



INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Goiás

Instituto Federal de Goiás

Comunicação entre processos (*Sockets*)

Professor: M. Sc. Mario Teixeira Lemes

`mario.lemes@ifg.edu.br`
`https://mariolemes.com.br`

Aula passada x Aula de hoje

Roteiro

Comunicação
entre processos
Datagrama x fluxo
Socket

Datagrama UDP
Características
Aplicações
Comandos Básicos
Resumo da
comunicação

Exemplo 1 -
JSockets (UDP)

Exemplo 2 -
JSockets (TCP)

Exercício em
SALA

Vamos trabalhar?



**INSTITUTO
FEDERAL**
Goiás

Aula passada:

- Formas de posicionar serviços em SDs.
- Modelo Arquitetural.
- Modelo Fundamental.

Aula de hoje:

- Comunicação entre processos.
- Sockets.
- Programação de Sockets com UDP e TCP através do Java (JSockets)

Comunicação entre processos

Roteiro

Comunicação entre processos

Datagrama x fluxo
Socket

Datagrama UDP

Características
Aplicações
Comandos Básicos
Resumo da comunicação

Exemplo 1 - JSockets (UDP)

Exemplo 2 - JSockets (TCP)

Exercício em SALA

Vamos trabalhar?



**INSTITUTO
FEDERAL**
Goiás

O que iremos estudar?

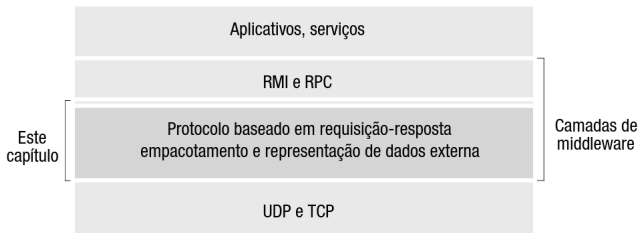


Figura 4.1 Camadas de *middleware*.

Comunicação entre processos

Roteiro

Comunicação
entre processos

Datagrama x fluxo
Socket

Datagrama UDP

Características

Aplicações

Comandos Básicos

Resumo da
comunicação

Exemplo 1 -
JSockets (UDP)

Exemplo 2 -
JSockets (TCP)

Exercício em
SALA

Vamos trabalhar?



INSTITUTO
FEDERAL
Goiás

Sockets:

- Datagrama → Protocolo de transporte **UDP**.
- Fluxo (*stream*) → Protocolo de Transporte **TCP**.

Datagrama:

- É um pacote independente que contém a mensagem que será transferida, **uma única vez**, entre cliente e servidor.

Fluxo (*stream*) :

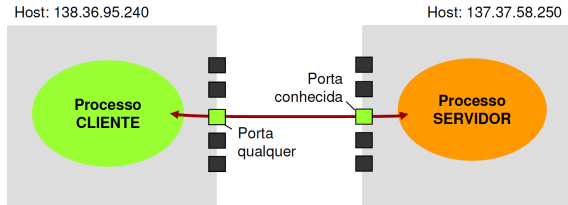
- A informação transmitida consiste em um fluxo contínuo de dados.

Comunicação entre processos

Soceks

Sockets:

- É uma referência indireta a uma **porta** utilizada no processo destino (servidor).
- É um ponto de destino para comunicação entre processos.



Socket = IP + porta

Roteiro

Comunicação entre processos

Datagrama x fluxo
Socket

Datagrama UDP

Características
Aplicações
Comandos Básicos
Resumo da comunicação

Exemplo 1 - JSockets (UDP)

Exemplo 2 - JSockets (TCP)

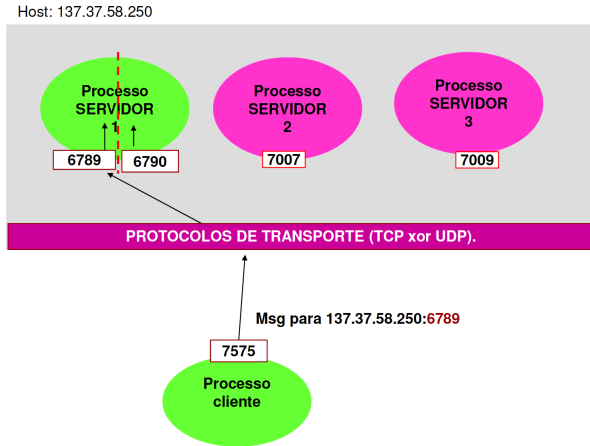
Exercício em SALA

Vamos trabalhar?

Comunicação entre processos

Sockets

Envio de uma mensagem usando sockets:



Roteiro

Comunicação entre processos

Datagrama x fluxo Socket

Datagrama UDP

Características Aplicações Comandos Básicos Resumo da comunicação

Exemplo 1 - JSockets (UDP)

Exemplo 2 - JSockets (TCP)

Exercício em SALA

Vamos trabalhar?

