5. Lista de itens a serem entregues

1. O tema que será desenvolvido. Envie a proposta para o meu email, luispscarvalho@gmail.com, e acompanhe o registro do seu tema na planilha disponibilizada via classroom;
2. O código-fonte do programa com todas as suas dependências;
3. Um vídeo contendo uma apresentação do tema e do funcionamento do programa.

6. Barema

	DISCIPLINA	COMPLEXIDADE DE ALGORITMOS
	Nota máxima	10 Pontos
	Professor	Luis Paulo da Silva Carvalho
	Data de entrega	01/11/2024

O cálculo da nota da avaliação obedecerá à fórmula:

NOTA DA UNIDADE I	
ITEM A SER ENTREGUE	VALOR
(a) Código-fonte do Programa FUNCIONANDO	8
(b) Vídeo de apresentação com TODOS os detalhes pedidos	2
Total (a + b)	10

ATENÇÃO (IMPORTANTE): a ocorrência de plágio (cópia de outros trabalhos) será punida, com o aluno obtendo uma nota ZERO. A ocorrência também será notificada à coordenação do curso.