



INSTITUTO FEDERAL

Espírito Santo
Campus Guarapari

INSTITUTO FEDERAL DO ESPÍRITO SANTO – CAMPUS GUARAPARI

| | | |
|---|--|------------------------------------|
| Curso: Bacharelado em Engenharia Elétrica | Disciplina: Algoritmos e Estruturas de Dados | |
| Professor: Cayo Magno da Cruz Fontana | Turma: 20251.ENEL.2D | Valor: 40,00 pontos (+20,00 EXTRA) |
| Data da Entrega: 26/06/2025 | Datas das Apresentações: 27/06, 03/07 e 04/07/2025 | |

TS – TRABALHO SEMESTRAL

1. ESPECIFICAÇÃO

A Coordenadoria de Registros Acadêmicos do IFES – Campus Guarapari – solicitou aos alunos da disciplina de Algoritmos e Estruturas de Dados, do curso de Bacharelado em Engenharia Elétrica, o desenvolvimento de um sistema de gestão para os cursos do campus.

Para melhor compreensão das atividades a serem desenvolvidas pelos programadores, o analista de sistemas definiu todas as funcionalidades que o sistema deverá prover aos usuários (disponível no item 2. deste documento) e solicita a você, programador, que implemente estas operações de acordo as necessidades da instituição.

Deve-se construir um programa utilizando a linguagem de programação C, aplicando os conceitos e técnicas de programação estruturada e gerenciamento de memória.

Este trabalho deve ser realizado pelo(s) aluno(s), individualmente ou em grupos de, no máximo, 05 (cinco) componentes.

Ao final, a solução construída deverá gerar um sistema (programa) com um comportamento semelhante ao que é apresentado no seguinte vídeo:

- <https://youtu.be/omwU7bsf6ss>

2. ATIVIDADES

Leia as operações a serem providas pelo sistema proposto e implemente-as conforme solicitado no item 1. deste documento.

Abaixo estão listadas as operações (funcionalidades) que o sistema **deve fazer**:

1. Inserir turmas. Cada turma deverá possuir, pelo menos, as seguintes informações:
 - a) número (sugestão: utilizar o tipo inteiro para esta informação);
PS: esta informação deve ser unívoca, ou seja, exclusiva para cada registro.
 - b) lista de disciplinas.
2. Inserir disciplinas. Cada disciplina deverá possuir, pelo menos, as seguintes informações:
 - a) código (sugestão: utilizar o tipo inteiro para esta informação);
PS: esta informação deve ser unívoca, ou seja, exclusiva para cada registro.
 - b) nome (sugestão: utilizar o tipo ponteiro de caractere para esta informação);
 - c) lista de alunos.
3. Inserir alunos. Cada aluno deverá possuir, pelo menos, as seguintes informações:
 - a) matrícula (sugestão: utilizar o tipo inteiro para esta informação);
PS: esta informação deve ser unívoca, ou seja, exclusiva para cada registro.
 - b) nome (sugestão: utilizar o tipo ponteiro de caractere para esta informação);
 - c) telefone (sugestão: utilizar o tipo inteiro longo para esta informação);
4. Remover uma determinada turma. Para tal, o usuário deve informar o número da turma a ser removida da lista.

PS1: esta operação, necessariamente, deve reduzir o tamanho da lista de turmas (sugestão: realocação dinâmica da memória utilizada).

PS2: esta operação deverá, necessariamente, liberar a memória da turma removida.

5. Remover uma determinada disciplina. Para tal, o usuário deve informar o código da disciplina a ser removida da lista.

PS1: esta operação, necessariamente, deve reduzir o tamanho da lista de disciplinas (sugestão: realocação dinâmica da memória utilizada).

PS2: esta operação deverá, necessariamente, liberar a memória da disciplina removida.

6. Remover um determinado aluno. Para tal, o usuário deve informar a matrícula do aluno a ser removida da lista.

PS1: esta operação, necessariamente, deve reduzir o tamanho da lista de alunos (sugestão: realocação dinâmica da memória utilizada).

PS2: esta operação deverá, necessariamente, liberar a memória do aluno removido.

7. Listar todas as turmas.

8. Listar todas as disciplinas.

PS: necessário informar a turma (através da sua informação unívoca) para que as disciplinas sejam listadas.

