

Universidade Católica de Pernambuco

Professor: Augusto César Oliveira

Disciplina: Programação Estruturada

Aluno(a): _____ data: ____/____/____

Observações:

* A prova deve ser feita em **dupla** ou **individual** e deve ser apresentada em sala de aula em até **7 minutos** na **data agendada no Teams**.

* Os códigos devem ser enviados pelo Teams até às 23:59 do dia que antecede a data da apresentação.

Esta prova consiste na construção de um **Sistema de Gerenciamento** em linguagem C. A escolha do **tipo do sistema é livre**, utilize a criatividade para escolher o domínio de aplicação. O projeto deve utilizar **Structs** para organizar os dados e **Arquivos** para persistência.

Questão 01. Definição da estrutura (struct) (peso 1,0):

- Defina o tipo de entidade que será gerenciada pelo sistema (ex: produtos, clientes, funcionários, etc.).
- Crie a estrutura (struct) que represente as informações relevantes para a entidade escolhida.

Questão 02. Implemente as seguintes ações através de um menu para o gerenciamento de registros do sistema utilizando arquivo de texto:

1. **Adicionar um novo registro** (peso 2,5):

- Solicite ao usuário as informações de um novo registro.
- Pergunte ao usuário se ele deseja adicionar este registro no **início** ou no **final** do arquivo.
- Execute a ação com base na escolha do usuário.

2. **Listar registros** (peso 2,0):

- Implemente uma função que mostra todas as informações dos registros contidos no arquivo de modo formato. Mas antes disso, o sistema deve solicitar ao usuário qual o tipo de visualização que ele deseja:
 - a. Os 5 primeiros registros;
 - b. Os 5 últimos registros;
 - c. Todos os registros.

3. **Buscar um registro** (peso 1,0):

- Crie uma função que permita buscar e exibir as informações de um registro específico com base em um parâmetro previamente definido. O parâmetro é definido no momento da implementação.

4. **Atualizar um registro** (peso 2,5):

- Implemente uma função que atualiza as informações de um registro.
- O registro deve ser buscado através de um **identificador**.
- Solicite as novas informações a serem atualizadas ao usuário.

5. **Remover um registro** (peso 1,0):

- Implemente uma função que remove um registro do arquivo. O registro deve ser buscado através de um **identificador**.