

# Programação Paralela e Distribuída

Aula de Modelos de PD

*Prof. César A. F. De Rose*

# Roteiro

- Definições
- Modelos de Programação
  - Cliente/Servidor
    - Servidores Replicados
  - Broadcast
  - Token-passing
  - Probe/Echo
- Combinações
- Exercícios de Modelagem

# Definições

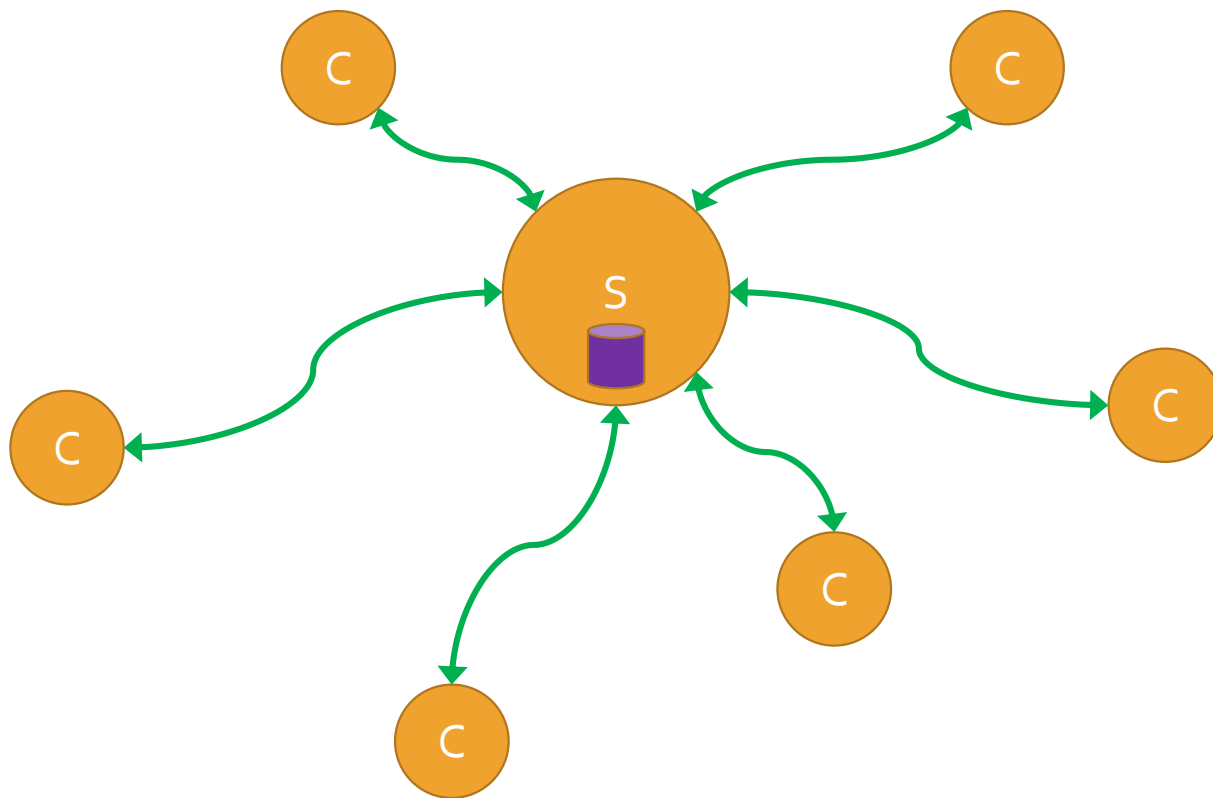
- Separação entre **Computação** e **Coordenação**
- Em um sistema Distribuído
  - **Computação** feita por processos que se encarregam de uma atividade executada de forma “independente” das outras
  - **Coordenação** é responsável pela comunicação e cooperação
    - “Cola” que liga as atividades dos processos em um só sistema (Gelernter e Carriero, 1992)
    - Também chamada de Articulação (De Rose)

