

# Arquiteturas de Sistemas Distribuídos, Paralelos e Concorrentes



Escola Politécnica – PUCPR  
Luiz A. de P. Lima Jr. • [luiz.lima@pucpr.br](mailto:luiz.lima@pucpr.br)

1

## Introdução

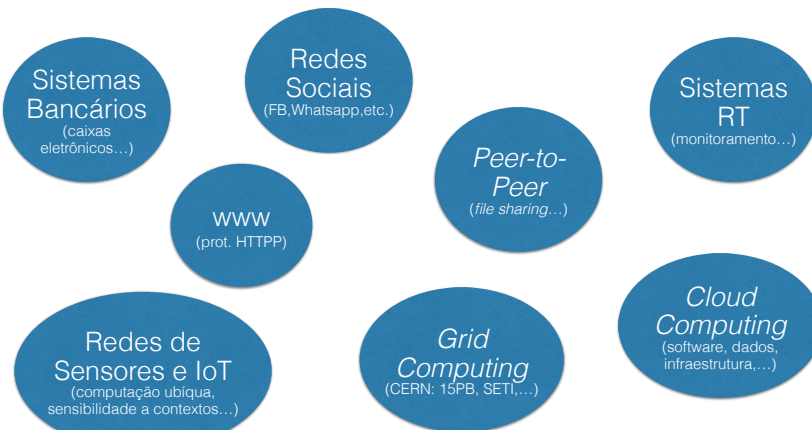
- Vida no século XXI:
  - ✦ **dependência** de serviços em rede
    - busca, vídeo-conferências, mercado de ações, Internet banking, compras on-line, redes sociais, jogos, notícias, etc.
- Rede:
  - ✦ conectividade básica
- Serviços sobre a rede:
  - ✦ exemplos de SISTEMAS DISTRIBUÍDOS

2



2

## Exemplos de Sistemas Distribuídos

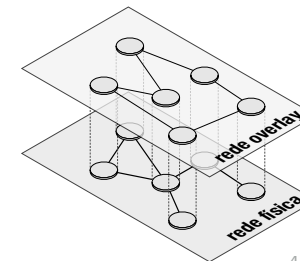


3

3

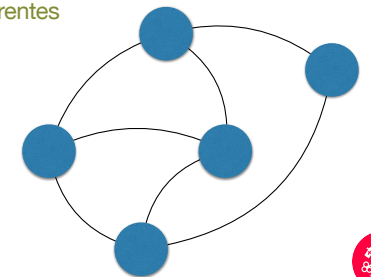
## (Re)visão Geral

1. O que são sistemas distribuídos?
  - ✦ “entidades” ou nós conectados
  - ✦ comunicação por trocas de mensagens
  - ✦ transparências
  - ✦ sistemas inerentemente paralelos/concorrentes



4

4

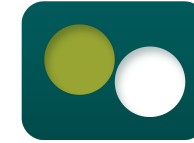


## Por que Sistemas Distribuídos?

- porque há sistemas **inerentemente distribuídos**
  - (e.g., *monitoramento remoto*, etc.)
- porque há necessidade de se melhorar o **desempenho**
  - (e.g., *balanceamento de carga*)
- porque é desejável **compartilhar recursos**
  - (“re-uso”)
- porque deseja-se manter o sistema “no ar” (**disponibilidade**)
  - (tolerância a falhas e “degradação progressiva”)

5

5



ASDPC

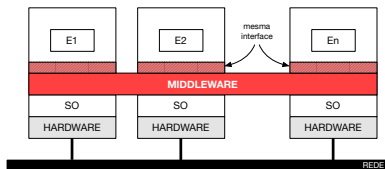
## Como construir um sistema distribuído?

Arquitetura de Sistemas Distribuídos, Paralelos e Concorrentes

6

## Como implementar um Sistema Distribuído?

- Interfaces de programação de redes — *socket API*
  - ✦ baixo nível → complexidade
  - ✦ suscetível a erros
- **Middleware** e sistemas distribuídos
  - ✦ plataformas de software que auxiliam o desenvolvimento
  - ✦ organizados em camadas (esconde diferenças de SO, HW, ...)



