FIAP GRADUAÇÃO

FIAP

ENTERPRISE APPLICATION DEVELOPMENT

Prof. Me. Thiago T. I. Yamamoto

#09 – REMOTE METHOD INVOCATION

 *thiagoyama@gmail.com*  *thiagoyama*

TRAJETÓRIA

- JPA Introdução
- JPA API
- Design Patterns e JUnit
- Relacionamentos
- JPQL
- Mapeamento Avançado
- Serialização de objetos e Sockets
- Remote Method Invocation

#09 - AGENDA

- Introdução ao RMI
- Cliente x Servidor
- Objetos Remotos com RMI
- Implementação do Servidor
- Desenvolvimento do Cliente

INTRODUÇÃO

FIAP

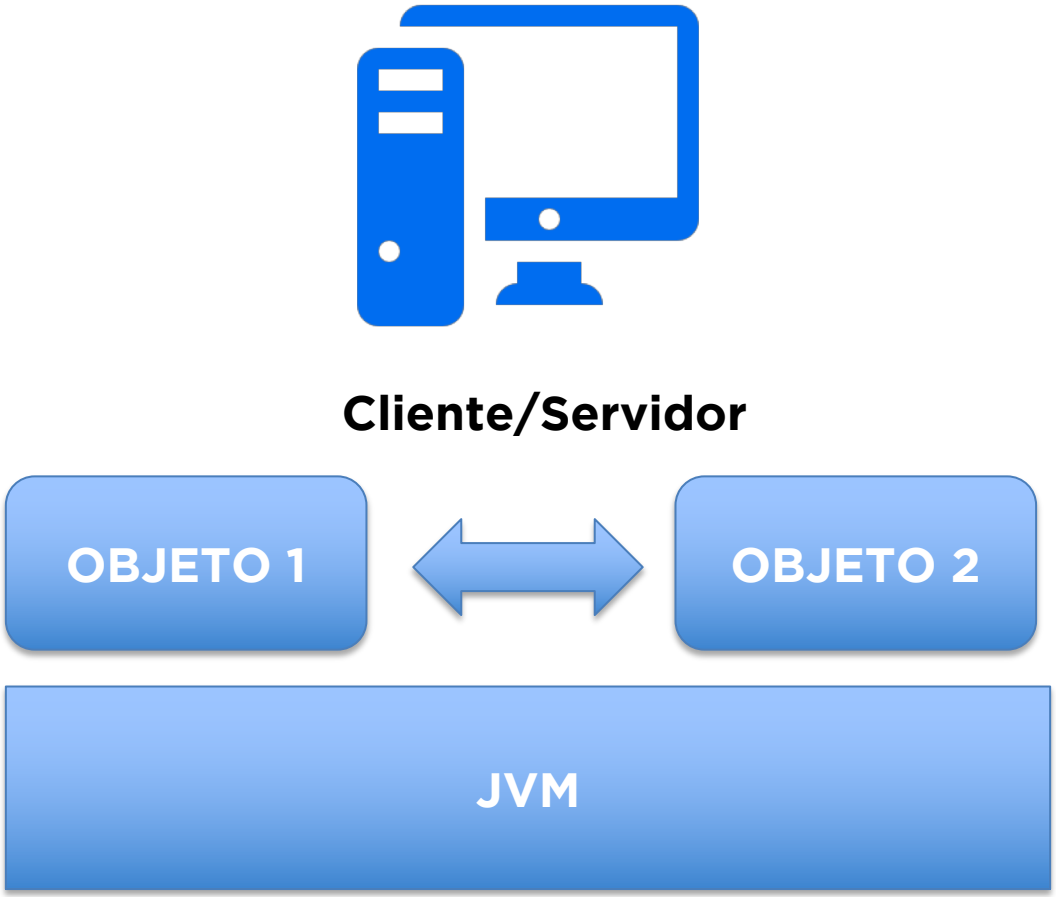
- **RMI – Remote Method Invocation**, ou chamada remota de método;
- Possibilita a **comunicação** remota entre objetos Java;
- Separação da **especificação (interface)** e **implementação (objeto remoto);**
- RMI lida com questões de **comunicação entre objetos remotos** e **serialização** de objetos;
- Por que estudar RMI?
  - Entender alguns princípios sobre objetos distribuídos para fundamentar conceitos futuros, como **EJB**;