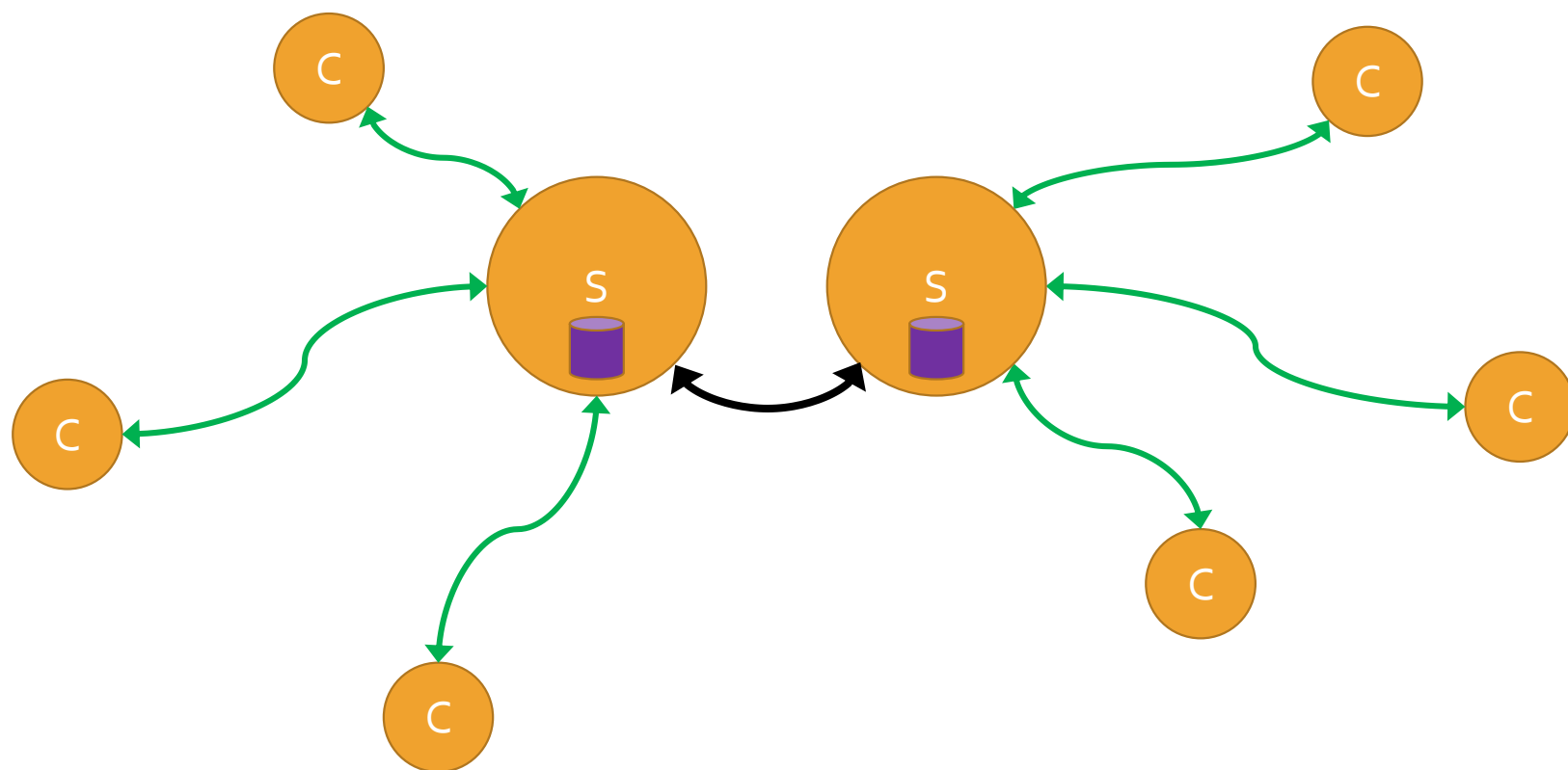
## Modelos de Programação Distribuída

- Expressão Modelo de Programação usada neste contexto para descrever como esta articulação será implementada
- Veremos alguns modelos clássicos para não partirmos do zero
  - Cada um com suas características específicas
    - Tanto vantagens quando desvantagens
  - Sistema é que escolhe o modelo e não ao contrário
    - Caso real pode necessitar uma combinação destes modelos

# Cliente/ Servidor

- Funcionamento
  - Servidor faz o papel de imagem única do sistema
  - Clientes consultam e atualizam este estado
- Características
  - Fácil implementação (+)
    - Processos sabem para quem perguntar
    - Comunicação *many-to-one*
  - Pouco escalável se muitos clientes (-)
  - Dedico um processo pra esta função (-)
  - Ponto único de falhas (-)
    - Melhoria: vários servidores, no entanto maior complexidade para manter replicação dos mesmos



