

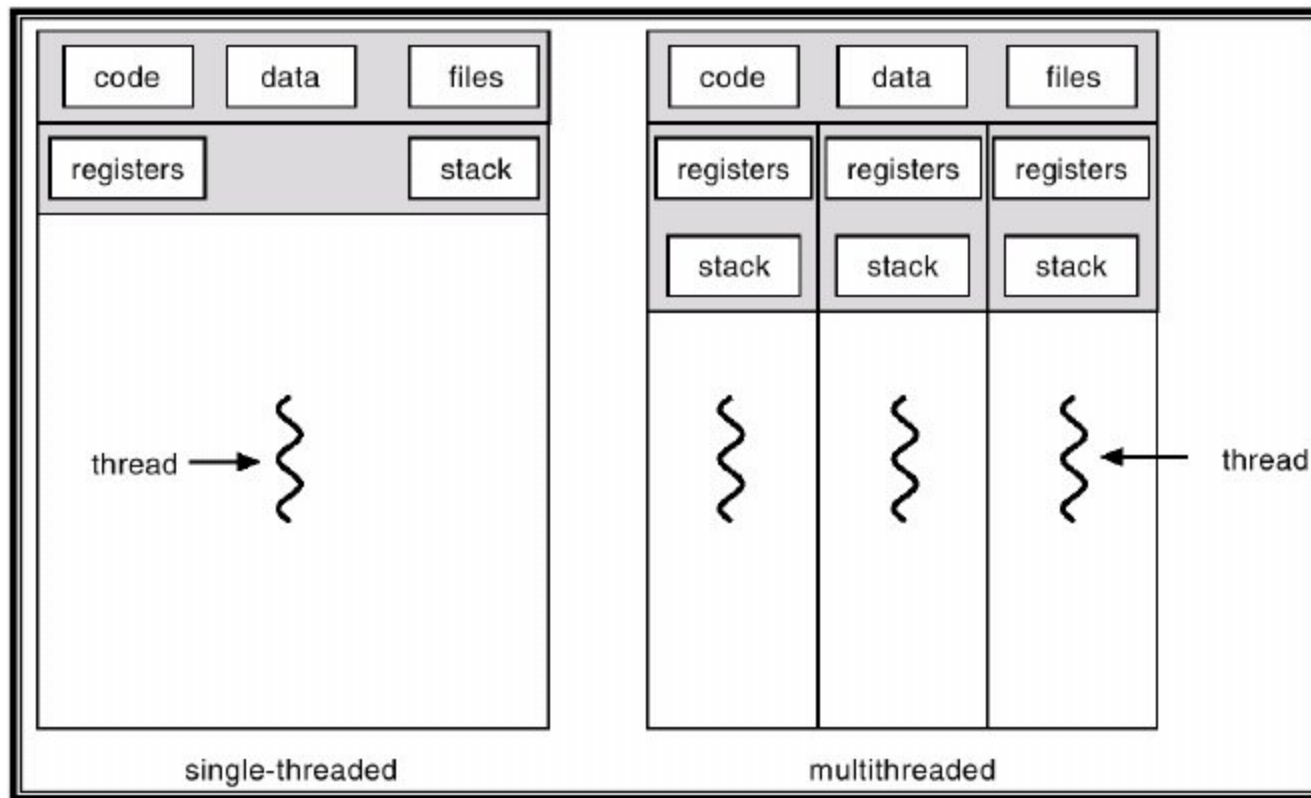
# Sistemas Hardware-Software

Aula 21 - Semáforos II e Modelos de Concorrência

2021 – Engenharia

Maciel Vidal  
Igor Montagner  
Fábio Ayres

# Processos e threads



# Conceito : Race Condition

"Ocorre quando a saída do programa depende da ordem de execução das threads"

Em geral ocorre quando

- uma variável é usada em mais de uma thread e há pelo menos uma operação de escrita.
- trabalhamos com os mesmos arquivos simultaneamente em várias threads

# Conceito : Região Crítica

"Parte do programa que só pode ser rodada uma thread por vez"

- elimina situações de concorrência
- elimina também toda a concorrência e pode se tornar gargalo de desempenho

# Mutex (Mutual Exclusion)

*Primitiva de sincronização para criação de regiões de exclusão mútua*

- Lock – se estiver destravado, trava e continua
  - se não espera até alguém destravar
- Unlock – se tiver a trava, destrava
  - se não tiver retorna erro

