

## Atividades de Programação - 3

### Instruções gerais:

- As atividades poderão ser entregues em grupo de até 4 integrantes.
- Entrega:
  - link no GitHub (ou similar) com o código desenvolvido
  - link para os vídeos descrevendo o código e seu funcionamento (máx. 5 min)
    - 1 vídeo por atividade
    - usar drive institucional ou Youtube
    - mostrar: principais trechos do código, exemplos de funcionamento (programa em execução), resultados obtidos (comentários, gráficos quando se aplicar)
    - sugestão: usar slides para informações estáticas, sem esquecer de demonstrar o programa em execução

OBS: os links podem ser entregues via arquivo no Google Docs

## Jogo da Vida em MPI

A partir da descrição da atividade 1/2 (Jogo da Vida *multithreaded*), construir um código-fonte em linguagem C com MPI para executar o Jogo da Vida, de forma concorrente/paralela, com múltiplos processos.

O desempenho deve ser avaliado para execuções com 1, 2, 4 e 8 processos paralelos em único ou múltiplos nós de processamento. Deve-se destacar a especificação da máquina, e o tempo de processamento, *speedup* e eficiência.

Compare os resultados obtidos com a melhor versão OpenMP já desenvolvida em atividade anterior, nas mesmas configurações (não esquecer de ativar as opções de otimização de compilação).

### Dicas:

a) Faça uma divisão de domínio para o tabuleiro do jogo da vida de forma **unidimensional**, de forma que cada processo seja responsável pelo cálculo de um pedaço da tabela que corresponde a um conjunto de linhas.

b) os processos vizinhos deverão trocar dados entre si, correspondente às linhas de fronteira entre as sub-tabelas adjacentes. Deve-se tomar cuidado para que não ocorram *deadlocks* nesta comunicação.

c) A inicialização da tabela deve ser feita em paralelo de forma a reproduzir o mesmo padrão da versão serial.

**Entrega:** vídeo e link no GitHub ou similar (ver Instruções Gerais).

**OBS:** as submissões de códigos estão sujeitas a análise de plágio através do software MOSS (<http://theory.stanford.edu/~aiken/moss/>). A checagem de similaridade baseia-se não apenas em uma simples diferenças entre arquivos texto, mas em critérios mais avançados que podem melhor entendidos aqui:  
<http://theory.stanford.edu/~aiken/publications/papers/sigmod03.pdf>