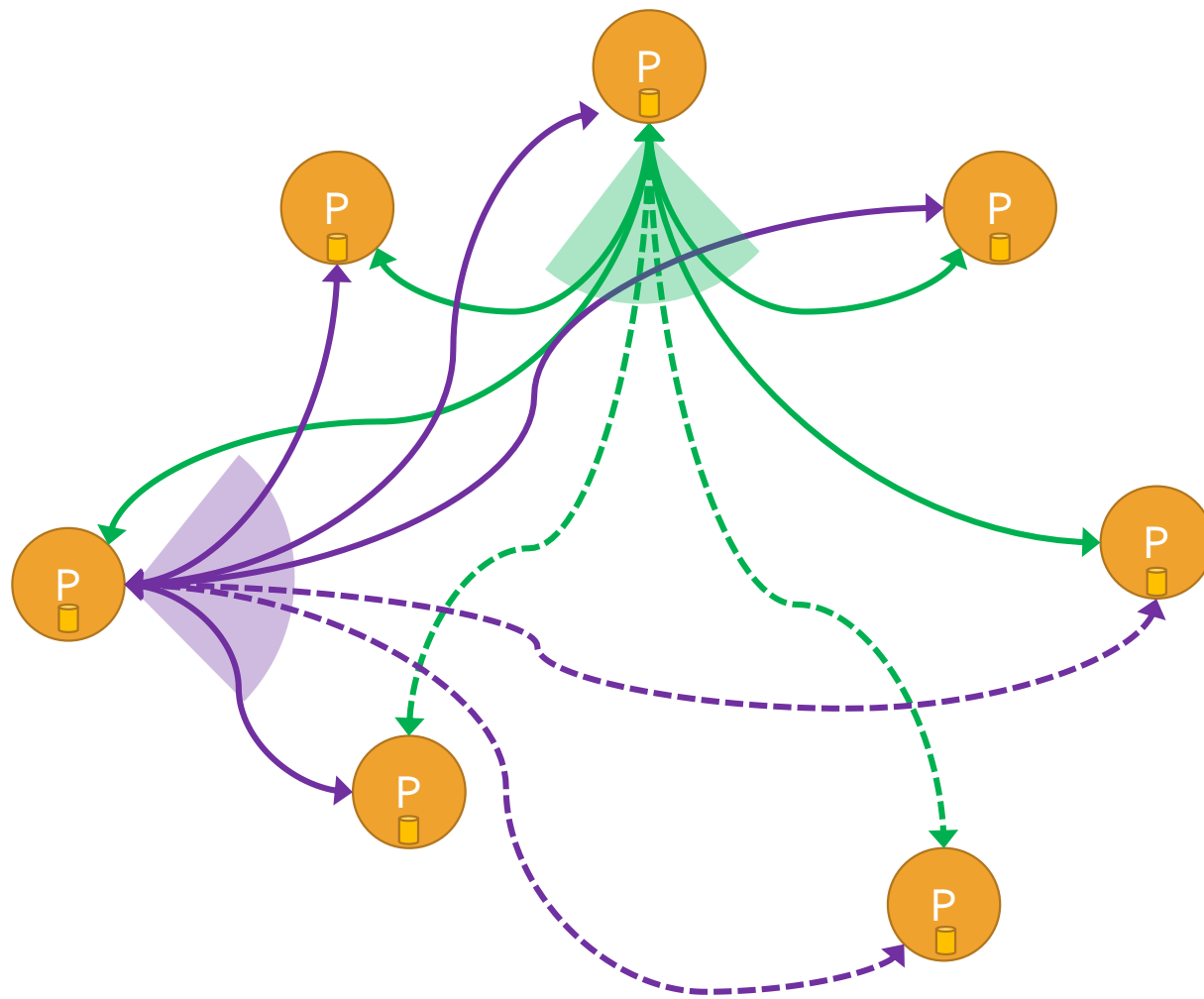


 Repositório central com redundância


# Broadcast

- Funcionamento
  - Não existe repositório de imagem única do sistema
  - Processos consultam os outros quando necessitam informações
  - Uso de comunicação coletiva
- Características
  - Não dedico processo para manter imagem única (+)
  - Alto custo com mensagens coletivas (-)
    - Comunicação *many-to-many*
    - Inundação da rede (flood)
  - Pouco escalável (-)
  - Mensagens coletivas assíncronas e propensas a falhas (-)
    - Melhoria: comunicação *one-to-one* com sorteio como no balanceamento de carga



