

Projeto final

O projeto final da disciplina de Algoritmos e Programação Estruturada consiste no desenvolvimento de um sistema de cadastro que equivale a 4 pontos na média final. A implementação de cada projeto deve ser feita em grupos de 5 ou 6 pessoas e o versionamento do código deve ser feito pelo GitHub. **O projeto a ser desenvolvido por cada grupo será sorteado em sala pelo professor na aula do dia 16/11.**

Os projetos desenvolvidos pelos alunos devem ser apresentados também em sala de aula, no horário da aula, na data provável de **07/12**. **Porém, a data limite de entrega é dia 06/12, até às 23:59. O envio deve ser feito pelo AVA e deve conter um link para uma pasta do GitHub contendo o código que implementa o jogo sorteado.**

O envio deve ser feito apenas pelo membro escolhido como líder do grupo, que deve ser informado ao professor, juntamente com os membros do grupo, na aula do dia 16/11.

Critérios de avaliação

A avaliação será baseada em dois tipos de critério: coletivos e individuais. Os critérios coletivos são:

- Implementação das 7 funcionalidades do projeto, exibindo o comportamento esperado em cada uma delas (1.5 + **0.25**);
- Uso dos conceitos (funções, structs e alocação dinâmica) (0.5);
- Interação com o usuário e “interface gráfica”/menu do projeto (opções no menu, clareza, interação com o usuário, etc.) (**0.25**);
- Organização e qualidade do código (0.4);
- Organização do GitHub (README, comentários e tamanho dos commits, etc.) (0.1);

Já os individuais são:

- Participação no desenvolvimento do projeto (quantidade e qualidade dos commits, proatividade, participação nas aulas de implementação, etc.) (1);
- Participação na apresentação final do projeto (explicação das funcionalidades/código) (0.5).

Soluções com erros de sintaxe (falta de “;”, nome de funções ou tipos escritos errados, etc.) ou de execução (variáveis indefinidas, escolha do identificador de tipo inadequado, etc.) serão penalizadas e receberão nota 0. **Portanto, testem o código antes de enviá-lo.**

Plágio não será tolerado! Grupos com projetos que tenham códigos claramente iguais ou com alto grau de semelhança (mesmos nomes de variáveis, lógica e estrutura praticamente idêntica, variação apenas de comentários, etc.) receberão nota 0!

Observações sobre os projetos

- Cada item cadastrado deve ter no mínimo 3 campos de dados;
- A implementação deve abordar os seguintes conceitos:
 - Funções;
 - Structs;
 - Alocação dinâmica.
- No caso da remoção de um item da lista, basta apagar/resetar os dados registrados;
- O menu deve ser feito de forma simples, via terminal, deixando claro as opções que o usuário possui;
- O menu deve ter uma opção que permita sair do programa

Apresentação final

Cada grupo terá 10 minutos para realizar sua apresentação, que deve incluir a exibição do código desenvolvido e a execução do projeto em si. A ordem de apresentação será baseada no número dos grupos, que será definido **dia 16/11, em aula**, seguindo o seguinte cronograma:

Horário	Grupo
09:00	Grupo 1
09:20	Grupo 2
09:40	Grupo 3
10:00	Grupo 4
10:20	Grupo 5
10:40	Grupo 6
11:00	Grupo 7
11:20	Grupo 8

Todos os membros devem estar presentes e ter tempo de fala, sendo a participação na apresentação parte da nota individual. Não será dado tempo extra em caso de atraso ou ausência de algum membro.

Faltas na apresentação final sem justificativas oficiais (atestados/licenças médicas, etc.) não poderão ser abonadas.

1 - Registro de chamada

Crie um sistema que permita o cadastro de alunos para a realização de uma chamada. O menu deve permitir as seguintes ações:

1. Inserir um novo aluno;
2. Listar alunos;
3. Buscar aluno;
4. Editar aluno;
5. Remover aluno;
6. Realizar chamada, marcando alunos faltantes;
7. Ler uma data e salvar chamada em arquivo nomeado pela data.

2 - Registro de hóspedes

Crie um sistema que permita o cadastro de hóspedes de um hotel. O menu deve permitir as seguintes ações:

1. Inserir um novo hóspede em um quarto vazio;
2. Listar hóspedes;
3. Buscar hóspede;
4. Editar hóspede;
5. Remover hóspede;
6. Mostrar os números dos quartos vazios;
7. Salvar lista de hóspedes com respectivos quartos em arquivo.

3 - Registro de gastos

Crie um sistema que permita o cadastro de gastos mensais. O menu deve permitir as seguintes ações:

1. Inserir informações de um gasto novo, incluindo data em que ocorreu o gasto;
2. Listar gastos;
3. Buscar pelo nome de um gasto;
4. Editar informações de um gasto;
5. Remover gasto;
6. Mostrar soma e média dos gastos;
7. Salvar a lista de gastos em um arquivo.

4 - Registro de livros

Crie um sistema que permita o cadastro de livros em uma biblioteca. O menu deve permitir as seguintes ações:

1. Inserir novo livro, com quantidade de cópias disponíveis na biblioteca;
2. Listar livros cadastrados;
3. Buscar por um livro;
4. Editar informações de um livro;
5. Remover livro;
6. Pegar livro emprestado, atualizando número de cópias disponíveis;
7. Salvar a lista de livros, com número de cópias de cada um, em um arquivo.

5- Playlist (registro de músicas)

Crie um sistema que permita o cadastro de músicas em uma *playlist*. O menu deve permitir as seguintes ações:

1. Inserir nova música;
2. Mostrar músicas cadastradas na *playlist*, exibindo quais foram favoritadas;
3. Buscar por uma música;
4. Editar informações de uma música
5. Remover música;
6. “Favoritar” música já cadastrada na *playlist*;
7. Salvar *playlist* em um arquivo.

6 - Cinema (registro de sessões)

Crie um sistema que permita o cadastro de sessões de filmes em um cinema. O menu deve permitir as seguintes ações:

1. Inserir nova sessão de um filme específico, criando campo para representar quantidade de cadeiras livres;
2. Mostrar filmes disponíveis e respectivas sessões;
3. Buscar por um filme, mostrando horários das sessões;
4. Editar informações da sessão;
5. Remover sessão;
6. Reservar/comprar lugar em uma sessão específica;
7. Salvar informações das sessões, incluindo quantidade de cadeiras livres e ocupadas, em um arquivo.

7- Registro de tarefas

Crie um sistema que permita o cadastro de tarefas. O menu deve permitir as seguintes ações:

1. Inserir nova tarefa;
2. Mostrar tarefas cadastradas
3. Buscar por tarefa;
4. Editar informações de uma tarefa;
5. Remover tarefa;
6. Marcar conclusão de uma tarefa específica;
7. Salvar lista de tarefas em um arquivo, incluindo se cada uma foi finalizada ou não.

8 - Sistema de banco

Crie um sistema que registre contas bancárias e operações feitas entre elas, simulando a entrada e saída de dinheiro. O menu deve permitir as seguintes ações:

1. Criar nova conta;
2. Depositar e sacar dinheiro em uma conta;
3. Mostrar saldo total de cada conta;
4. Editar informações do titular de uma conta;
5. Remover conta;
6. Transferir valor de uma conta para outra;
7. Salvar informações de cada conta, incluindo saldo total, em um arquivo;

