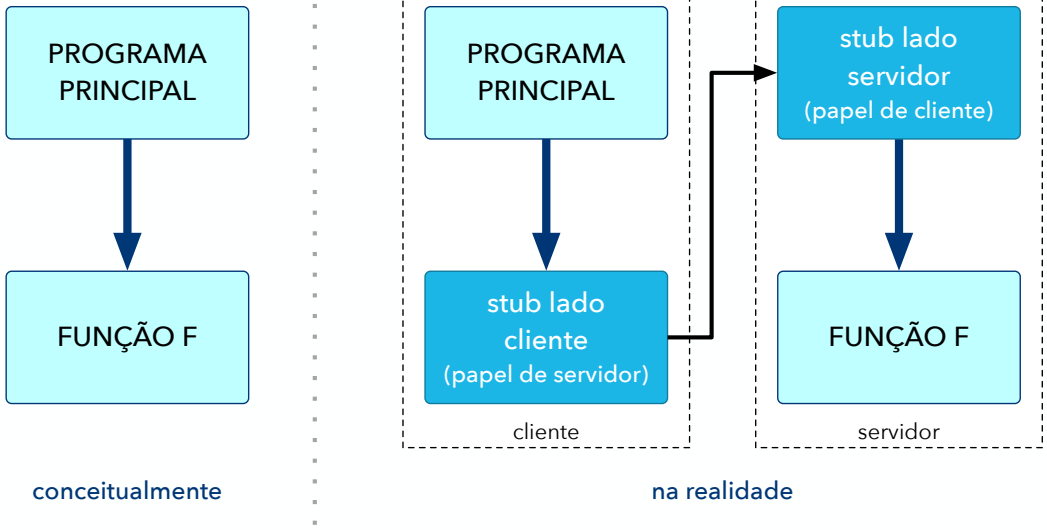
Remote Procedure Calls (RPC)

- Código adicional (***stubs/proxies***):
  - *Marshalling*
  - Envia mensagem
  - Espera resposta
  - *Unmarshalling*



© Luiz Lima Jr.

Tecnologias de Middleware

Principais Tecnologias de Middleware

- **Objetos Distribuídos**:
  - **CORBA** (OMG)*
  - RMI, J2EE (Oracle)
- **Web services**
  - **JWS/Glassfish** (Oracle)*
  - Axis (Apache)
  - .Net (Microsoft)
  - **Flask-RESTFUL***
- **Orientados a Mensagens**
  - DSS (OMG)
  - **RabbitMQ** (Pivotal)*
- **Memória Compartilhada Distribuída**
  - Java TupleSpaces (Oracle)
  - **CSpaces***

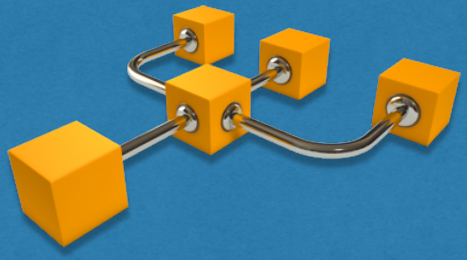
© Luiz Lima Jr.
CC BY-NC

Entidades Comunicantes e Paradigmas de Comunicação

Entidades Comunicantes		Paradigmas de Comunicação		
Orientadas a Sistemas	Orientadas a Problemas	IPC	Inovação Remota	Comunicação Indireta
Processos	Objetos Distribuídos Componentes Distribuídos Serviços	Trocas de Mensagens (por *sockets*)	Requisição-Reposta (simulado: "Serv-Ops") RPC RMI	*Publish-Subscribe* Comunicação em Grupo Espaço de Tuplas

© Luiz Lima Jr.

Objetos e Objetos Distribuídos



Middleware

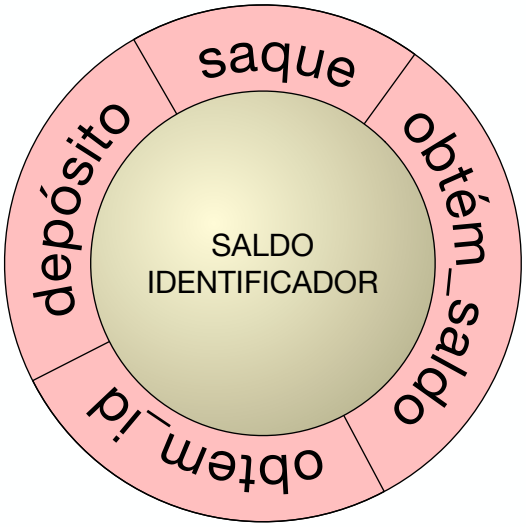
Problemas

- O que é um **objeto distribuído**?
- Como **compartilhar** estes objetos? Como lidar com a **distribuição**?
- Como lidar com a **heterogeneidade**?
- Como garantir **escalabilidade**?
- Como introduzir **tolerância a faltas** e outras **características não funcionais**?
- **Meta-programação**?

