

 INSTITUTO FEDERAL CEARÁ	CURSO ENGENHARIA DE COMPUTAÇÃO		DATA: 03/05/ 2021
	PROJETOS DE ESTRUTURAS DE DADOS USANDO C		1º semestre 2ª etapa
			Turno: Tarde
	DISCIPLINA: Estruturadas de Dados		Nota:
Professor(a): Ernani Leite			
Aluno (a):			

1. Informações gerais do projeto:

Este projeto tem por objetivo identificar aplicações práticas de estruturas de dados. Usando a linguagem C, cada equipe (máximo, dois alunos!) deve implementar uma biblioteca contendo no mínimo um algoritmo das estruturas de dados apresentadas em sala de aula Árvore e Grafos. Este projeto está previsto para ser construído em duas etapas. A primeira etapa contemplará os conteúdos referentes a Árvore. A segunda etapa contemplará os conteúdos referentes a Grafos. Ambas as etapas constarão de apresentação online ou vídeo conforme abaixo detalhadas.

2. Descrição geral dos temas de trabalho e outras informações:

Cada equipe deverá construir uma biblioteca única e diferente das demais equipes. Cada tema será atribuído a uma equipe por meio de validação do Professor. No geral, conforme dito anteriormente, cada biblioteca irá modelar algum algoritmo das duas estruturas de dados, para compor a nota N2.

3. Avaliação dos Projetos:

O projeto valerá 5,0 pontos cada implementação (Árvore/Grafos) na composição da N2. Cada projeto deverá ser defendido online ou enviado vídeo pela equipe. Ambos os integrantes devem explicar alguma parte do projeto igualmente distribuída. No geral, cada equipe terá, no máximo, 20min para apresentar sua solução! A apresentação pode ser considerado online ou o envio de um vídeo pela equipe.

Vale lembrar que o projeto não é opcional, ou seja, se a equipe resolver não entregar o projeto ficará com ZERO referente a composição da nota N2 da etapa! Não haverá uma nova data para defesas de projetos. Além disso, se um dos integrantes não apresentar sua parte para a turma, ficará sem esta nota – não há reposição de projetos.

Importante: na apresentação, a nota de cada aluno não será dada pelo desempenho da equipe como um todo, mas sim pelo desempenho individual de cada aluno na hora da apresentação. Assim, se um integrante da equipe não explicar nada (ou quase nada) durante a apresentação, ficará com ZERO (ou uma nota muito baixa).

Os temas dos projetos serão avaliados, considerando:

- **Implementação:**

- a) A equipe conseguiu explicar a parte relevante do código do algoritmo para a turma?
- b) Utilizou conceitos de modularização e reuso com bibliotecas (o Main consegue importar sua biblioteca e utilizá-la normalmente)?
- c) O algoritmo funcionou corretamente?

- **Apresentação:**

- a) A equipe começou sua apresentação na hora combinada?
- b) Os dois integrantes participaram?
- c) A equipe conseguiu apresentar tudo em 20 min?
- d) O programa apresentou algum tipo de erro durante a apresentação (travamentos, situações de erros, etc.)?

4. Data das defesas: cada tema terá uma data e ordem de apresentação associada, sempre no horário da aula, com início das apresentações dia 31 de maio (Árvores) e 30 de junho (Grafos).

5. O que sua equipe deve entregar ao final da apresentação?

Ao final de sua apresentação, sua equipe deve entregar o código fonte e executável do programa num arquivo zipado e **postado no Google Classroom**. Deverá conter o arquivo zipado bem como a informação dos membros da equipe.

Exemplos de aplicações:

☐ **ÁRVORES DE BUSCA BINÁRIA**

- Efetuar buscas em Pré-Ordem, Pós-Ordem ou Em-Ordem.
- Exemplos: Calculadora para expressões matemáticas; Árvores genealógicas;

☐ **GRAFOS**

- Encontrar o menor caminho através dos vértices do grafo (Caixeiro Viajante);
- Exemplos: Rota para a coleta de lixo da Fortaleza; Rota de um Uber na cidade; Roteamento para uma rede de computadores, entre outros.

Para refletir...

A linguagem C opera muito próxima ao hardware. Se, por um lado, isto gera uma dificuldade extra para seu aprendizado, por outro, programando em C, temos um bom conhecimento de como o computador funciona, e este conhecimento é fundamental para programar bem em qualquer linguagem. Em C, somos obrigados a gerenciar explicitamente a memória que alocamos, podemos manipular diretamente endereços de memória, precisamos entender o conceito de passagem de parâmetro por valor e por referência, aprendemos o conceito de variáveis do tipo ponteiro etc. Adquirimos um conhecimento que nos permite escrever programas corretos e eficientes.

Este conhecimento é particularmente importante quando precisamos projetar novas estruturas de dados para atender demandas específicas. Linguagens de programação mais abstratas oferecem suporte nativo às estruturas de dados mais usuais; em C, precisamos implementá-las. O conhecimento adquirido ao implementar as estruturas nos faz ter um melhor discernimento de qual estrutura devemos usar numa determinada aplicação, nos capacita a adaptar estruturas já existentes quando necessário, e nos ensina a avaliar a eficiência dos algoritmos que fazem uso dessas estruturas.

Além do conhecimento adquirido, o aprendizado de C se justifica por si só. A linguagem C é ubíqua, estando presente em todas as plataformas computacionais, desde supercomputadores a processadores ARM embutidos. Por isso, escrever um código padrão C é escrever um código portátil; além disso, ganhamos em versatilidade, pois programamos desde sistemas embarcados em hardware simples a sistemas computacionais complexos. A linguagem C nos permite fazer diferentes aplicações, escritas em diferentes linguagens, interoperarem entre si. A grande maioria das linguagens de programação provê suporte a programação de interfaces em C. A linguagem C é a “língua franca” de programação de computadores.

Lembre-se: *você só irá aprender algo em Estruturas de Dados se trabalhar duro! Bons estudos!*

***“A única segurança que o homem pode ter na vida
é a sua reserva de conhecimento”***
(Henry Ford)