# Semáforos

"Inteiro especial que nunca fica negativo"

Só pode ser manipulado por duas operações atômicas

POST:

- Aumenta o valor em 1

WAIT:

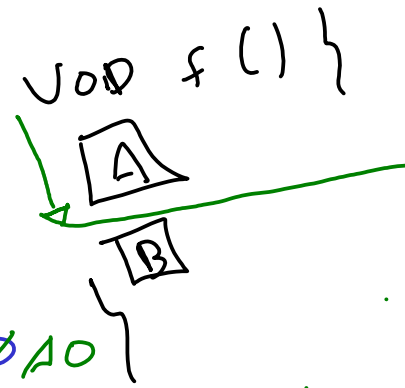
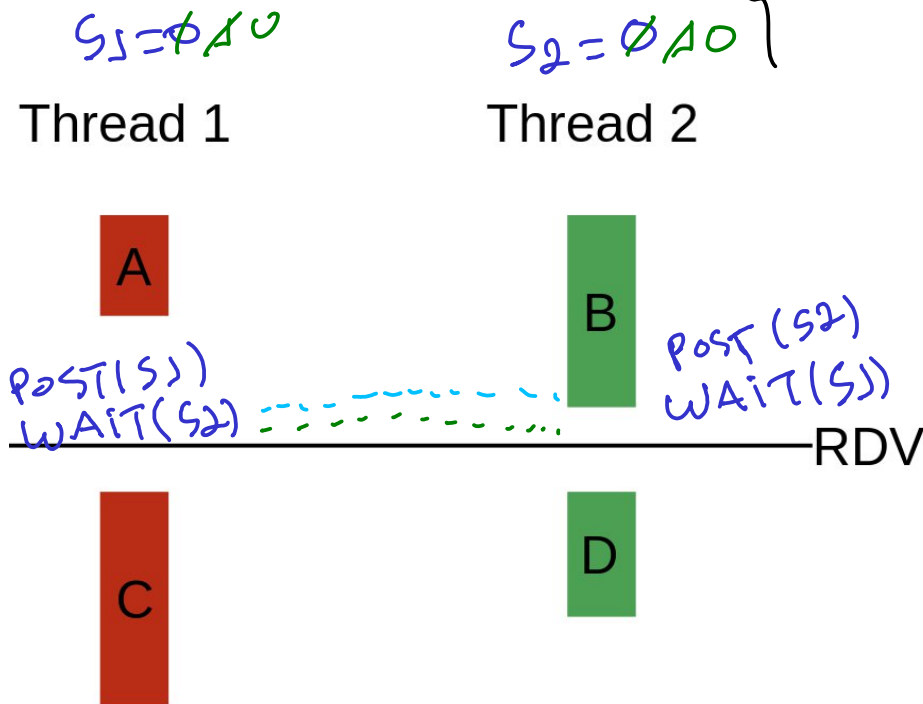
- Se for positivo, diminui em 1
- Se for 0 fica esperando;



# Correção

**Implementação do RDV com Semáforos POSIX**

# Relembrando RDV



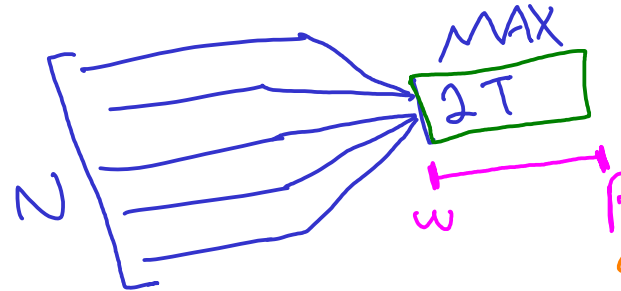
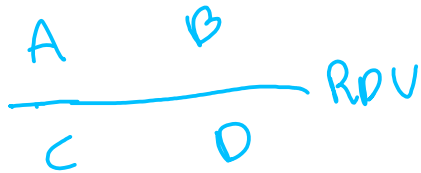
VA'LIDO  
 ABCD  
 BADC  
 ABCD  
 BADC  
 INVALIDO  
 ACBD  
 BDAC



SOMA GLOBAL

5.35

MUTEX  
SOMA += VALOR



$$S1 = 3/2 \neq \phi$$

## Atividade prática

$T_1$  WAIT(S)  
 $T_2$  WAIT(S)  
 $T_3$  WAIT(S)  
 $T_4$  WAIT(S)

