



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
Instituto de Ciências Exatas e de Informática

Saulo de Moura Zandona Freitas<sup>1</sup>

## Lista #9

Computação Distribuída

---

<sup>1</sup>Aluno de Graduação em Ciência da Computação– saulomzf@gmail.com

**Crie e explore uma sequência de processos simulados, incluindo DMUTEX; explique e mostre o que fez, através de prints.**

O código utilizado é o mesmo disponibilizado na descrição da tarefa 9. Esse código está disponível no [Github](#).



Figura 1: Imagem da Interface gráfica

Nessa figura podemos ver a aplicação gráfica. Na parte esquerda da aplicação, fica os controladores que ajustam quantos processos e quantas mensagens estarão presentes na simulação, assim como a duração da simulação e a velocidade que será executado. também estão presentes os controladores da simulação com a simulação livre ou a simulação com passo a passo, e assim por diante.

obs.: Lembrando que cada execução desse simulador é aleatória, ou seja, a cada simulação a quantidade de processos e timestamp dos processos poderá ser diferente(e muito provavelmente será).



Figura 2: Imagem da simulação iniciada

Nessa imagem é perceptível que o processo 2, no clock 0, enviou uma mensagem para o processo 4 que recebeu a mensagem no clock 1, assim cada tempo seguinte do processo 4 foi aumentado em 1

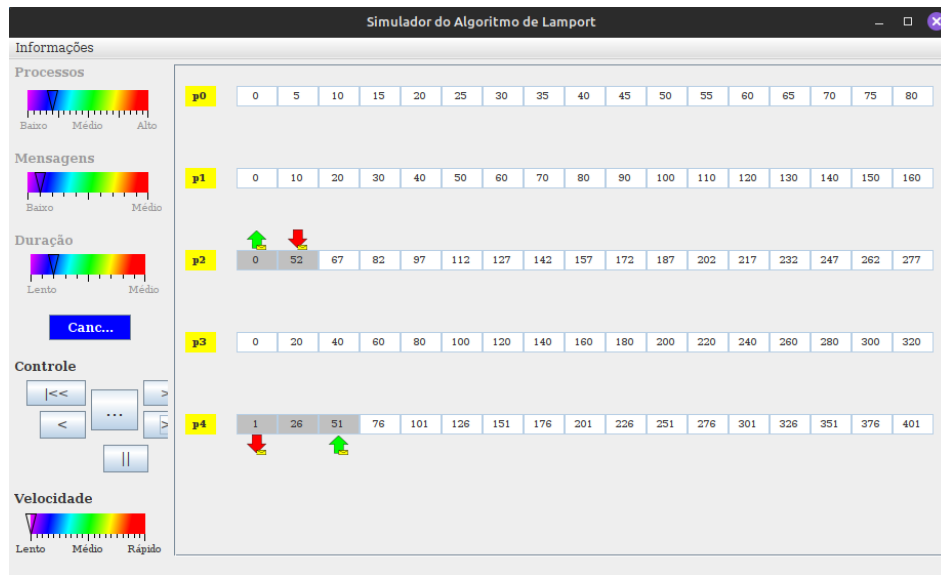


Figura 3: Imagem da segunda iteração da simulação

Na segunda iteração o processo 4, no clock 51, envia uma mensagem para o processo 2 que recebe no clock 52, mas o relógio lógico do processo 2 como um todo é atualizado para sincronizar o relógio lógico do processo 2 com os outros processos para o recebimento de novas mensagens.

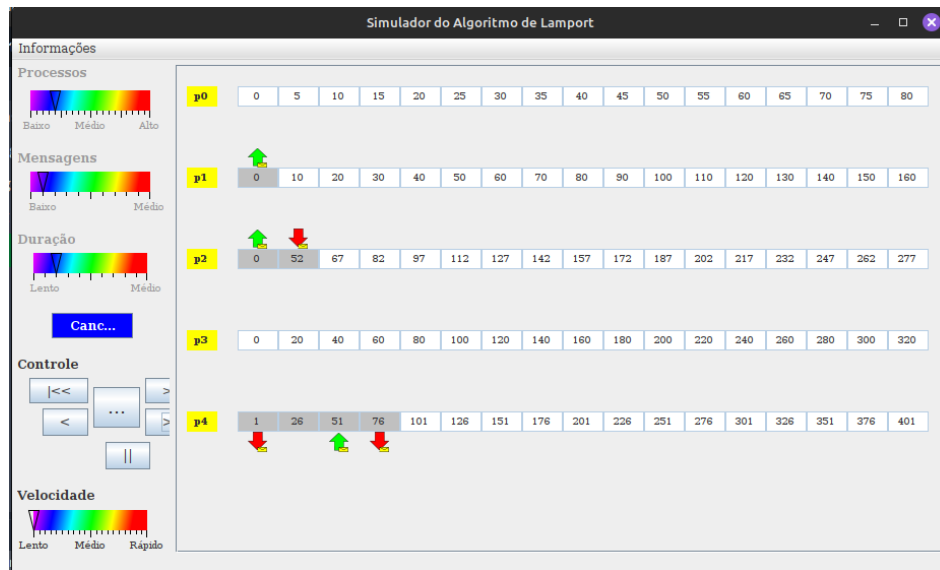


Figura 4: Imagem da terceira iteração da simulação

Na terceira iteração, podemos ver que a mensagem é enviada pelo processo 1 no clock 0 e é recebida no clock 76 pelo processo 4, assim, como a mensagem foi enviada em um clock menor que o clock do processo recebido, então não é necessário uma sincronização e o clock do processo 4 continuou o mesmo.