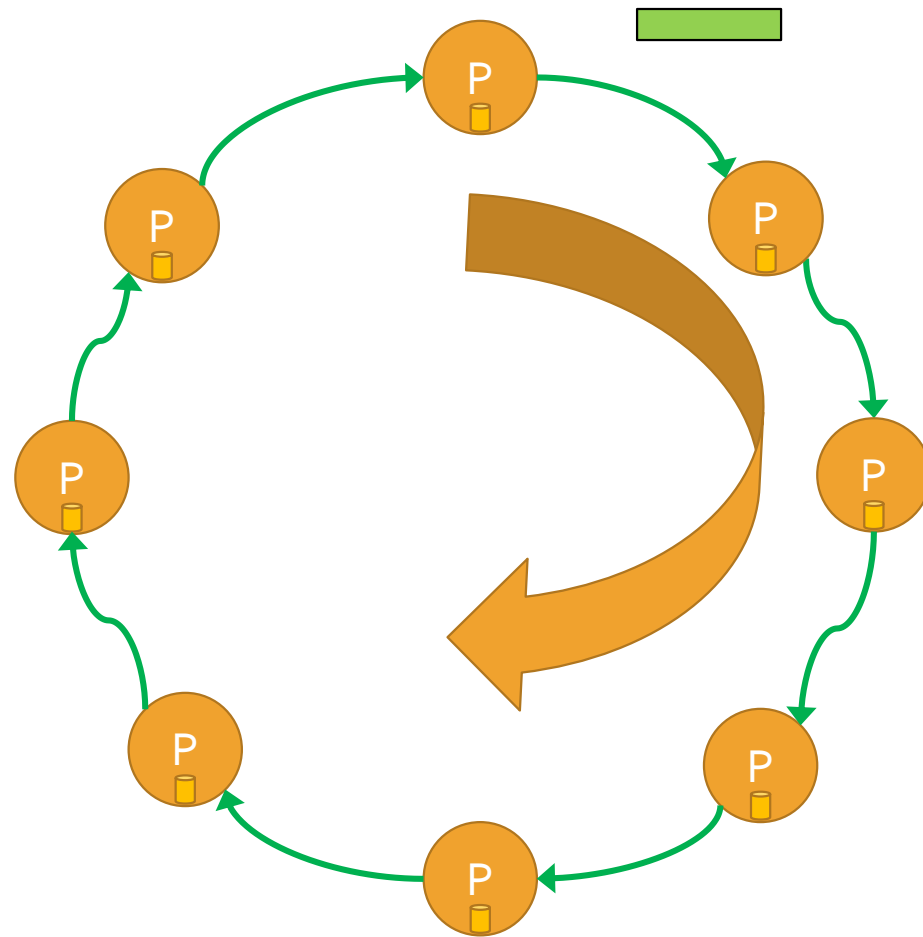




 Comunicação em grupo (broadcast)

