

# JANTAR DOS FILOSOFOS

Luccas Henrique Vieira Ribeiro  
Matheus Garcia Sampaio



# Tópicos

## A serem discutidos

- 01 O que é?
- 02 Representação para computação
- 03 Deadlock e Starvation
- 04 Solução
- 05 Implementação
- 06 Rede de Petri
- 07 Tina



# O QUE É?

O problema dos filósofos (ou problema dos filósofos comendo, em inglês Dining Philosophers Problem) é um problema clássico da computação concorrente proposto por Edsger Dijkstra em 1965. Ele é usado para ilustrar problemas relacionados à sincronização de processos que compartilham recursos limitados, como deadlock, fome (starvation) e concorrência.

Enunciado do problema:

Cinco filósofos estão sentados em uma mesa redonda. Cada filósofo alterna entre dois estados:

- Pensar
- Comer

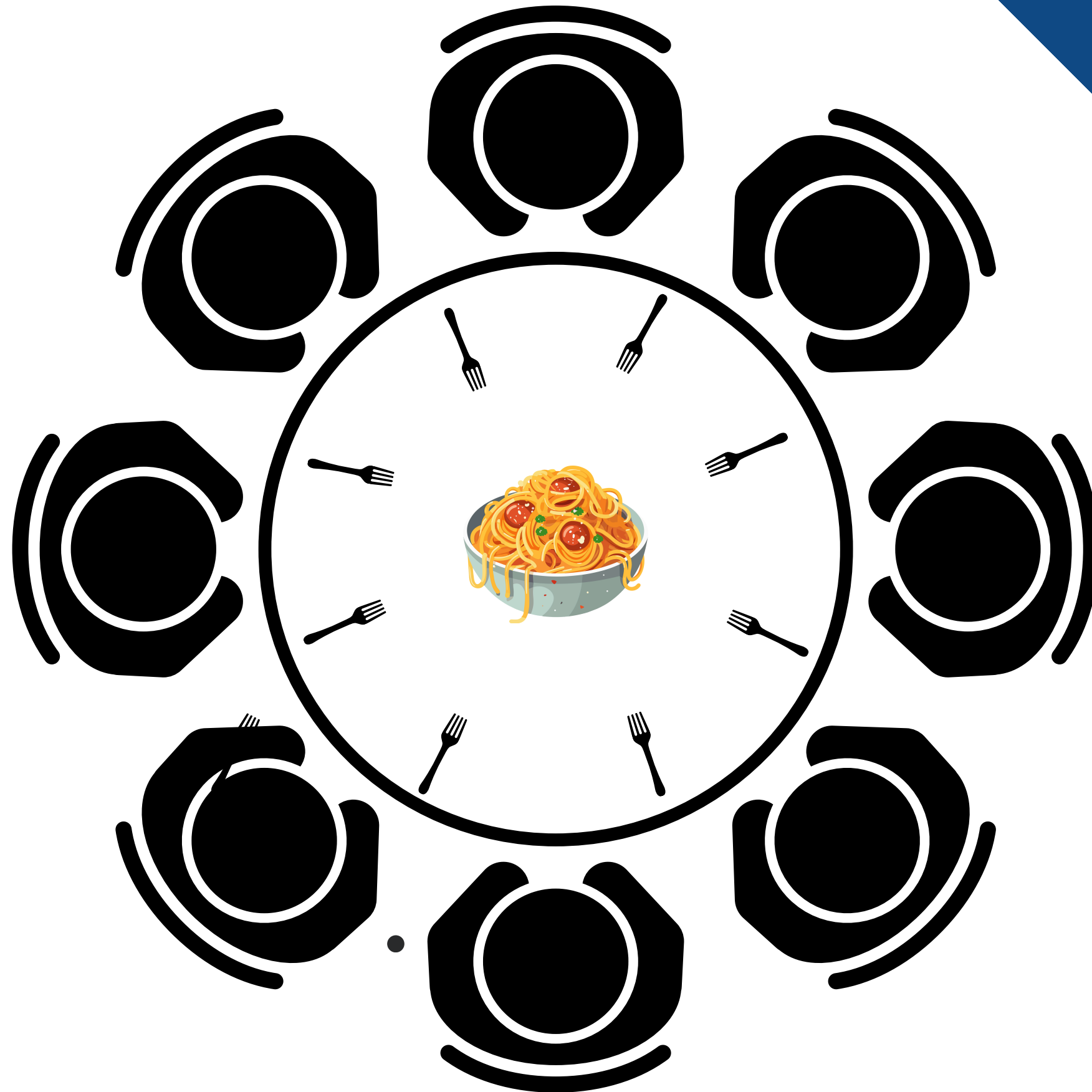
Para comer, cada filósofo precisa de dois garfos (um à sua esquerda e outro à sua direita). Porém, há apenas cinco garfos, um entre cada par de filósofos.

Regras:

- Um filósofo só pode comer se conseguir os dois garfos ao mesmo tempo.
- Depois de comer, ele coloca os garfos de volta e volta a pensar.



# REPRESENTAÇÃO





**EDSGER DIJKSTRA**

# REPRESENTAÇÃO PARA COMPUTAÇÃO

O problema dos filósofos representa, na computação, uma abstração de problemas reais de concorrência e compartilhamento de recursos em sistemas multitarefa ou multiprocessamento. Ele serve como uma metáfora para os desafios enfrentados quando vários processos ou threads precisam acessar recursos limitados de forma coordenada, evitando erros, bloqueios e ineficiências.

## CONCORRÊNCIA

Mostra como múltiplos processos ou threads podem tentar executar ao mesmo tempo e precisam de controle para evitar conflitos.



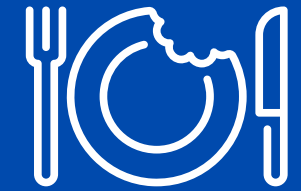
# PROBLEMAS



## Deadlock

Situação em que processos ficam bloqueados e esperando uns pelos outros indefinidamente.

No problema: todos pegam o garfo da esquerda e ficam esperando o da direita → ninguém consegue comer.



## Starvation

Um processo nunca recebe os recursos que precisa, porque os outros sempre têm prioridade.

No problema: um filósofo nunca consegue os dois garfos porque seus vizinhos sempre pegam primeiro.

# SOLUÇÕES

## 1. Mutex (Mutual Exclusion)

- Um mutex garante que apenas uma thread possa acessar um recurso (como um garfo) por vez.
- Cada garfo seria representado por um mutex.
- Antes de comer, o filósofo deve adquirir os dois mutexes correspondentes aos seus garfos.
- Solução simples, mas pode causar deadlock se todos tentarem pegar os mutexes ao mesmo tempo.

## 2. Semáforos

- Um semáforo pode controlar o número de acessos permitidos.
- Com um semáforo global (por exemplo, com valor 4), podemos permitir no máximo 4 filósofos pegando garfos ao mesmo tempo.
- Isso evita deadlock, porque sempre sobra pelo menos um garfo livre.