

# Laboratório MIPS

## Projeto

*Baseado em enunciado do Prof. Montek Singh (UNC Chapel Hill)*

O grupo deverá propor um **mini projeto** e concluí-lo em cerca de três semanas. A complexidade do projeto deve ser equivalente aproximadamente a três das tarefas regulares de laboratório. O projeto deve ser implementado usando o simulador MARS.

### Exemplos de mini projetos:

- Implemente um programa que calcule estatísticas de um conjunto de números, por exemplo, média, variância, etc. Você deve incluir cálculos de ponto flutuante ou algum outro recurso para tornar isso não trivial.
- Implemente o Jogo da Vida de Conway, redesenhando a tela repetidamente usando caracteres ASCII ([https://pt.wikipedia.org/wiki/Jogo\\_da\\_vida](https://pt.wikipedia.org/wiki/Jogo_da_vida)).
- Implemente o algoritmo de Dijkstra para calcular o caminho mais curto entre as cidades.
- Implemente um reprodutor de música MIDI. O MARS inclui suporte para geração de som usando chamadas de sistema.
- Projete uma calculadora NPR (notação polonesa reversa). Ela deve manipular pelo menos as seguintes operações: adicionar, subtrair, multiplicar e dividir sobre números inteiros. Por exemplo, a expressão  $16\ 10 - 5\ 3 - /$  deve fornecer 3.
- Implemente um programa que execute alguns processamentos interessantes de números de ponto flutuante (por exemplo, calcular sen/cos usando a série de Taylor, problemas de aproximação sucessiva, etc.).
- Implemente um programa que resolva o problema das Torres de Hanói usando recursão. Certifique-se de que o número de discos possa variar durante a demonstração e que a saída seja fácil de entender.
- Implemente um programa com recursos de ordenação e busca. Por exemplo, você pode implementar o *quicksort* e/ou o *mergesort* recursivamente.
- Implemente um jogo da velha para 2 jogadores no qual seu programa gerencie e exiba o tabuleiro e anuncie o resultado; seu programa não precisa jogar contra você.
- Embarcados/IOT: Implemente um simulador de sistema embarcado para algum dispositivo de IOT (eletrodomésticos, automotivo, equipamentos de academia, planta industrial, cidades inteligentes, etc). O simulador deve apresentar uma interface intuitiva para a entrada e o mostrador do equipamento, sensores podem ser simulados para utilização como entrada de dados).

- Desenvolva um programa de manipulação de imagens. Por exemplo, dada uma imagem PPM ou PGM (você pode especificar o tamanho máximo que seu programa irá manipular), seu programa pode ampliá-la ou reduzi-la (por fatores simples de 2, 3 ou 4, por exemplo) ou implementar um filtro de desfoque simples, etc
- Implemente um jogo interessante, com ou sem interface gráfica no MARS.

Acima são apenas alguns exemplos. O grupo é encorajado a propor qualquer outro problema interessante que gostaria de resolver. Mas, certifiquem-se de que pode ser concluído dentro do prazo! O trabalho envolvido deve ser aproximadamente igual ao de três tarefas de laboratório. Se você está pensando em propor um projeto que provavelmente exigiria mais de 50 linhas de código C e/ou mais de 100 linhas de montagem MIPS, recomendo fortemente que proponha algo mais simples ;).

Como um guia geral, para tornar seu projeto de complexidade/desafio adequado, tente incorporar pelo menos um (ou mais) dos seguintes:

- Uso de funções e recursão.
- Estruturas de dados (por exemplo, pilhas, filas, estruturas, arrays multidimensionais, etc.)
- I/O sofisticado (por exemplo, I/O de arquivo ou saída de texto bem formatado, arte ASCII, etc.)
- Aritmética não trivial (por exemplo, ponto flutuante, operações com matrizes, etc.)
- Tomada de decisão complexa (por exemplo, condicionais profundamente aninhados típicos de IA)

Seu projeto será avaliado em:

- (i) adequação da dificuldade/desafio;
- (ii) qualidade da implementação;
- (iii) originalidade (os programas estão sujeitos a verificação de similaridade e desconto total da nota em caso de plágio).

## Cronograma

As atividades abaixo serão publicadas no Classroom:

1. **Proposta:** Enviar uma proposta breve (somente 1-2 parágrafos) sobre o que o grupo pretende implementar. Incluir detalhes suficientes para que seja possível avaliar o nível de dificuldade e fornecer feedback. Incluir uma estimativa aproximada de quantas linhas código assembly MIPS serão necessárias.
2. **Demonstração do MIPS:** O grupo demonstrará sua implementação do MIPS durante os horários de lab neste período.
3. **Entrega:**
  - a. Vídeo de até 5 minutos explicando o código e demonstrando o programa;
  - b. Link do código no GitHub.