



# **Introdução ao Processamento Paralelo e Distribuído**



**Videoaula**

# Notas dos slides

## APRESENTAÇÃO

O presente conjunto de slides pertence à coleção produzida para a disciplina *Introdução ao Processamento Paralelo e Distribuído* ofertada aos cursos de bacharelado em Ciência da Computação e em Engenharia da Computação pelo Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas.

Os slides disponibilizados complementam as videoaulas produzidas e tratam de pontos específicos da disciplina. Embora tenham sido produzidos para ser assistidos de forma independente, a sequência informada reflete o encadeamento dos assuntos no programático previsto para a disciplina.





# **Programação em Ambiente com Memória Distribuída**

RPC e RMI



**Rien ne se perd, rien ne se crée,  
tout se transforme.**

Antoine Lavoisier

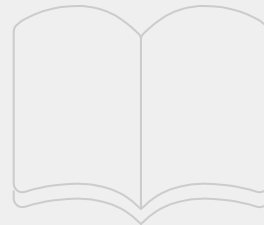
# Notas da videoaula

## DESCRIÇÃO

Nesta videoaula são apresentados os principais conceitos associados às ferramentas de programação RPC e RMI.

## OBJETIVOS

Nesta videoaula o aluno conhecerá os fundamentos da programação com duas importantes ferramentas para programação distribuída: RPC e RMI.



# Princípio

A ideia é oferecer uma semântica de utilização dos recursos de encapsulamento de código, no contexto de procedimentos ou métodos, em ambiente com memória distribuída. Isso é conseguido com relativo sucesso, mas a abstração completa do ambiente distribuído não é, de fato obtida.

**RPC: Remote Procedure Call**

**RMI: Remote Method Invocation**



# Abstração Incompleta

## Endereçamento

O cliente precisa conhecer o *endereço* do serviço. Isso envolve conhecer o nome do serviço, seu hospedeiro e até a porta com a qual deve ser estabelecida uma conexão.

## Parâmetros por cópia

Não existe um espaço de endereçamento compartilhado. Parâmetros de entrada e resultados são passados por cópia, não havendo a possibilidade de ser por referência.

## Ambiente de rede

Altos tempos de latência, ocorrência de falhas e outras situações típicas em ambiente de rede devem ser toleradas pela aplicação.



# Componentes

## Interface

Especifica o serviço: o nome do procedimento ou classe/método e os parâmetros de entrada e retorno.

## Stub

Proxy no cliente para cumprir as funções de chamada ou invocação.

## Serviço

Implementação do serviço.

## Servidor

Processo responsável por receber invocações e prover o atendimento

## Cliente

Requisita a execução do serviço.

## Skeleton

Requisita a execução do serviço.





# Componentes

## Interface



Especifica o serviço: o nome do procedimento ou classe/método e os parâmetros de entrada e retorno.

## Serviço



Implementação do serviço.

## Cliente



Requisita a execução do serviço.

## Stub



Proxy no cliente para cumprir as funções de chamada ou invocação.