CLIENTE X SERVIDOR

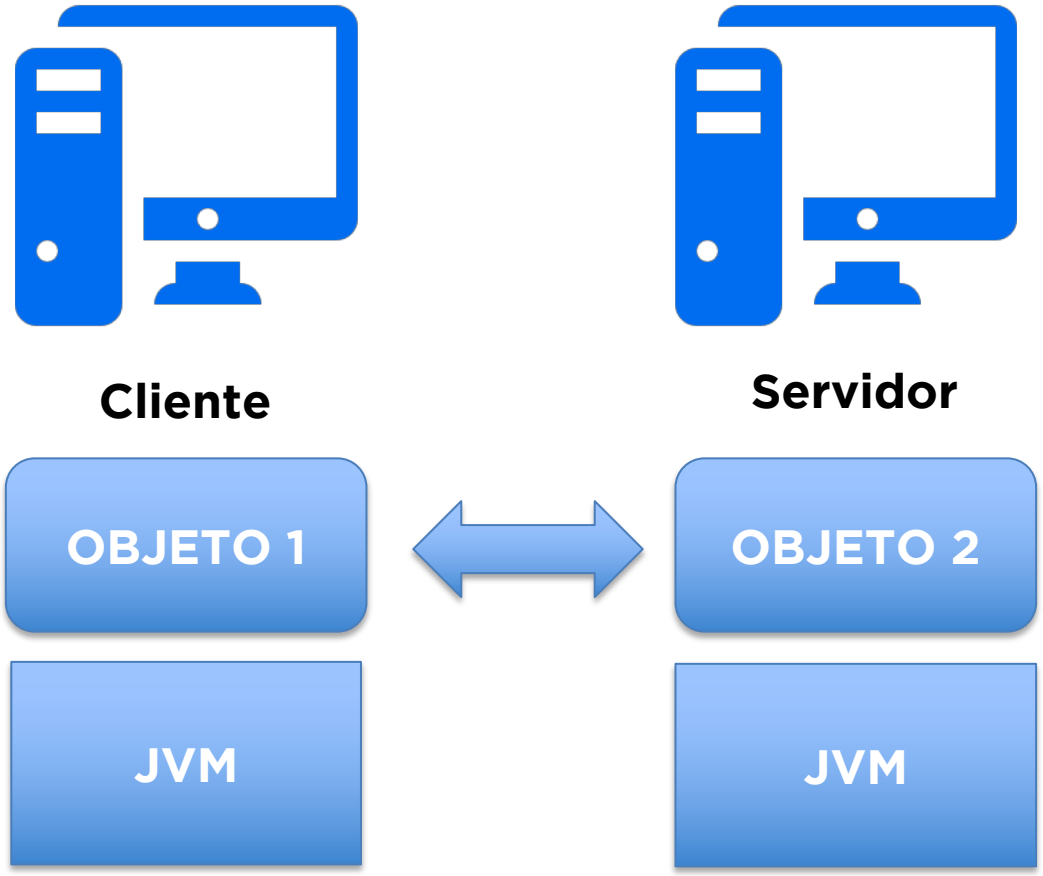
FIAP

- Aplicações típicas **RMI** são compostas por dois programas: **cliente e servidor;**
- **Servidor**:
  - Criar objetos remotos;
  - Tornar a referência remota a estes objetos acessível;
  - Aguardar as requisições dos clientes aos objetos remotos;
- **Cliente**:
  - Obter referência a um ou mais objetos remotos;
  - Acionar os métodos desses objetos;

