

Universidade Federal de Uberlândia - UFU
Bacharelado em Sistemas de Informação

FACOM32201 – Algoritmos e Programação II
Turmas 2025-1

Profa. Alessandra Aparecida Paulino e Prof. Thiago Pirola Ribeiro

Trabalho Prático

→ Instruções Gerais para Entrega ←

- (a) Antes de enviar o seu trabalho, considere com atenção e cuidado as instruções aqui contidas. Não segui-las pode levar o grupo a receber o conceito mínimo (zero).
- (b) O trabalho deverá ser realizado em quatro ou cinco alunos (não serão aceitos trabalhos individuais ou em grupos fora das especificações). Não deve haver troca de informações entre grupos distintos. Trabalhos iguais, conceito mínimo para ambos os grupos.
- (c) A entrega do trabalho será feita via Moodle por meio de link, que será disponibilizado conforme prazos determinados no final destas instruções.
- (d) Os envios devem ser realizados até a data e horário limite estabelecidos para consideração do trabalho. Entregas com atraso de 1 minuto até 24 horas serão considerados com peso de 75%. Não serão aceitas entregas com atraso superior a 24 horas.
- (e) O trabalho deve ser implementado na linguagem C, e será executado para fins de correção utilizando-se o CodeBlocks versão 25.03 no sistema operacional MS-Windows.
- (f) É obrigatória a utilização de indentação no código.
- (g) Todos os arquivos essenciais do trabalho devem estar contidos em uma pasta com o nome dos autores (sem espaços ou acentos), e esta pasta deve ser compactada e um único arquivo submetido (por exemplo, a entrega pode ser um arquivo nomeado MariaPereira_JoaoSilva.CatarinaGomes.CarlosSouza.zip para os alunos Maria Costa Pereira, João Braga Silva, Catarina Gomes e Carlos Alves Souza, que contém uma pasta nomeada MariaPereira_JoaoSilva.CatarinaGomes.CarlosSouza). **Não inclua os arquivos compilados**, mantenha apenas o arquivo do projeto no CodeBlocks (.cbp), os códigos fonte (.c), os arquivos pré-preenchidos (.txt ou .bin) e outros arquivos que possam ser necessários para a compilação (e.g., arquivos de cabeçalho .h).
- (h) A pasta deve conter um arquivo README.txt com todas as informações que julgar necessárias para a compilação e ou correção do seu trabalho, além do nome completo e matrícula de cada integrante do grupo.

→ Critérios de Avaliação ←

1. Compilação sem erros (será utilizado o CodeBlocks versão 25.03 no sistema operacional MS-Windows).
2. Execução sem erros (efetue muitos testes diferentes para verificar a presença de erros em tempo de execução).
3. Lógica utilizada na implementação.
4. Estruturas implementadas corretamente.
5. Todas as funções devem ser implementadas manualmente.
6. Completude do programa apresentado considerando os requisitos.

→ Descrição do Trabalho Prático ←

1. O trabalho consiste na implementação de um programa para gerenciar algum estabelecimento escolhido pelo grupo. Devem ser gerenciados pelo menos dois tipos de registros independentes, e deve ser gerenciado também um relacionamento relevante entre os registros.

Um exemplo disso (que não poderá ser utilizado por vocês) seria gerenciar os pacientes, os médicos e os atendimentos, em uma clínica médica. Deve ser criada uma estrutura (struct) para cada um deles:

- A estrutura 1 seria para os pacientes;
- A estrutura 2 para os médicos e;
- A estrutura 3 para os atendimentos, que relaciona pacientes e médicos.

Cada estrutura deve ter pelo menos 4 campos de dados, e um deles deve ser do tipo que não se repete para registros diferentes, por exemplo, para pacientes poderia ser o CPF ou um código único atribuído na clínica.

Utilize dados de tipos diversos (texto, inteiros, reais, etc).

Os campos da terceira estrutura devem relacionar as outras duas estruturas, por exemplo, no caso da clínica, médico e paciente se relacionam por meio de atendimentos, que precisam registrar quem é o médico que atendeu, quem foi o paciente atendido, e alguma informação adicional por exemplo data do atendimento, considerando que um paciente pode ser atendido pelo mesmo médico mais de uma vez.

2. O programa deve utilizar vetores de estruturas para cada uma das estruturas base, e estas devem ser salvas em arquivos binários (separadamente, um arquivo por vetor de estrutura).
3. É obrigatória a persistência dos dados. No início do programa, os dados das três estruturas devem ser carregados para seus respectivos vetores a partir dos arquivos para que o programa funcione corretamente com os dados inseridos anteriormente.