## Servidor

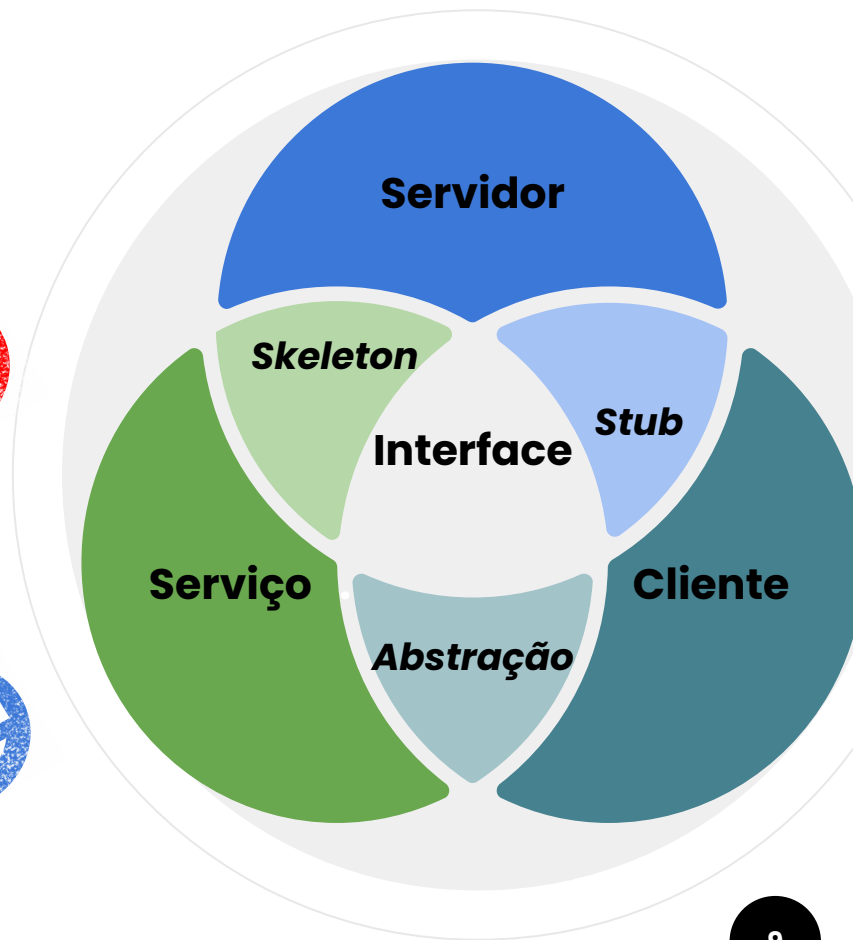


Processo responsável por receber invocações e prover o atendimento

## Skeleton



Requisita a execução do serviço.



# RPC vs. RMI



Possuem características tão próximas que é possível apresentá-los em conjunto. No entanto, suas diferenças são marcantes, identificando vocações de uso, e, portanto, campos de aplicação, completamente diferentes.



# Componentes: RPC

## Interface

Descrita em um arquivo de definição de interface. Identifica os procedimentos oferecidos e os respectivos parâmetros de entrada e retornos.

## Stub

Módulo ligado ao cliente para tratar as comunicações envolvidas.

## Serviço

Módulo contendo a implementação dos procedimentos a serem oferecidos.

## Servidor

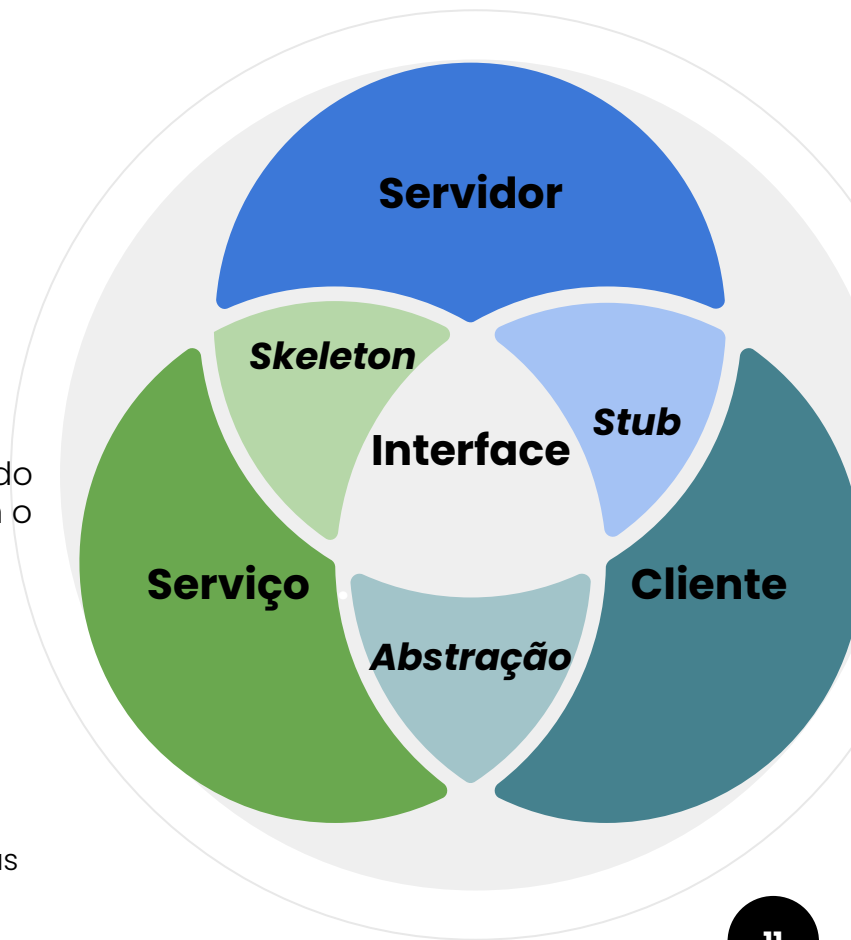
Programa que registra o serviço, recebe e provê tratamento às requisições.