 Repositório local parcial

# Token-passing

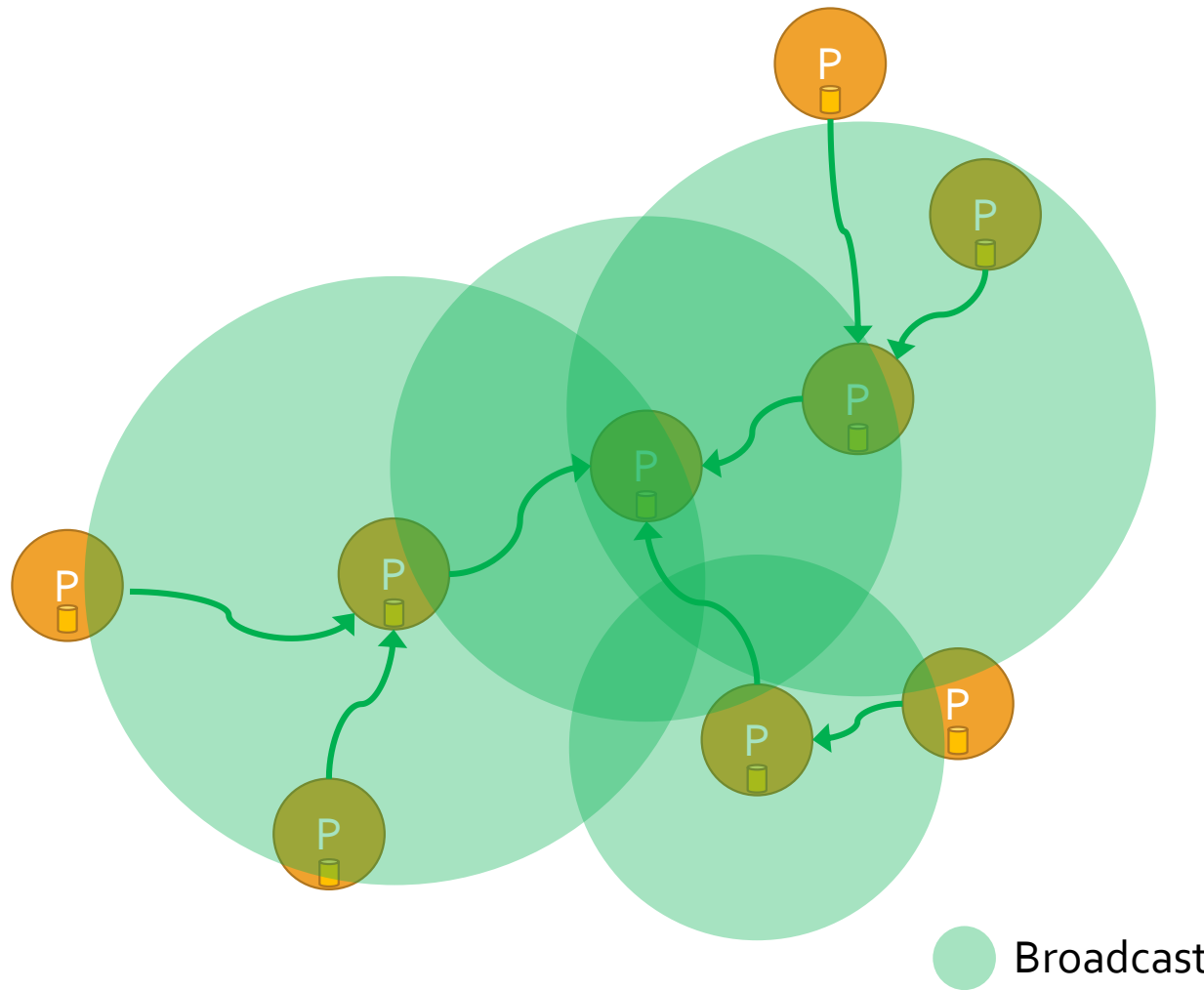
- Funcionamento
  - Não existe repositório de imagem única do sistema
  - Processos consultam os outros quando necessitam informações
  - Uso de comunicação ponto-a-ponto (anel)
- Características
  - Não dedico processo para manter imagem única (+)
  - Baixo custo com mensagens (+)
    - Comunicação *one-to-one*
  - Escalável (+)
  - Anel pode romper com falha de processo e precisa ser reestabelecido (-)



