

# Especificação dos Laboratórios da disciplina TT005 – Tópicos Especiais em Telecomunicações I Programação de Alto Desempenho

Prof. André Leon S. Gradvohl, Dr.

## Introdução

Os Laboratórios ( $L_1, L_2, L_3$ ) são um dos componentes de avaliação da disciplina TT005 – Tópicos Especiais em Telecomunicações I (Programação de Alto Desempenho) e servem para avaliar o conhecimento dos alunos sobre o tema. Os laboratórios devem ser realizados em dupla ou individualmente. A mesma dupla ou indivíduo que fará o seminário deverá realizar os laboratórios.

Os três laboratórios vão resolver o mesmo problema utilizando bibliotecas para programação de alto desempenho (PAD) diferentes, **obrigatoriamente na Linguagem C e no ambiente Linux**. A definição do problema a ser resolvido, bem como as demais informações sobre os laboratórios estão nas seções a seguir.

## Definição do problema a ser resolvido

O problema a ser resolvido em todos os laboratórios é o seguinte.

Dadas três matrizes,  $A_{y \times w}$ ,  $B_{w \times v}$ ,  $C_{v \times 1}$ , calcule a matriz  $D_{y \times 1}$  tal que  $D_{y \times 1} = (A_{y \times w} \times B_{w \times v}) \times C_{v \times 1}$ . Além disso, calcule a redução pela soma dos elementos na matriz  $D_{y \times 1}$ , isto é, a soma de todos os elementos em  $D_{y \times 1}$ .

As dimensões das matrizes são estabelecidas pelas variáveis  $y, w, v$ , cujos valores deverão ser informados pela linha de comando do programa. Além disso, os dados que compõem as matrizes  $A_{y \times w}$ ,  $B_{w \times v}$ ,  $C_{v \times 1}$  devem ser números reais, com até duas casas decimais, que variam entre -10 e +10, gerados aleatoriamente.

Esses dados devem estar dispostos sequencialmente em cada arquivo texto, isto é, o arquivo deve ter um valor real em cada linha, sem espaços antes ou depois de cada número. Os nomes dos arquivos que contêm os dados também deverão ser informados pela linha de comando.

A matriz resultante da computação, isto é, a matriz  $D_{y \times 1}$ , deve estar em um arquivo texto, cujos dados também devem estar dispostos sequencialmente, com duas casas decimais. O nome desse arquivo também deve ser especificado na linha de comando. Já a redução pela soma dos elementos na matriz  $D_{y \times 1}$  deve ser mostrado na tela. Esse valor deve ser exibido sem nenhuma *string* nem antes, nem depois do número.

## Exemplo de linha de comando para instanciar o programa

A linha de comando para o programa de cada laboratório deve ter a seguinte sintaxe:

```
./programa y w v arqA.dat arqB.dat arqC.dat arqD.dat
```

Onde:

- `./programa` é o programa que resolverá o problema.
- `y` é o número de linhas da primeira matriz.
- `w` é o número de colunas da primeira matriz e de linhas da segunda matriz.
- `v` é o número de colunas da segunda matriz e de linhas da terceira matriz.
- `arqA.dat` é o nome do arquivo que contém a primeira matriz.
- `arqB.dat` é o nome do arquivo que contém a segunda matriz.
- `arqC.dat` é o nome do arquivo que contém a terceira matriz.

- `arqD.dat` é o nome do arquivo que contém a matriz resultante da computação.

Observe um exemplo a seguir.

```
./prog 10 20 15 arqA.dat arqB.dat arqC.dat arqD.dat
```

### Requisitos importantes dos programas implementados em cada laboratório

Há alguns requisitos fundamentais que devem ser respeitados nos programas de cada laboratório. O não atendimento de pelo menos um desses requisitos implicará em nota zero em cada laboratório.

O primeiro requisito é que todas as matrizes precisam ser alocadas dinamicamente (com o comando `malloc` ou equivalente). Havendo qualquer tipo de alocação estática de tipos compostos (matrizes ou vetores) no programa, isso implicará em nota zero no respectivo laboratório.

O segundo requisito é que todas as matrizes necessárias para os programas precisam ser alocadas em uma única etapa, isto é, em uma única chamada do comando `malloc` ou equivalente. Veja uma discussão sobre a alocação em única etapa no site <https://gradvohl.github.io/alocaMatrizes>. Havendo qualquer tipo de alocação em duas etapas no programa, isso implicará em nota zero no respectivo laboratório.

O terceiro requisito é que o programa deve compilar sem erros. Para tanto, as instruções para a compilação devem estar muito bem definidas. Recomenda-se que seja criado um `makefile` ou um *script* para a compilação. Se o programa não compilar, ou aparecerem erros de compilação, isso implicará em nota zero no respectivo laboratório.

O quarto requisito é que os dados também devem estar dispostos sequencialmente em arquivos texto, conforme estabelecido no início da seção. Se o arquivo de saída do programa não respeitar esse requisito, isso implicará em nota zero no respectivo laboratório.

Por fim, o quinto requisito é utilizar exatamente a linha de comando indicada no exemplo informado na subseção anterior. Se o programa não funcionar com a linha de comando indicada, isso implicará em nota zero no respectivo laboratório.