ESPERA

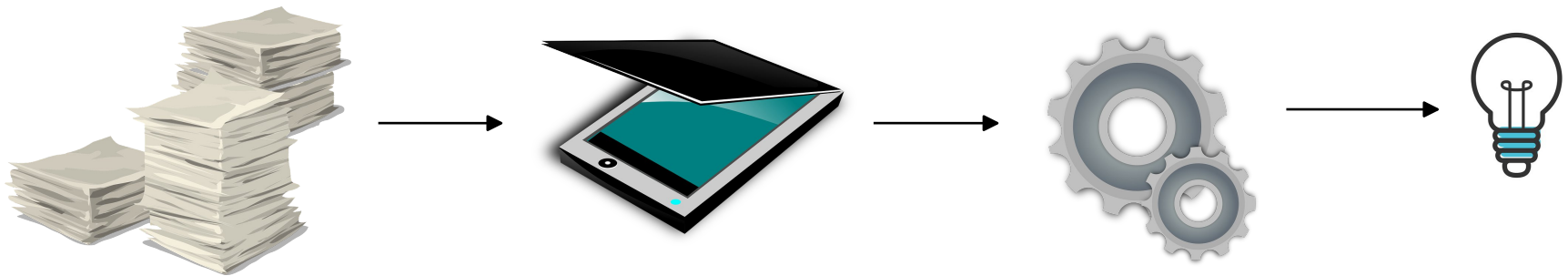
Aplicação de **Semáforos** POSIX (20 minutos)

