

Universidade do Estado de Santa Catarina (UDESC)
Curso: Bacharelado em Ciência da Computação
Disciplina: Linguagem de Programação (LPG0001)
Prof. Rui Jorge Tramontin Jr.

Trabalho 2: Agenda de Eventos

O objetivo do trabalho é desenvolver um programa que gerencie um cadastro simples de uma **agenda de eventos**. O programa deve alocar (e realocar) um **vetor de estruturas** cujo modelo é apresentado a seguir:

- **Tipo Data** (*estrutura contendo dia, mes, ano*);
- **Tipo Horário** (*estrutura contendo hora, minuto*);
- **Tipo Evento** (*estrutura contendo os campos a seguir*):
 - data (*estrutura Data*);
 - horário início (*estrutura Horario*);
 - horário fim (*estrutura Horario*);
 - descrição (*string de 50 caracteres*);
 - local (*string de 50 caracteres*);

O programa deve **carregar os dados de um arquivo de texto** e alocá-los em um **vetor de estruturas**. Caso o arquivo não exista, o ponteiro do vetor deve apontar para **NULL**. A seguir, o programa deve oferecer um **menu de texto** com as seguintes opções:

1. **Cadastrar** (somente um) novo evento na agenda;
 - Deve fazer a realocação do vetor;
 - Inserção deve ser feita de modo que os eventos estejam sempre em **ordem crescente** (por data e horário de início) → *Sugestão: insira no final e depois ordene o vetor*;

Restrição: o novo evento não pode sobrepor outro evento, ou seja, se o novo evento for no mesmo dia, ele não pode ocupar o intervalo (horários de início e de fim) de outro evento já cadastrado!

2. **Mostrar todos os eventos da agenda**;
3. Dada uma data, **mostrar todos os eventos dessa data**;
4. Dada uma descrição, **mostrar todos os eventos que tenham essa descrição**;
 - Para simplificar, a busca pode ser pela *string* exata, ou seja, a comparação pode ser feita com `strcmp()` ou `strcmpi()`. Por outro lado, quem quiser se aventurar pode tentar usar a função `strstr()` para encontrar um *substring* dentro de outra *string*. Assim, será possível fazer uma busca menos restritiva.

5. **Remover evento:** dadas uma data e hora inicial, remover o respectivo evento;

- Observações:

- a. Aqui, o programa deve identificar qual índice do vetor deve ser removido;
- b. Em seguida, se houver dados "à direita", os mesmos devem ser movidos de modo a sobrescrever o índice a ser removido;
- c. Por fim, a quantidade de eventos é decrementada e o vetor é realocado para menos.

<u>[0]</u>	<u>[1]</u>	<u>[2]</u>	<u>[3]</u>	<u>[4]</u>
12/04/2025	14/04/2025	14/04/2025	16/04/2025	20/04/2025
21:00	14:00	20:30	19:00	11:00
Jantar	Reunião	Jantar	Prova	Pagar boleto
Casa da mãe	UDESC	Casa da tia	UDESC	Banco

Remove evento de 14/04/2025 às 14:00

<u>[0]</u>	<u>[1]</u>	<u>[2]</u>	<u>[3]</u>
12/04/2025	14/04/2025	16/04/2025	20/04/2025
21:00	20:30	19:00	11:00
Jantar	Jantar	Prova	Pagar boleto
Casa da mãe	Casa da tia	UDESC	Banco

6. **Sair** do programa: salva vetor de eventos no arquivo e libera a memória.

Requisitos do programa

- Validação da entrada (data e hora com valores válidos);
- Cuidado com a interface de usuário: coloque mensagens apropriadas para situações excepcionais, tais como erro na validação da entrada ou que o evento já existe (opção 1), lista de eventos vazia (opções 2, 3, 4), ou que o evento não foi encontrado (opção 5);
- Uso apropriado da realocação, para utilizar a memória de maneira otimizada;
- *Desalocação* do vetor ao final da execução (antes de salvar no arquivo).

Critérios de avaliação

- Execução correta e alinhamento com o que foi solicitado neste enunciado.