Os dados das três estruturas devem ser salvos nos arquivos binários para utilização na próxima execução do programa.

4. É obrigatório o uso de alocação dinâmica de memória para os vetores das Estruturas. O controle da memória alocada deve ser feito a todo momento, realocando memória se necessário quando da inclusão de novos registros.

Recomenda-se que seja alocada um pouco a mais de memória, 5 posições a mais (utilize uma constante para definir esse número de forma que seja fácil alterar quando necessário), de forma que não seja necessário realocar memória a cada inclusão de um registro. Não deve ser ultrapassada a quantidade de 5 posições livres. Caso ocorra alguma remoção de itens dos vetores, deve-se observar essa quantidade.

5. O programa deverá apresentar um menu que permita que o usuário:

- (a) Cadastre, altere, consulte ou remova (CRUD¹) registros do tipo da Estrutura 1.
- (b) Cadastre, altere, consulte ou remova (CRUD¹) registros do tipo da Estrutura 2.
- (c) Cadastre, altere, consulte ou remova (CRUD¹) registros do tipo da Estrutura 3.
- (d) Liste os dados de todos os registros da Estrutura 1 cadastrados.
- (e) Liste os dados de todos os registros da Estrutura 2 cadastrados.
- (f) Mostre todos os dados de todos os registros da Estrutura 3 a partir de característica específica, incluindo campos de pelo menos uma entre as Estruturas 1 e 2 relacionadas com cada registro. Obrigatório ter, pelo menos, 3 características específicas para esse item, ou seja, mostrar todos os registros a partir de uma característica específica de valor dado pelo usuário, com três opções possíveis de campos para a listagem.
No exemplo da clínica, poderiam ser mostrados a data do atendimento, o nome do médico e o CRM de todas as consultas feitas a partir de uma certa data de valor indicado pelo usuário.
- (g) Gere um relatório da Estrutura 3 a partir de informação dada pelo usuário. Obrigatório permitir que o usuário escolha entre, pelo menos, 3 informações. Ou seja, haverá opção de 3 relatórios distintos, cada um a partir de uma característica (campo) diferente.
Este relatório deve ser salvo em um arquivo texto, bem formatado para que o usuário possa compreender as informações ali incluídas, e que inclua informações das outras duas estruturas relacionadas com cada registro.
No exemplo da clínica, poderia ser um relatório contendo as informações de todos os registros com data a partir de uma data definida pelo usuário até o dia atual, incluindo para cada atendimento todas as informações do paciente e do médico.
- (h) Sair do programa.

6. **Todas as alterações e consultas devem ser feitas a partir de uma escolha do usuário para algum dos campos** (exemplos: o usuário digita o nome do paciente que ele quer alterar; o usuário digita o CRM do médico que ele quer buscar, etc).

7. As restrições a seguir devem ser atendidas no programa:

- O programa não pode permitir que um novo cadastro das estruturas seja efetuado se para o campo de valor único já houver um registro com aquele valor (por exemplo não podem ser feitos dois cadastros de pacientes com o mesmo CPF).
- Um registro da estrutura 3 só pode ocorrer a partir de registros existentes nas estruturas 1 e 2 (no exemplo da clínica, não posso cadastrar um atendimento se o médico ou o paciente não estiverem cadastrados).
- Um registro da Estrutura 1 ou da Estrutura 2 não pode ser removido se estiver presente na Estrutura 3 (por exemplo, um médico que deu atendimento, portanto está em pelo menos um registro do vetor de atendimentos, não pode ser excluído).

8. **As restrições mencionadas também não podem ser violadas em caso de alterações.**

9. Lembre-se de controlar o número de registros já feitos em cada estrutura, e que esse número não pode ultrapassar o tamanho do vetor sem que haja realocação de memória.

10. O programa principal deve conter apenas o suficiente para o funcionamento do menu e das estruturas. Todas as outras funcionalidades devem ser implementadas por meio de funções.

11. Devem ser obrigatoriamente incluídos na pasta arquivos pré-preenchidos com pelo menos 10 registros por estrutura criada, feitos a partir da execução do programa por vocês.

¹CRUD = Create (Criar), Read (Ler), Update (Atualizar) e Delete (Apagar).

12. **DICA:** crie uma função de busca para cada estrutura que retorna o índice do elemento encontrado, e chame esta busca sempre que necessário nas outras funções. Isso facilitará bastante o trabalho.