1. Limitar a N threads o acesso a um recurso

30 TH  
3 TH

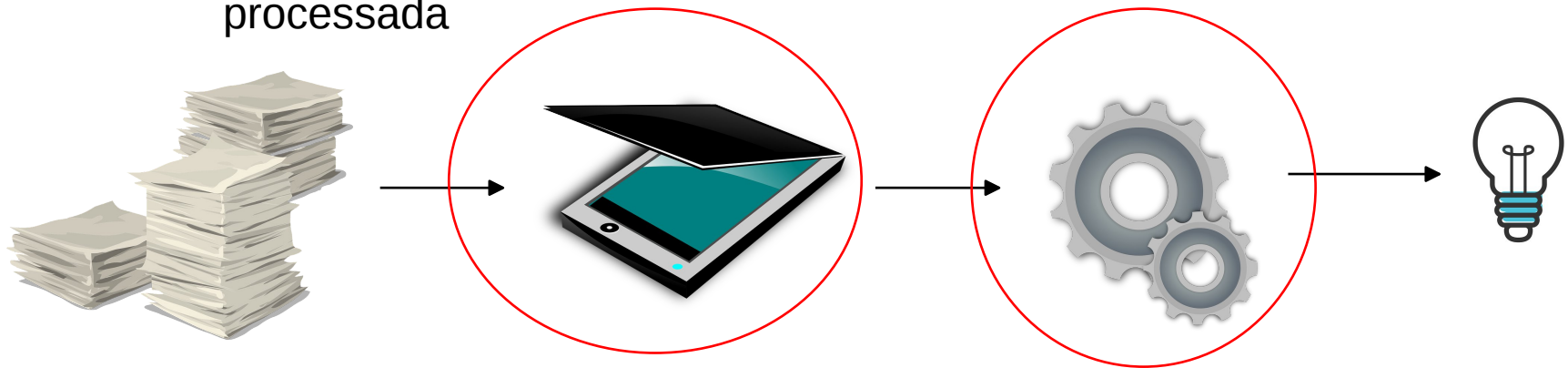
MAX N PODEM

# Problema – leitura de informações



# Exemplo 1 – produtor consumidor

Produtor: Escaneia e devolve imagem a ser processada



## Sincronização

1. Consumidor: espera produtor enviar item
2. Produtor: cria item e avisa Consumidor

Consumidor:  
transforma  
imagem em informação

# Exemplo 1 – produtor consumidor

Dois conjuntos de threads

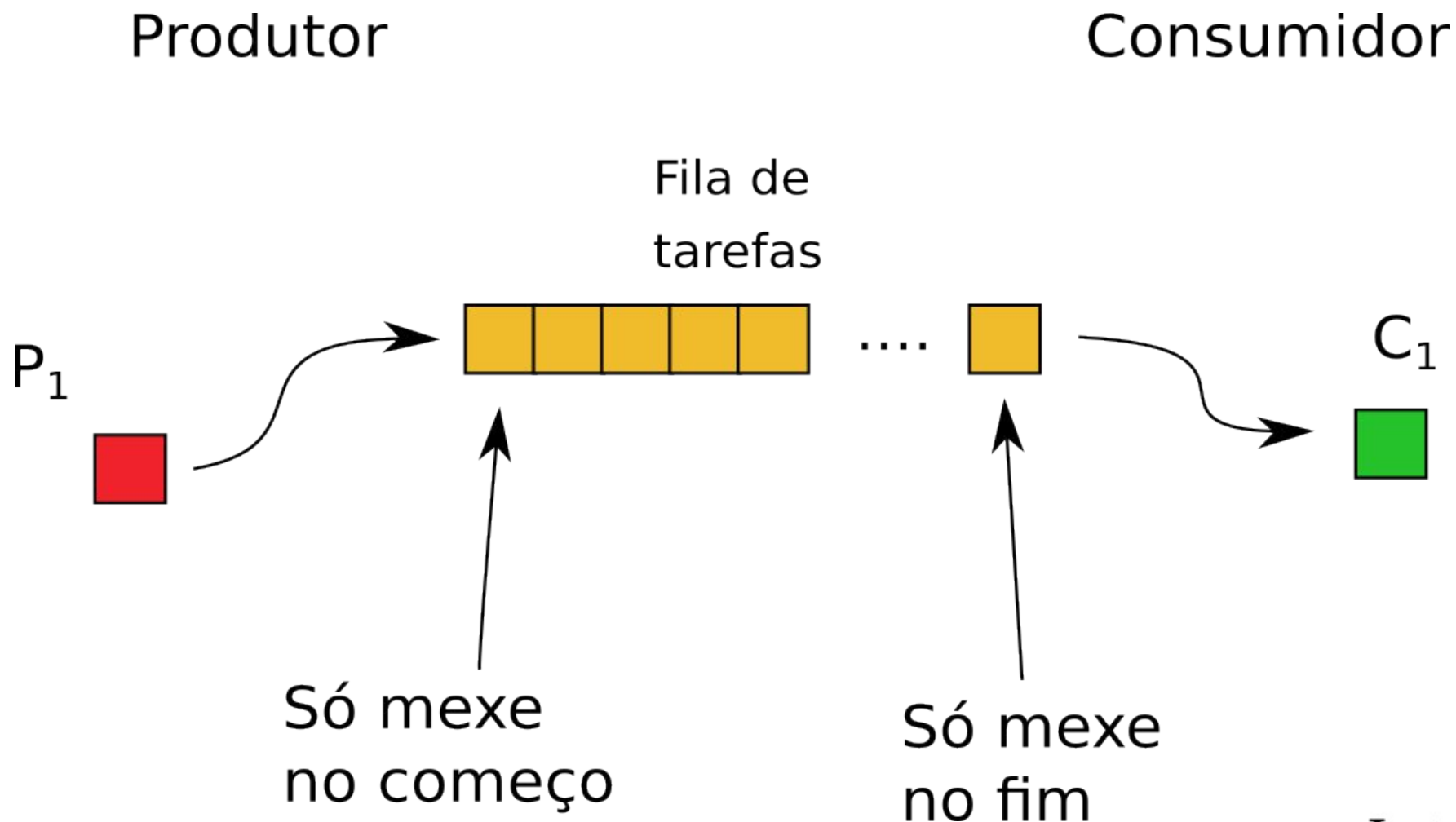
- Produzem tarefas a serem executadas

Pode depender de um recurso compartilhado  
controlar tamanho das tarefas.

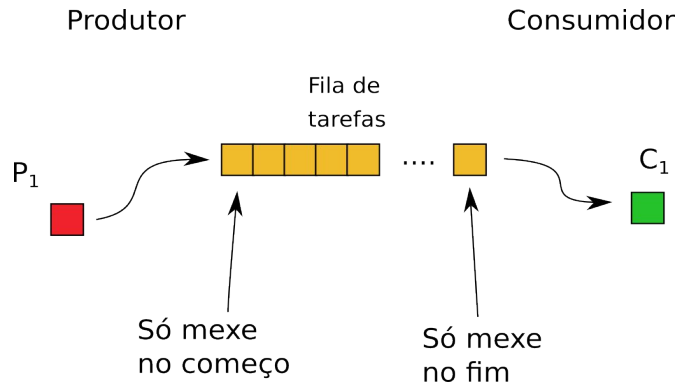
- Consomem as tarefas e as executam.