## Cliente

Programa contendo (i) a conexão com o servidor e (ii) a chamada aos serviços.

## Skeleton

Módulo ligado ao cliente para tratar as comunicações envolvidas.

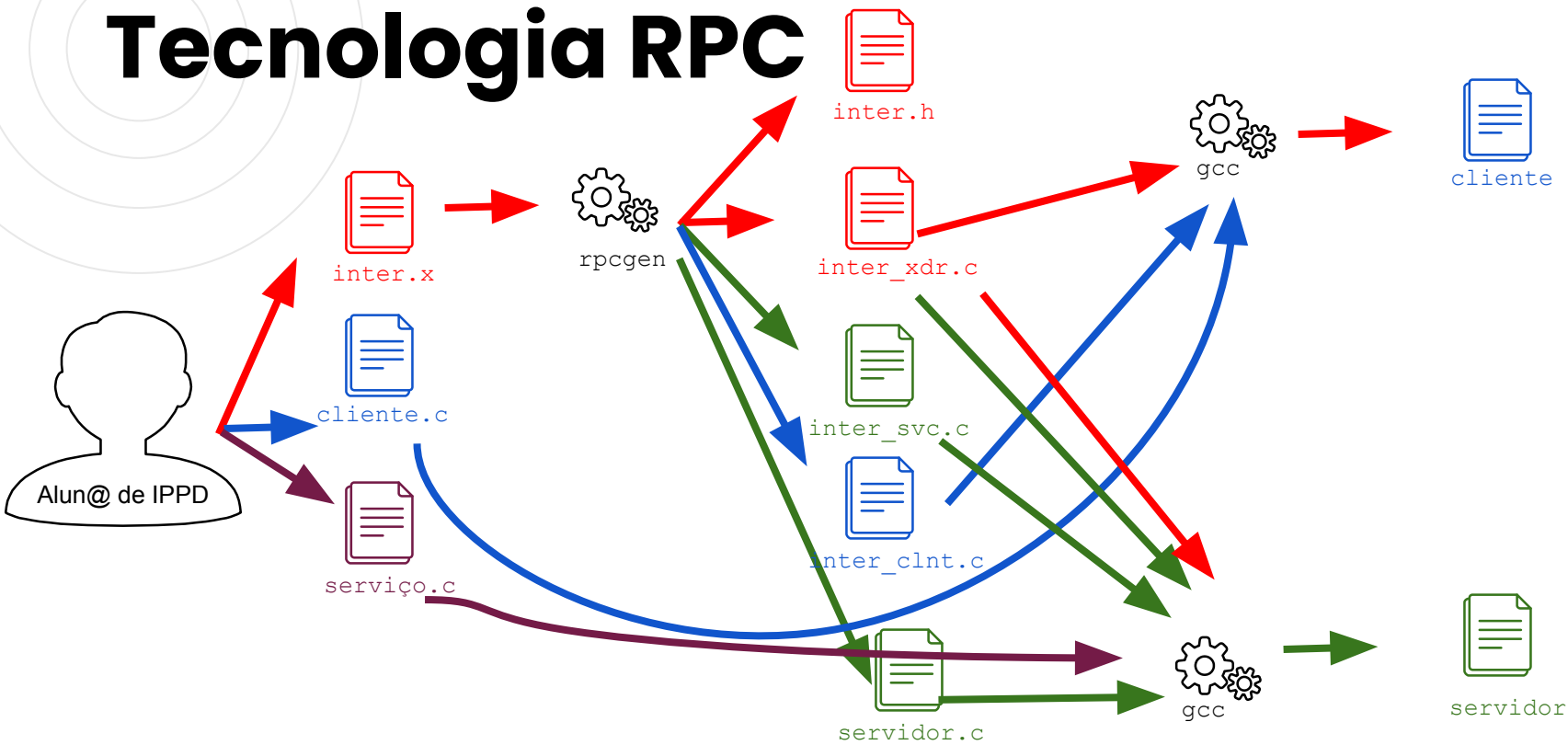


# Tecnologia RPC

- **XDR:**
  - Abstração para compartilhamento de dados, permitindo o empacotamento e desempacotamento dos dados envolvidos na execução do serviço.
- **rpcgen**
  - Gerador, a partir da interface apresentada, dos *stubs* e *skeletons* para uma aplicação.



# Tecnologia RPC



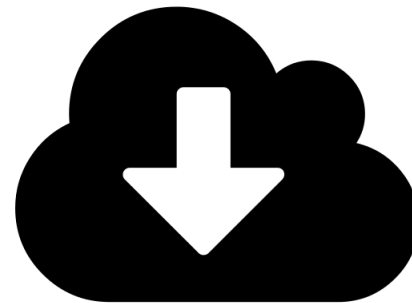
# Tecnologia RPC

**addsub.x**: Interface

**cliente.c**: cliente

**servico.c**: implementação do serviço

```
