

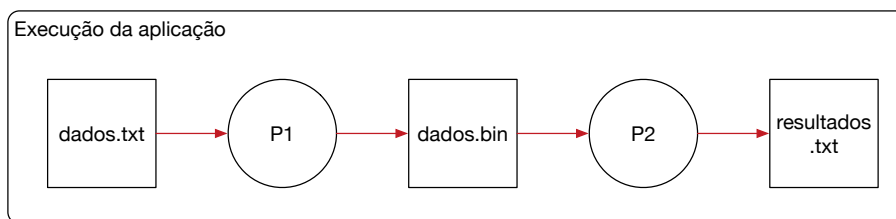
Programação Imperativa - PjBL

- Atividade individual: implementação de uma aplicação na linguagem C
- Prazo para conclusão e postagem no AVA: véspera da última aula do semestre
- Deve ser entregue, além dos arquivos contendo código fonte e dados, **links** para dois vídeos de, no máximo, cinco (5) minutos cada um, a saber:
 - um vídeo que mostre o uso da aplicação
 - um vídeo que explique o código fonte

Estrutura geral da aplicação:

A aplicação deve ser composta por dois (2) programas na linguagem C, a saber:

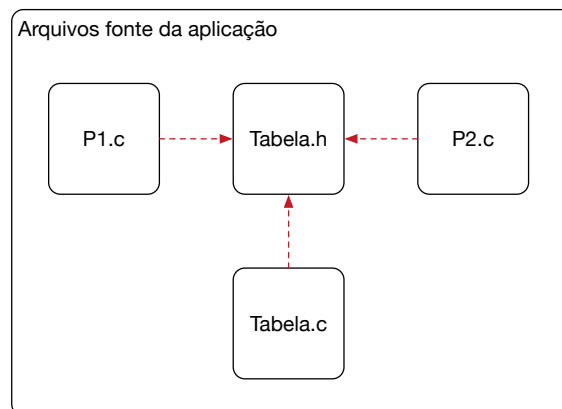
- O primeiro programa deve ler dados e um arquivo texto, armazenar internamente esses dados em forma de tabela e, então, gravar esses dados em um arquivo binário.
- O segundo programa deve ler os dados do arquivo binário gerado pelo primeiro programa, armazenar internamente esses dados em forma de tabela, realizar algum processamento sobre esses dados e, então, gravar os resultados do processamento em um novo arquivo texto.



Requisitos de implementação:

1. Sobre a tabela que armazena os dados tanto no primeiro como no segundo programa:
 - a. Deve ser implementada como um array (vetor) cujos elementos são instâncias de um struct.
 - b. O struct que define o tipo de cada elemento da tabela deve conter, no mínimo, cinco (5) campos, sendo que:
 - i. ao menos um campo é definido como uma string
 - ii. ao menos um campo é definido como um enum
 - iii. ao menos um campo é definido como um double
 - c. Deve conter, ao menos, 20 elementos, ou seja, 20 instâncias do struct.
2. Sobre o processamento realizado pelo segundo programa:
 - a. Deve permitir a realização de, ao menos, três operações distintas, a critério do usuário.
 - b. Ao menos, uma das operações deve implicar na realização de cálculos sobre os dados da tabela.

- c. Deve haver uma interação do programa com o usuário para permitir a escolha de uma operação, fazendo-se as devidas verificações para consistência dos dados fornecidos pelo usuário. Essa interação deve ser contínua, até o usuário escolha a opção de encerramento do programa.
 - d. Deve utilizar algum comando de repetição (while, do-while, for).
 - e. Deve utilizar algum comando de seleção (if, switch).
- 3. Sobre os arquivos texto e binário:
 - a. Os seus nomes devem ser fornecidos como parâmetros de execução dos programas (argv).
 - b. Devem ser feitas todas as verificações necessárias sobre o sucesso da operação de abertura de arquivo.
- 4. Sobre a organização do código:
 - a. Cada programa deve ser estruturado em, no mínimo, quatro (4) funções, incluindo a função main. Ou seja, a aplicação completa deve ter, no mínimo, seis (6) funções, além das duas funções main dos dois programas.
 - b. Deve ser organizado em, no mínimo, três arquivos de implementação (arquivo .c) e um arquivo cabeçalho (arquivo .h) que seja incluído por três arquivos de implementação. Sugestão: isolar todo o código de manipulação da tabela de dados em um arquivo específico.



- 5. Sobre a codificação:
 - a. Deve ser feito uso de alocação dinâmica de memória.
 - b. Deve haver, no mínimo, um parâmetro de função passado por referência (isto é, com uso de ponteiro).
 - c. Deve haver, no mínimo, um retorno de função que seja um ponteiro.
- 6. Sobre a apresentação do código fonte:
 - a. Deve respeitar todas as boas práticas de codificação estudadas no TDE 1.