7

7

## Como implementar um Sistema Distribuído?

- **Middleware:**
  - ✦ recursos para facilitar comunicação entre entidades computacionais
    - “transparências”
  - ✦ serviços adicionais:
    - segurança
    - contabilidade
    - localização
    - etc...
  - ✦ “container” de funcionalidades necessárias para um SD:
    - comunicação (*Remote Procedure Call* — *RPC*)
    - transações (propriedades ACID...)
    - composição de serviços (programar = montar “quebra-cabeça” → *Web Services*)
    - confiabilidade (e.g., garantias de entrega, tolerância a falhas)

8

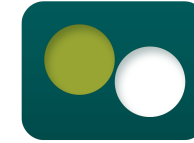
8

## RESUMO

1. O que são sistemas distribuídos?
  - ✦ "entidades" ou nós conectados
  - ✦ comunicação por trocas de mensagens
  - ✦ transparência
  - ✦ inerentemente paralelos/concorrentes...
2. Por que sistemas distribuídos?
  - ✦ desempenho
  - ✦ disponibilidade (tolerância a faltas)
3. Como implementar sistemas distribuídos?
  - ✦ comunicação de baixo nível (*bytes*)
  - ✦ *middleware*
  - ✦ *algoritmos e técnicas fundamentais*
  - ✦ *tolerância a faltas*

9

9



ASDPC

## A Disciplina

Arquitetura de Sistemas Distribuídos, Paralelos e Concorrentes

10

## A Disciplina de ASDPC

1. Conceitos Fundamentais
  - ✦ heterogeneidade e interoperabilidade
  - ✦ modelos de comunicação e aspectos práticos
2. Coordenação Distribuída de Processos
3. **Middleware\***
  - ✦ princípios gerais
  - ✦ classes (OD, WS, MOM, DSM, ...)
4. **Algoritmos Distribuídos\***
  - ✦ especificação e formalização
  - ✦ algoritmos distribuídos fundamentais
5. Computação na Presença de Falhas
  - ✦ estudos de caso
    - *diferentes linguagens de programação (C++, Java, Python, ...)*
  - ✦ Replicação e Consistência
  - ✦ Consenso Distribuído

11

11



## Método

- Teoria + Prática (PBL/PjBL)
- Prática:
  - *plataformas de desenvolvimento/middlewares SD*
  - *Python; C++; Java*
  - ✦ Opções:
    - *acesso remoto ao servidor da PUCPR*
    - *instalação própria (responsabilidade do estudante!)*
  - ✦ Ferramentas de acesso remoto
    - *RDP*
    - *SSH (MobaXTerm, ...)*

12

12



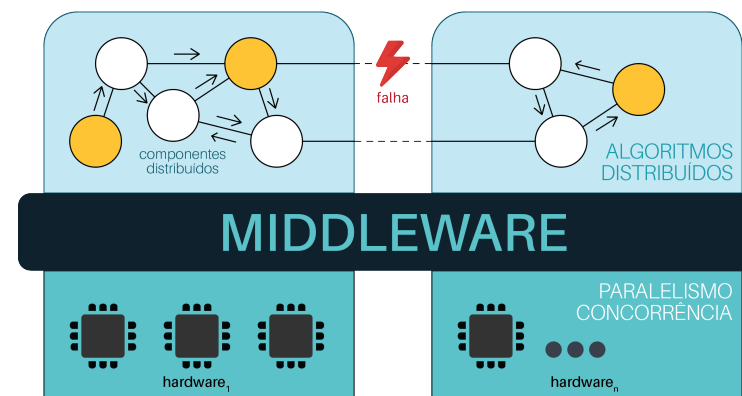
# Plano de Ensino

Arquitetura de Sistemas Distribuídos, Paralelos e Concorrentes

13



## Mapa Mental



14



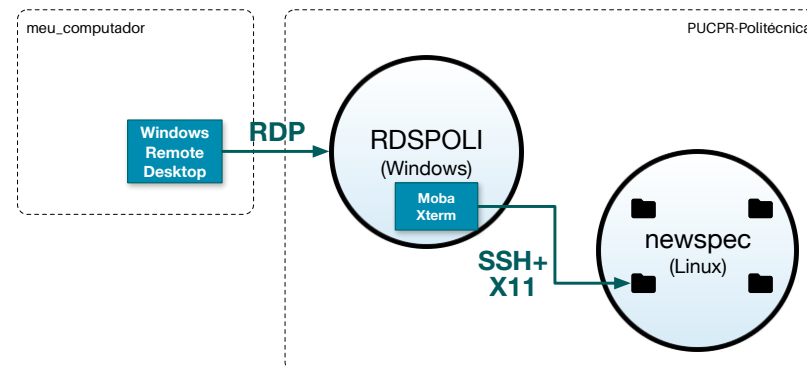
# Atividades Práticas e Ambiente de Trabalho

Arquitetura de Sistemas Distribuídos, Paralelos e Concorrentes

15



## Ambiente de Trabalho



16