$> rpcgen addsub.x
$> gcc cliente.c addsub_xdr.c addsub_clnt.c -o cliente
$> gcc servico.c addsub_xdr.c addsub_svc.c -o servidor
$> ./servidor // No terminal 1
$> ./cliente 5 4 // No terminal 2
```



**Download na página  
da disciplina!**

# Componentes: RMI

## Interface

Descrita por uma interface Java. Identifica o tipo abstrato para o serviço.

## Serviço

Classe contendo a implementação da interface oferecida.

## Cliente

Programa contendo (i) a conexão com o servidor e (ii) a chamada aos serviços.

## Stub

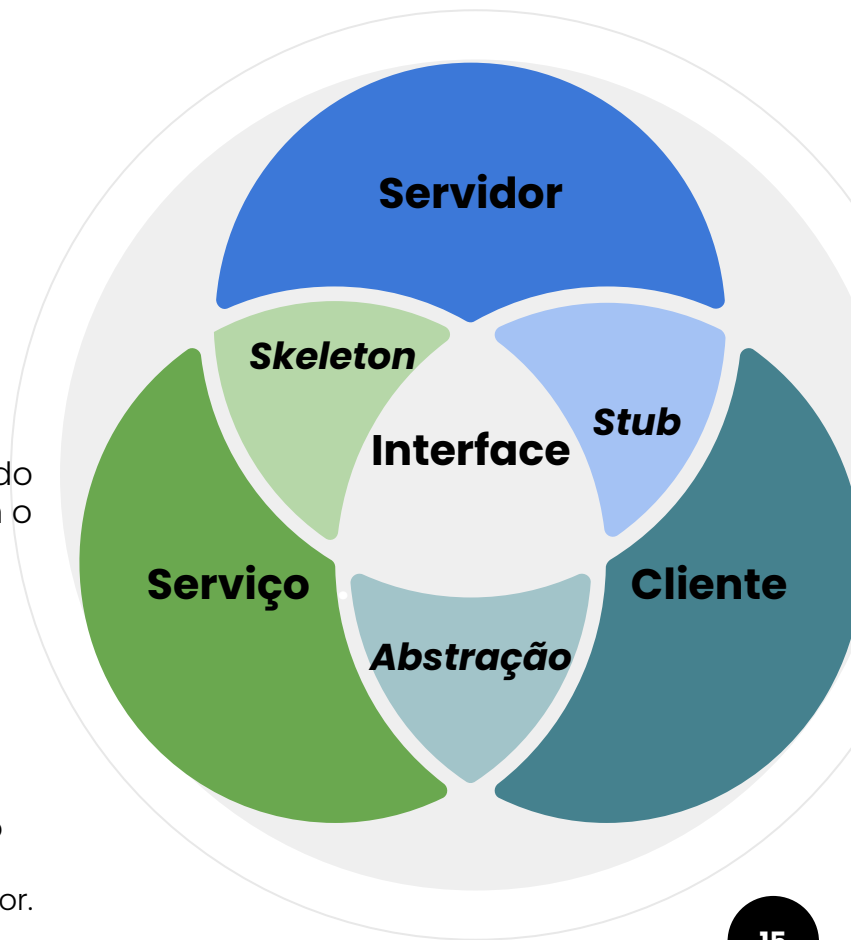
Classe gerada implicitamente pelo compilador Java, utilizada pelo cliente.