### Especificação de cada laboratório

Conforme já mencionado, os laboratórios ( $L_1, L_2, L_3$ ) resolverão exatamente o mesmo problema, com os mesmos requisitos. No entanto, cada laboratório deverá utilizar uma biblioteca diferente para resolver o problema, de acordo com a lista a seguir.

- O Laboratório  $L_1$  deve resolver o problema utilizando a biblioteca OpenMP, com 4 núcleos de processamento.
- O Laboratório  $L_2$  deve resolver o problema utilizando a biblioteca OpenACC, com uma GPU.
- O Laboratório  $L_3$  deve resolver o problema utilizando a biblioteca MPI, com dois computadores e 4 núcleos de processamento para cada computador.

Para cada laboratório, os alunos deverão produzir um relatório contendo a forma de compilação e execução de cada programa, as análises e os gráfico com os tempos de processamento (descontadas as operações de E/S) para os seguintes valores das variáveis  $y, w, v$ :

- 🕒  $y = 10, w = 10, v = 10$
- 🕒  $y = 100, w = 100, v = 100$
- 🕒  $y = 1000, w = 1000, v = 1000$

A análise deverá explicar as eventuais variações nos tempos de processamento (descontadas as operações de E/S).

A medição do tempo de processamento deve considerar apenas o tempo para resolver a equação  $D_{y \times 1} = (A_{y \times w} \times B_{w \times v}) \times C_{v \times 1}$  e o cálculo da redução por soma. Ou seja, não devem ser considerados os tempos para leitura e gravação dos arquivos.

## Os produtos dos laboratórios e a entrega

Em cada laboratório, as equipes produzirão códigos fontes para a linguagem C e relatórios. Toda essa produção deve estar em um repositório no GitHub.

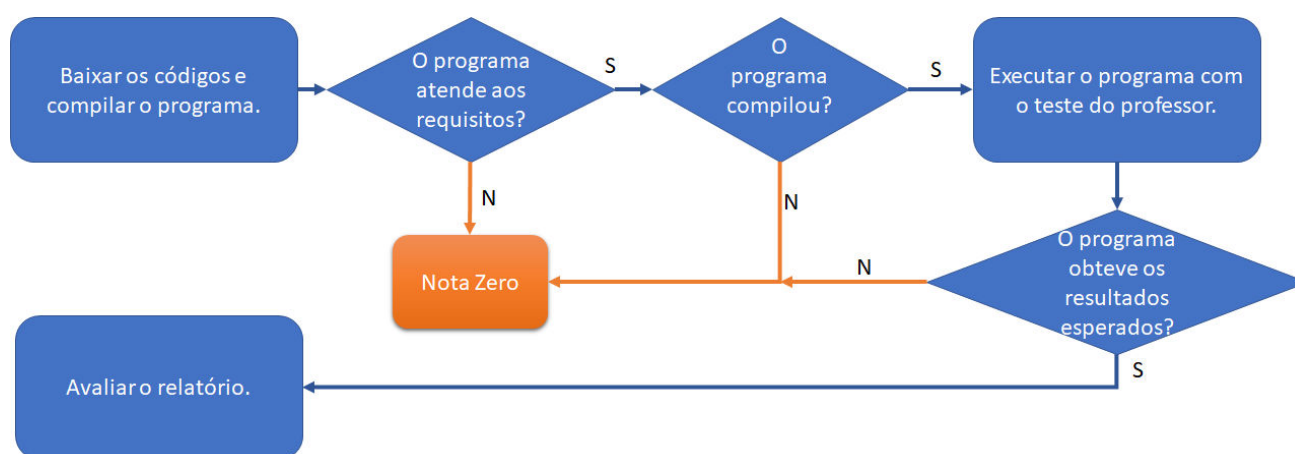
No repositório, para cada laboratório deve haver uma pasta (diretório). Cada pasta deve ter os códigos fontes – devidamente documentados – e o relatório referente àquele laboratório. O relatório deve estar obrigatoriamente em um arquivo no formato *Portable Document Format* (PDF). Alternativamente, o relatório pode estar em um arquivo `README.md` (escrito na linguagem de programação *Markdown*), dentro da respectiva pasta.

A falta do relatório – seja em PDF ou no `README.md` – ou dos códigos implicará em nota zero no respectivo laboratório.

Nas datas indicadas no plano de ensino da disciplina, até as 23h55, cada relatório deverá ser entregue no Ambiente Virtual de Aprendizagem da Disciplina (Moodle). Esse ato – a entrega do relatório – indicará que os laboratórios foram concluídos e estão prontos para a avaliação. Qualquer modificação posterior à data e horário, implicará em nota zero para esse laboratório.

## Avaliação dos laboratórios

A avaliação dos laboratórios seguirá o seguinte fluxograma:



Nos produtos dos laboratórios, serão avaliados os seguintes critérios.

- A qualidade dos códigos produzidos, inclusive o grau de modularidade dos códigos, e a inovação no que se refere ao uso dos núcleos de processamento disponíveis.
- A qualidade da documentação do código.
- A qualidade da documentação do relatório, incluindo a correta utilização da norma gramatical da língua portuguesa.

## Recursos computacionais para a implementação dos laboratórios

Na ocasião da especificação dos laboratórios, não havia ainda uma definição da infraestrutura para a implementação dos resultados. Provavelmente, essa infraestrutura utilizará os serviços da Amazon Web Services.