9. Listar todos os alunos.

PS: necessário informar a disciplina (através da sua informação unívoca) para que os alunos sejam listados.

10. Após o usuário sair do sistema, toda a memória alocada dinamicamente deverá ser liberada.

11. ITEM EXTRA (20,00 pontos): A cada cadastro de turma, disciplina e aluno, o sistema deverá persistir (armazenar) suas respectivas informações em um arquivo em memória não-volátil (txt).

PS 1: ao remover uma turma, todas as disciplinas associadas (e seus respectivos alunos associados), também serão removidos.

PS 2: ao remover uma disciplina, todos os alunos associados a disciplina serão removidos.

3. SUBMISSÃO

Os trabalhos deverão ser submetidos, eletronicamente, no AVA da disciplina.

Um arquivo compactado, agrupando todos os arquivos de código fonte (ou demais tipos, caso necessário) do projeto. O arquivo compactado poderá ter as extensões: 7z, zip, rar, gzip, bzip2 ou tar.

NOTA1: não incluir arquivos de ambientes de desenvolvimento – IDE's (como extensões .proj, .sln, .suo, entre outros) e nem arquivos executáveis.

PS 1: a cada dia de atraso, a nota do trabalho terá 8,0 (oito) pontos reduzidos. Ao final do 5º dia de atraso, o trabalho não valerá nota.

PS 2: trabalhos iguais receberão nota zero!

NOTA: plágio é crime, previsto em legislação, e caracteriza uma conduta corrupta do cidadão! Contribua para uma sociedade menos corrupta! Diga NÃO ao plágio!!

4. APRESENTAÇÃO

O trabalho será apresentado em sala de aula (sala de informática), no formato pergunta–resposta entre o professor e o grupo, em uma das possíveis datas definidas neste documento (sorteio realizado em sala de aula). Contudo, o professor SEMPRE direcionará a pergunta para um componente (aluno) do grupo, por vez! Caso outro(s) aluno(s) manifeste(m)-se durante a entrevista de um componente específico, sem que esta interlocução seja previamente autorizada pelo entrevistador (docente), o grupo inteiro perderá ponto.

Os alunos utilizarão o computador do professor e apresentarão seus respectivos trabalhos no datashow. Os discentes não poderão utilizar quaisquer outros meios de consulta. As perguntas serão exclusivamente sobre seus respectivos trabalhos que já estarão explicitamente projetados em sala, pelo datashow, no momento da sua apresentação.

NOTA: não poderá, em hipótese alguma, conter quaisquer comentários no código fonte deste trabalho. Caso haja, toda a nota de apresentação será invalidada pois, pela natureza da avaliação, a prática caracteriza fraude.

5. AVALIAÇÃO

O trabalho será tecnicamente avaliado levando em consideração a realização de todas as tarefas solicitadas no item 2. deste documento, enviado no AVA da disciplina, até a data limite de submissão. Nesta etapa, o programa deverá estar compilando, executando e realizando todas as operações solicitadas no item 2. desta publicação.

No dia da apresentação o grupo deverá compilar, executar e testar as operações do programa. Para este último, o professor selecionará, aleatoriamente, quais operações serão desejadas para o teste.

Durante a entrevista, o entrevistador (professor) escolherá, aleatoriamente, alunos do grupo para responder a cada pergunta. A resposta do aluno será avaliada e pontuada para todo o grupo. O professor poderá oportunizar a um outro aluno (que também será escolhido aleatoriamente) para responder a pergunta que eventualmente não foi respondida pelo primeiro componente (ou que tenha sido respondida incorretamente), ou poderá passar para uma próxima pergunta para o mesmo ou outro componente do grupo.

6. PONTUAÇÃO

A pontuação deste trabalho será dividida da seguinte forma:

- Submissão do trabalho no AVA até a data limite de submissão. (Valor: 16,00 pontos);
 - Programa compilando com sucesso;
 - Programa executando com sucesso;
 - Programa realizando todas as operações que serão testadas aleatoriamente.
- No dia da apresentação, serão avaliados os seguintes quesitos (Valor: 24,00 pontos):
 - Respostas corretamente claras, coesas e objetivas acerca do que será questionado na entrevista, pelo entrevistador (docente).