Cada consumidor não depende dos produtores nem de outros consumidores.

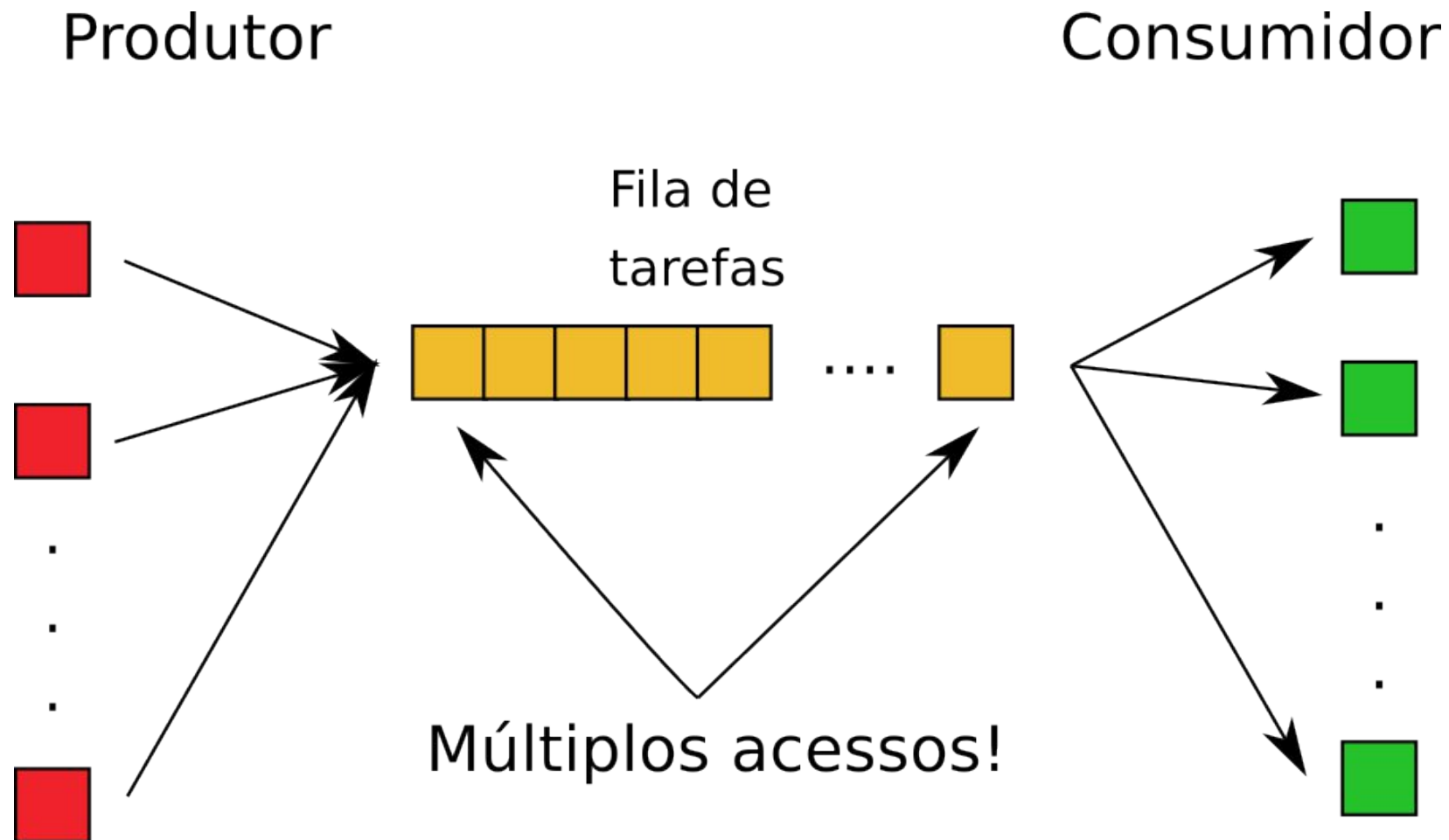
# Exemplo 1 – produtor consumidor



# Exemplo 1 – produtor consumidor



# Exemplo 1 – produtor consumidor



# Insper

[www.insper.edu.br](http://www.insper.edu.br)