→ Entregas ←

Entrega Inicial:

- Um integrante do Grupo deverá acessar a planilha (https://ufubr-my.sharepoint.com/:x:/g/personal/tpribeiro_ufrbr/ESVqVn3ksnJHtA_oj0JFCA0B2ZIaETV8pppdTtZ_MrzuAQ?e=ghl93W) e preencher com:
 - * **Tema;**
 - * **Turma:** S ou Extra
 - * **Nomes dos integrantes:** Apenas quatro ou cinco alunos (não serão aceitos trabalhos individuais ou em grupos fora das especificações).
- **IMPORTANTE:** Não serão aceitos temas iguais, independente da turma. Por isso, defina o quanto antes os grupos e coloque na planilha.
- **Data da Entrega: 21/07/2025 até 23:59.**

1ª Entrega:

- O arquivo modelo (Definicao_1_Entrega.docx) deve ser utilizado para incluir as informações desta Entrega.
- **Data da Entrega: 22/07/2025 até 23:59.**
- *** Os grupos que não enviarem a 1ª entrega até a data limite, *não terão direito de fazer as próximas entregas e a avaliação do mesmo, obtendo conceito zero no trabalho* ***.
- O nome do arquivo deve seguir as instruções de nome apresentadas anteriormente (nome e sobrenome de todos os integrantes, separados por sublinha) no formato PDF (com extensão .pdf).

2ª Entrega:

- **Data da Entrega: 17/08/2025 até 23:59.**
- Para esta entrega deverão estar presentes no trabalho:
 - * As 3 Estruturas;
 - * Alocação Dinâmica;
 - * Ponteiros;
 - * Menus;
 - * Funções iniciais;
 - * CRUDs, pelo menos, das Estruturas 1 e 2 (Ver nota de rodapé 1).
- Devem ser entregues todos os arquivos essenciais do trabalho, que devem estar contidos em uma pasta com o nome dos autores (sem espaços ou acentos), e esta pasta deve ser compactada e um único arquivo submetido (por exemplo, a entrega pode ser um arquivo nomeado MariaPereira_JoaoSilva_CatarinaGomes_CarlosSouza.zip para os alunos Maria Costa Pereira, João Braga Silva, Catarina Gomes e Carlos Alves Souza, que contém uma pasta nomeada MariaPereira_JoaoSilva_CatarinaGomes_CarlosSouza). **Não inclua os arquivos compilados**, mantenha apenas o arquivo do projeto no CodeBlocks (.cbp), os códigos fonte (.c), os arquivos pré-preenchidos (.txt) e outros arquivos que possam ser necessários para a compilação (e.g., arquivos de cabeçalho .h).
- *** Os grupos que não enviarem a 2ª entrega até a data limite, *não terão direito de fazer entrega final e a avaliação do mesmo, obtendo conceito zero no trabalho* ***.

Entrega Final:

- **Data da Entrega: 11/09/2025 até 23:59.**
- Para esta entrega deverão estar presentes no trabalho:
 - * Todos os itens da 2^a Entrega;
 - * Todos os itens especificados na "Descrição do Trabalho Prático".
- Devem ser entregues todos os arquivos essenciais do trabalho, que devem estar contidos em uma pasta com o nome dos autores (sem espaços ou acentos), e esta pasta deve ser compactada e um único arquivo submetido (por exemplo, a entrega pode ser um arquivo nomeado MariaPereira_JoaoSilva_CatarinaGomes_CarlosSouza.zip para os alunos Maria Costa Pereira, João Braga Silva, Catarina Gomes e Carlos Alves Souza, que contém uma pasta nomeada MariaPereira_JoaoSilva_CatarinaGomes_CarlosSouza). **Não inclua os arquivos compilados**, mantenha apenas o arquivo do projeto no CodeBlocks (.cbp), os códigos fonte (.c), os arquivos pré-preenchidos (.txt) e outros arquivos que possam ser necessários para a compilação (e.g., arquivos de cabeçalho .h).
- A pasta deve conter um arquivo README.txt com todas as informações que julgar necessárias para a compilação e ou correção do seu trabalho, além do nome completo e matrícula de cada integrante do grupo.
- *** Os grupos que não enviarem a entrega final até a data limite, *não terão direito de fazer a avaliação do trabalho, obtendo conceito **zero** no trabalho* ***.

→ **Nota** ←

$$NotaTrabalho = AvTrab * AvInd$$

Avaliação do Trabalho Entregue (AvTrab) = máximo 25 pontos;

Avaliação Individual escrita sobre o Trabalho (AvInd): máximo 1 ponto;

A avaliação individual do Trabalho (**AvInd**) será no dia **16/09/2025** em sala de aula, de forma escrita e **individual**.