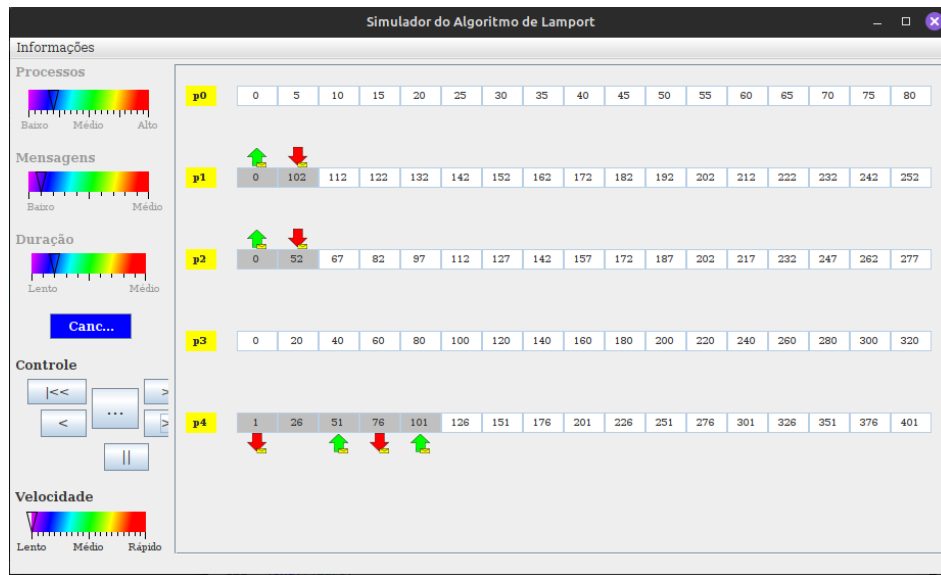


Figura 5: Imagem da quarta iteração da simulação

Na quarta iteração acontece a mesma situação que na segunda iteração, em que o processo 4 envia uma mensagem no clock 101 e o processo 1 recebe a mensagem e precisa sincronizar seu clock para o recebimento de novas mensagens, assim fazendo todos os clocks do processo 1, após o recebimento da mensagem, serem atualizados.

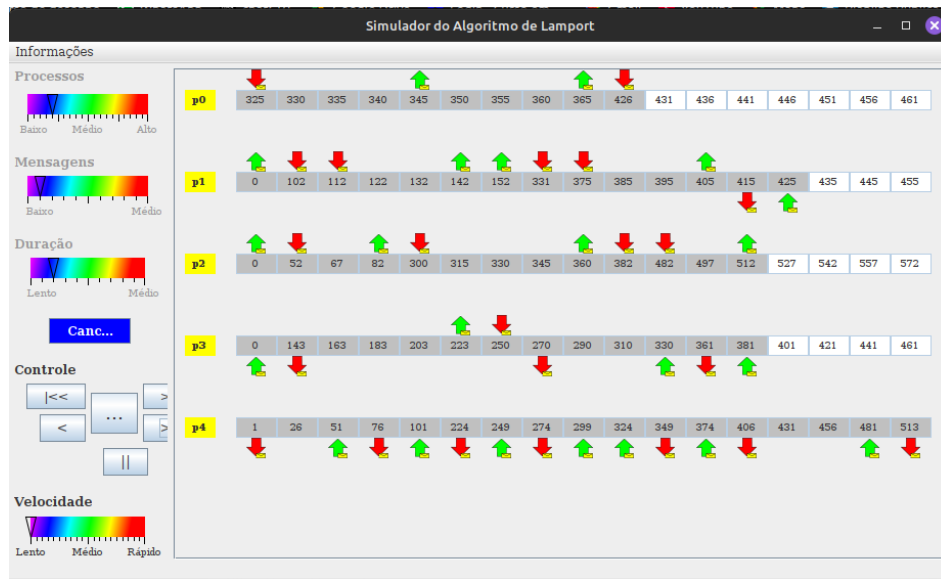


Figura 6: Imagem da iteração final da simulação

Para evitar a repetição de inúmeras iterações com situações semelhantes, aqui está a última iteração da simulação em que é possível ver que todos os processos passaram por pelo menos uma situação de sincronização de clocks. Com esse simulador é possível de ver de forma iterativa as iterações e as sincronizações com atualização de clock dos processos.

espero que goste da minha explicação, acredito que eu tenha explicado tudo da melhor maneira possível(dentro das minhas capacidades) :)