## Servidor

Programa que registra o serviço, cria uma instância do serviço, recebe e provê tratamento às requisições.

## Skeleton

Classe gerada implicitamente pelo compilador Java, utilizada pelo servidor.





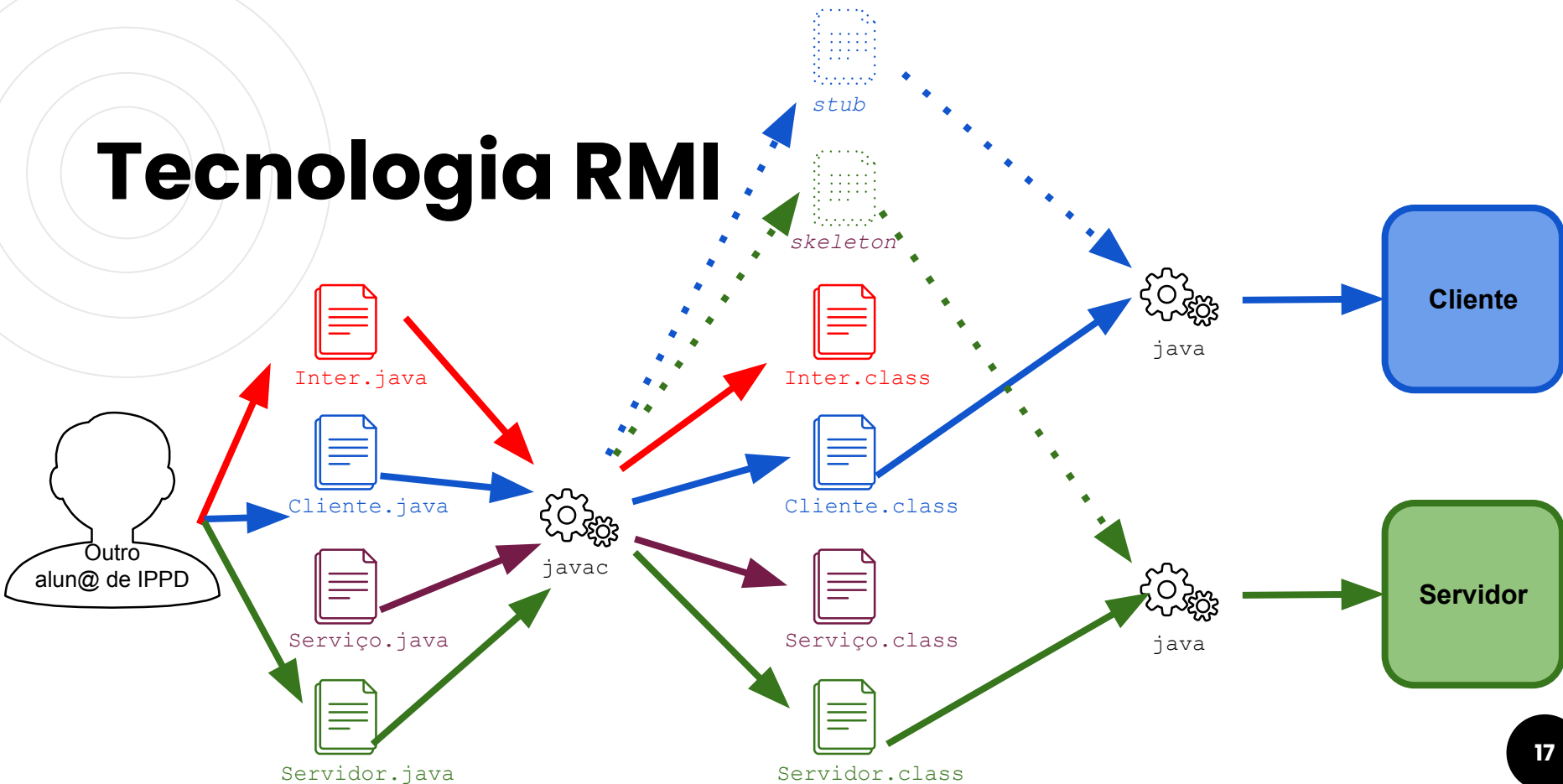
# Tecnologia RMI

- **JVM:**
  - Permite alta portabilidade, além de garantir segurança pela execução em uma máquina virtual.
- **java.RMI.\***
  - Pacote implementando os serviços de suporte necessário.
- **rmiregistry**
  - Serviço de registro de objetos remotos.





# Tecnologia RMI



# Tecnologia RMI

**Fionacci.java:** Interface

**FiboCliente.java:** cliente

**FiboImplementacao.java:** implementação do serviço

**FiboServidor.java:** implementação do servidor

```
$> javac *.java
```

```
