



Universidade Federal de Ouro Preto - UFOP  
Instituto de Ciências Exatas e Biológicas - ICEB  
Departamento de Computação - DECOM  
Disciplina: Estrutura de Dados I – BCC 202  
Aluno: Felipe Fontenele de Ávila Magalhães  
Matrícula: 15.1.4331



## Documentação Questão 1

### Implementação

O código em questão, foi separado em 4 arquivos **.h** (TRecurso.h, TTarefa.h, TProcesso.h e TComputador.h), 4 arquivos **.cpp** (TRecurso.cpp, TTarefa.cpp, TProcesso.cpp e TComputador.cpp) e o **Main.cpp**. Será explicado o que foi feito nos arquivos.cpp e no main.cpp, já que os arquivos.h contêm somente as definições e structs. No **TRecurso.cpp** foram feitas as seguintes funções: inicializar o recurso em questão, as funções get e set de cada campo da struct TRecurso, função para verificar se está disponível tal recurso pela quantidade de demanda, função para reservar recurso pela quantidade de demanda, função para liberar recurso pela quantidade de demanda, função para calcular tempo, seguindo a fórmula ( $\text{velocidade} \times \text{demanda}$ ) e outra função para liberar a memória do recurso, já que todas as implementações foram com alocação dinâmica de memória. No **TTarefa.cpp** foram feitas as seguintes funções: inicializar a tarefa em questão, as funções get e set de cada campo da struct TTarefa e uma função para executar a tarefa. No **TProcesso.cpp** foram feitas as seguintes funções: inicializar o processo em questão, função para somar a demanda do vetor que foi feito para receber os tipos de demanda, função para retornar esse vetor feito, as funções get e set de cada campo da struct TProcesso, função para adicionar tarefa onde o vetor de tarefas do processo recebe as informações de tarefa, como tipo e demanda, nesta função também é incrementado o contador de processos e soma-se a demanda por meio da função dita anteriormente que faz tal operação, há ainda uma função para executar o processo, que dentro de um **for** é chamado a função de executar tarefa dentro de um **for** que parte de 0 até o contador de processos, e por último, uma função de liberar memória do processo. No **TComputador.cpp** foram feitas as seguintes funções: inicializar o computador em questão, uma função para verificar a disponibilidade do computador, isto é, verificar se ele suporta determinada quantidade de demanda, uma função para adicionar processo pois há um vetor de processo no computador, uma função para adicionar recurso pois também há um vetor de recurso no computador, outra função para executar o computador analisando se o computador pode executar tal processo, calculando o tempo necessário de execução e uma função para liberar a memória do computador. No **Main.cpp** pede-se ao usuário para digitar a quantidade de computadores, a quantidade de recursos e processos de cada computador, tipo de recurso, demanda e velocidade, onde são feitas as devidas chamadas de funções para fazer, por exemplo, calcular o tempo de execução do processo dentre outros; foi feito também um vetor de computadores e



Universidade Federal de Ouro Preto - UFOP  
Instituto de Ciências Exatas e Biológicas - ICEB  
Departamento de Computação - DECOM  
Disciplina: Estrutura de Dados I – BCC 202  
Aluno: Felipe Fontenele de Ávila Magalhães  
Matrícula: 15.1.4331



um vetor de processos; nesta parte do código é feito a análise dos computadores, ou seja, o escalonador do sistema analisando qual PC consegue executar determinado processo. Ao final do programa, imprime a quantidade de computadores com seus respectivos processos executados com o tempo total de execução e também a quantidade de processos com seus respectivos computadores na qual foram executados com o tempo de execução de cada processo.

## **Impressões gerais**

Em minha opinião, essa questão foi de um nível muito difícil, já que havia várias TADs cujo trabalho interligava entre si, foi necessário acrescentar diversas coisas para obter o devido resultado, como por exemplo, criação de novas funções e campos dentro das structs para ter um controle do que estava sendo feito. Precisei ir ao monitor para que pudesse ajudar, pois estava confundindo diversos conceitos.

## **Conclusões**

Esse código como foi dito anteriormente, foi de um nível difícil onde fiz com muita atenção para não desobedecer a nenhuma norma imposta, contudo foi necessária a ajuda do monitor para que o código ficasse pronto. Foram feitas funções que na questão não foi pedido, mas foi necessário para a execução do código.