OBJETOS LOCAIS

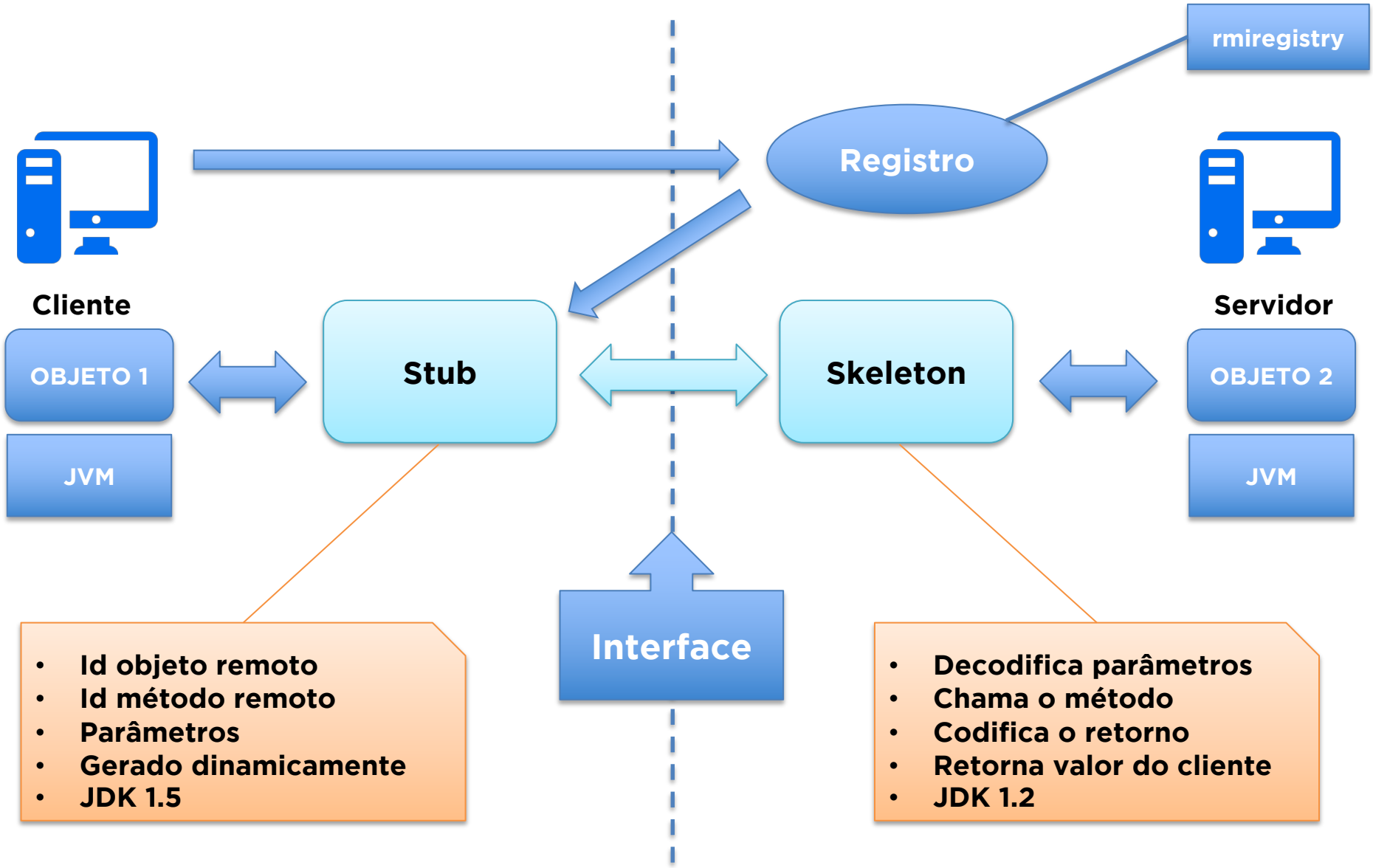


OBJETOS REMOTOS



OBJETOS REMOTOS COM RMI

FIAP



IMPLEMENTAÇÃO

RMI - PASSOS

- Para implementar um **objeto remoto**, utilizando o protocolo **RMI**, você deve:
  1. Definir uma **subinterface** de ***Remote*** contendo as operações que poderão ser acessadas remotamente;
  2. Implementar **a interface**, isto é, as operações da mesma em uma classe concreta;
  3. Criar uma classe que **registre o objeto remoto** junto ao servidor de registro;
  4. Criar o **cliente** que acessará o objeto remoto e executará as operações desejadas;

INTERFACE ENTRE COMPONENTES

FIAP

- Todos os **métodos** a serem **acionados remotamente**, devem ser declarados em uma interface, que estende a interface **java.rmi.Remote**;
- Todos os métodos devem lançar a exceção **java.rmi.RemoteException** em suas declarações;

```
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface Mensagem extends Remote {

        public String getMensagem() throws RemoteException;

}
```

IMPLEMENTAÇÃO DA INTERFACE

FIAP

- A **implementação** dos **métodos** ocorre em uma **classe concreta** que implemente a **interface remota**;

```
public class MensagemImpl
        extends UnicastRemoteObject implements Mensagem{

        public MensagemImpl() throws RemoteException{ }

        public String getMensagem() throws RemoteException {
                return “Boa noite”;
        }

}
```

IMPLEMENTAÇÃO DO SERVIDOR

O **servidor** deve tornar **disponível** o **objeto remoto** para atender às solicitações dos clientes por meio dos passos abaixo:

- Iniciar o serviço de registro de nomes por meio do método estático **createRegistry(int P1)** da classe **LocateRegistry**:
  - `LocateRegistry.createRegistry(1099);`

- Com o objeto remoto criado é necessário associá-lo ao serviço de registro de nomes (*rmiregistry*) por meio do método **bind(String P1, Remote P2)** da classe **Naming**, onde P1 representa o nome do objeto e P2 o objeto em si:
  - `Naming.bind("msg", stub);`

IMPLEMENTAÇÃO DO SERVIDOR

Caso o objeto não extenda UnicastRemoteObject , é necessário:

- Exportar o objeto remoto com o método estático **exportObject(Remote P1, int P2)** *da* classe **UnicastRemoteObject** que retorna um *stub* do objeto remoto a ser repassado aos clientes:
  - `MsgInt stub = (MsgInt) UnicastRemoteObject.exportObject(new Msg(), 1099);`

IMPLEMENTAÇÃO DO CLIENTE

FIAP

- O **cliente** de um objeto remoto pode residir tanto na máquina do próprio servidor quanto em um computador fisicamente separado;
- É necessário **localizar** o objeto remoto por meio do serviço de registro de nomes com o método estático **lookup(String P1)** da classe **Naming**, onde P1 corresponde à URL do serviço de registro de nomes no formato:
  - `rmi://IP_SERVIDOR:PORTA/NOME_REGISTRO_OBJETO`

```
Mensagem m = (Mensagem)
        Naming.lookup("rmi://127.0.0.1:1099/msg");
m.getMensagem();
```

VOCÊ APRENDEU

- Trabalhar com **objetos** e **chamadas** de **métodos remotos** através de **RMI**;
- **Implementar** o **servidor** desenvolvendo a **interface**, **classe** remotos e **registrando** esse objeto para o acesso;
- **Acessar** o objeto remoto **implementando** um **cliente**;

"A vida vai ficando cada vez mais dura perto do topo"
Friedrich Nietzsche