$> rmiregistry // No terminal 1, observe estar no mesmo diretório!
```

```
$> java FiboServidor // No terminal 2
```

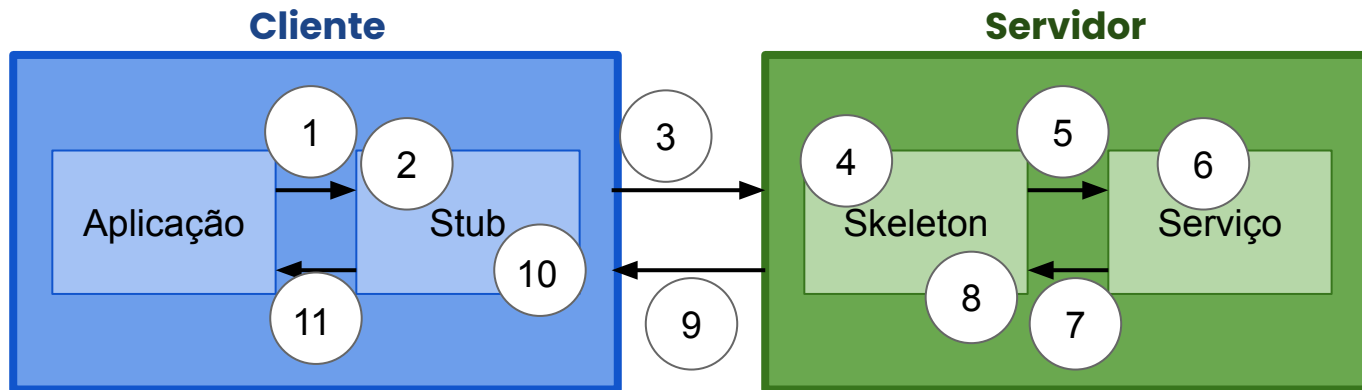
```
$> java FiboCliente 13 // No terminal 3
```



**Download na página  
da disciplina!**

# Tecnologia RPC/RMI

1. Invoca serviço local
2. Monta mensagem
3. Mensagem é enviada
4. Mensagem desempacotada
5. Chamada ao procedimento
6. Executa o procedimento/serviço
7. Retorna o resultado
8. Mensagem é empacotada
9. Mensagem enviada
10. Desempacota o retorno
11. Retorna à aplicação



# Atividade de Acompanhamento

1. Recupere os exemplos de RPC e RMI na página da disciplina e anote suas observações.
2. Modifique a implementação RPC para que ela retorne o Fibonacci de um número.

<https://forms.gle/HhEKfwNKJoeRxcGr5>





**Obrigado!**