-  Mensagem (token)
-  Repositório local parcial

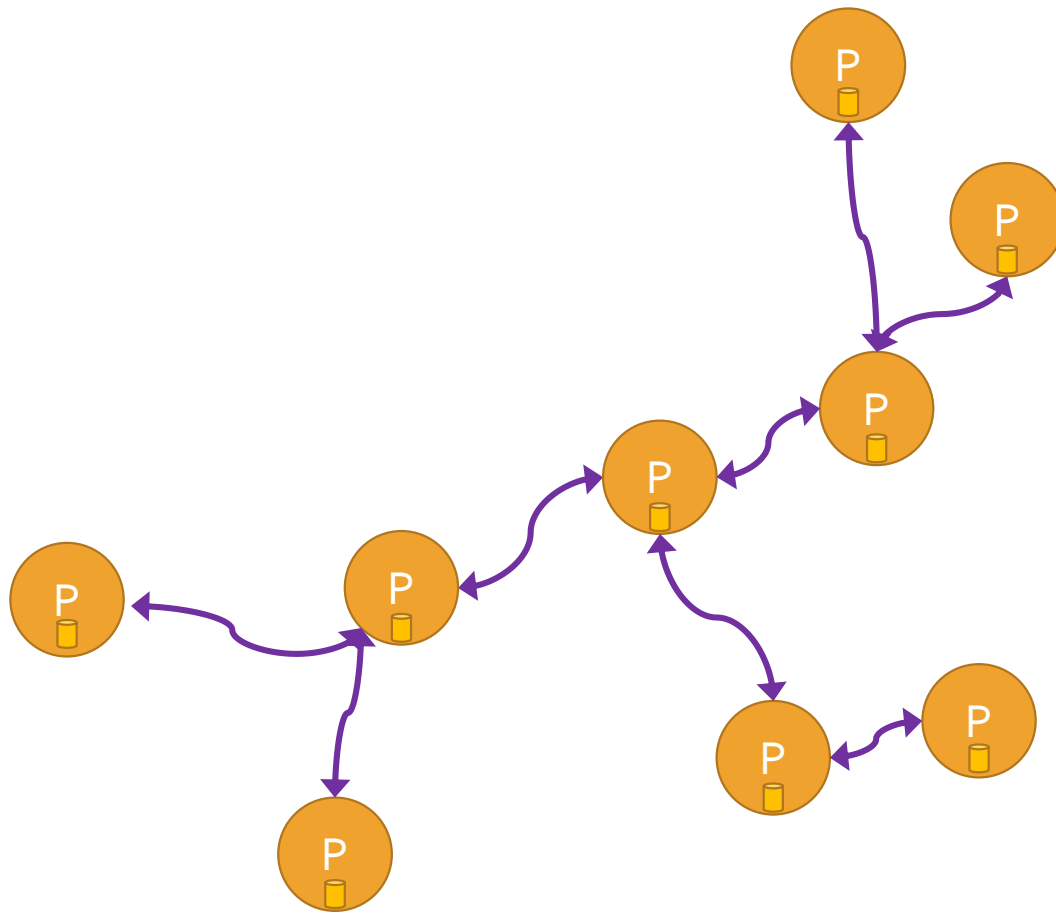
# Probe/echo

- Funcionamento
  - Não existe repositório de imagem única do sistema
  - Processos consultam os outros quando necessitam informações
  - Uso de comunicação ponto-a-ponto com vizinhos
- Características
  - Não dedico processo para manter imagem única (+)
  - Baixo custo com mensagens ponto-a-ponto (+)
    - Comunicação *one-to-one* com vizinhos
  - Escalável (+)
  - Estrutura da rede precisa ser inicialmente definida (-)
    - Muitas vezes é usada uma árvore
  - Pode ter que ser redefinida com falha de processos (-)



→ Mensagens de construção da topologia de comunicação

Repositório local parcial



↔ Mensagens de comunicação  
🗄 Repositório local parcial

# Probe/echo

- Funcionamento
  - Não existe repositório de imagem única do sistema
  - Processos consultam os outros quando necessitam informações
  - Uso de comunicação ponto-a-ponto com vizinhos
- Características
  - Não dedico processo para manter imagem única (+)
  - Baixo custo com mensagens ponto-a-ponto (+)
    - Comunicação *one-to-one* com vizinhos
  - Escalável (+)
  - Estrutura da rede precisa ser inicialmente definida (-)
    - Muitas vezes é usada uma árvore
  - Pode ter que ser redefinida com falha de processos (-)

# Combinações

- Na maioria dos casos são usadas combinações dos modelos e não apenas um
  - Por serviço
  - Por camada do sistema
  - De forma hierárquica

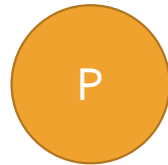
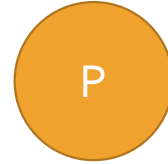


# Exemplo

- Algoritmo Distribuído de Eleição

# Exercícios de Modelagem

- Server Farm
  - Distribuição de arquivos a serem consultados
  - controle de diretório para consultas
  - criação e eliminação de arquivos com rebalanceamento de carga



# Exercícios de Modelagem

- Sistema de Banco
  - Central, Agências e Caixas Automáticos
  - Criação de conta, autenticação e operações de extrato, retirada e depósito de dinheiro
  - Implementar operações não idempotentes
  - Controle de contas tolerante a falhas



Dúvidas?