© Luiz Lima Jr.
CC BY-NC

Objetos

- **Objetos**:
  - Dados → **ATRIBUTOS**
  - Operações → **MÉTODOS**

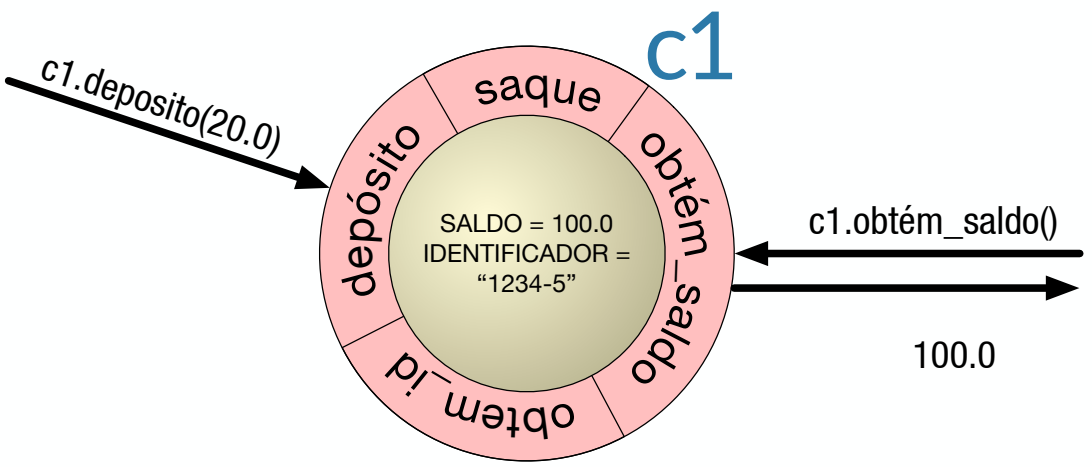


- **Interfaces**:
  - grupos de métodos
  - para comunicação:
    - ✦ *o que importa é sabe O QUE o objetos faz e não COMO*

© Luiz Lima Jr.
CC BY-NC

Comunicação entre Objetos

- **Comunicação** entre objetos:
  - exclusivamente através de trocas de mensagens
- **Mensagens**:
  - Objeto-alvo
  - Método
  - Parâmetros



© Luiz Lima Jr.

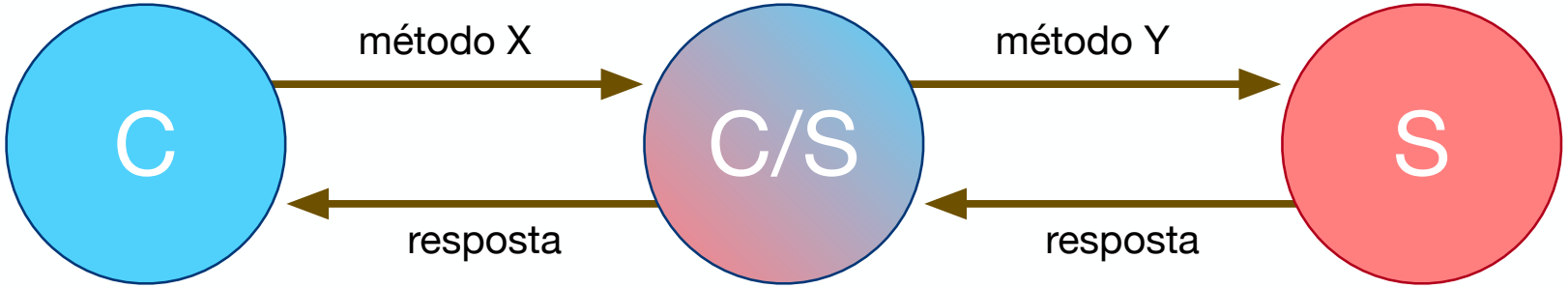
Objetos x Objetos Distribuídos

- **Objetos**:
  - Internos a um programa (reuso de código)
- **Objetos Distribuídos**:
  - Disponíveis na rede
  - Componentes “autônomos”
  - Relacionamentos cliente-servidor (ou P2P)

© Luiz Lima Jr.
CC BY-NC

Comunicação entre Objetos Distribuídos

- Paradigma **cliente-servidor**
- Relações C/S são dinâmicas
- Invocações estilo **RPC** ("*Remote Procedure Call*")



© Luiz Lima Jr.
CC BY-NC