Comunicação entre processos

Sockets

Roteiro

Comunicação
entre processos

Datagrama x fluxo

Socket

Datagrama UDP

Características

Aplicações

Comandos Básicos

Resumo da
comunicação

Exemplo 1 -
JSockets (UDP)

Exemplo 2 -
JSockets (TCP)

Exercício em
SALA

Vamos trabalhar?



INSTITUTO
FEDERAL
Goiás

Características da comunicação por sockets:

- TCP e UDP usam a abstração de sockets.
- Sockets é originário do UNIX BSD, mas está presente na maioria das versões UNIX (Linux, Windows e Macintosh).
- O **mesmo** socket pode enviar e receber mensagens.
- Cada computador possui 2^{16} números de portas disponíveis.
- Cada socket deve ser associado a um protocolo de transporte (UDP ou TCP).

Comunicação por Datagrama UDP

Sockets - UDP

Roteiro

Comunicação
entre processos
Datagrama x fluxo
Socket

Datagrama UDP

Características
Aplicações
Comandos Básicos
Resumo da
comunicação

Exemplo 1 -
JSockets (UDP)

Exemplo 2 -
JSockets (TCP)

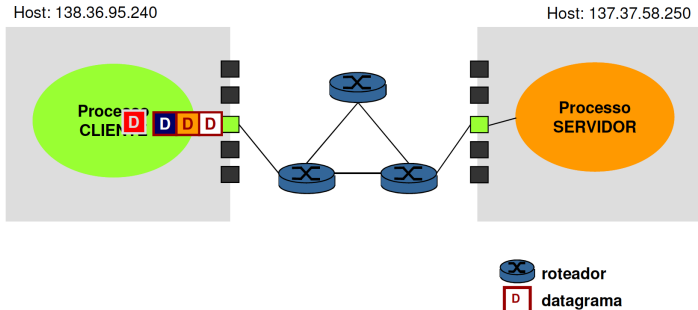
Exercício em
SALA

Vamos trabalhar?



**INSTITUTO
FEDERAL**
Goiás

Envio de uma mensagem usando sockets sobre UDP:



Comunicação por datagrama UDP

Sockets - UDP

Roteiro

Comunicação entre processos

Datagrama x fluxo
Socket

Datagrama UDP

Características

Aplicações
Comandos Básicos
Resumo da
comunicação

Exemplo 1 - JSockets (UDP)

Exemplo 2 - JSockets (TCP)

Exercício em SALA

Vamos trabalhar?



INSTITUTO
FEDERAL
Goiás

- Canal **não** confiável:
 - **Não garante** entrega dos datagramas.
 - Pode entregar datagramas **duplicados**.
 - **Não garante ordem** de entrega dos datagramas.
 - **Não tem estado** de conexão (escuta, estabelecida).

Comunicação por datagrama UDP

Sockets - UDP

Roteiro

Comunicação entre processos

Datagrama x fluxo
Socket

Datagrama UDP

Características

Aplicações

Comandos Básicos

Resumo da comunicação

Exemplo 1 -
JSockets (UDP)

Exemplo 2 -
JSockets (TCP)

Exercício em
SALA

Vamos trabalhar?

- Para algumas aplicações, é **aceitável** usar um serviço que esteja exposto a falhas por omissões ocasionais.



Comunicação por datagrama UDP

Comandos Básicos - Java

Roteiro

Comunicação
entre processos
Datagrama x fluxo
Socket

Datagrama UDP

Características
Aplicações
Comandos Básicos
Resumo da
comunicação

Exemplo 1 -
JSockets (UDP)

Exemplo 2 -
JSockets (TCP)

Exercício em
SALA

Vamos trabalhar?



**INSTITUTO
FEDERAL**
Goiás

■ Criar um socket:

- `DatagramSocket s = new DatagramSocket(6789);`

■ Receber um datagrama:

- `s.receive(req);`

■ Enviar um datagrama:

- `s.send(resp);`

■ Fechar um socket:

- `s.close();`

Comunicação por datagrama UDP

Comandos Básicos - Java

Roteiro

Comunicação entre processos

Datagrama x fluxo
Socket

Datagrama UDP

Características

Aplicações

Comandos Básicos

Resumo da
comunicação

Exemplo 1 - JSockets (UDP)

Exemplo 2 - JSockets (TCP)

Exercício em SALA

Vamos trabalhar?



**INSTITUTO
FEDERAL**
Goiás

- **Montar** um datagrama para **receber** uma mensagem:
 - `new DatagramPacket(buffer, buffer.length);`
- **Montar** um datagrama para **ser enviado**:
 - `new DatagramPacket(msg, msg.length, inet, porta);`

Comunicação por datagrama UDP

Resumo da aplicação cliente-servidor usando UDP

Roteiro

Comunicação
entre processos

Datagrama x fluxo
Socket

Datagrama UDP

Características

Aplicações

Comandos Básicos

Resumo da
comunicação

Exemplo 1 -
JSockets (UDP)

Exemplo 2 -
JSockets (TCP)

Exercício em
SALA

Vamos trabalhar?



◇ Cliente

1. **Criar socket:** um socket pode ser utilizado para enviar datagramas para qualquer socket servidor
2. **Montar datagrama** com
<servidor:porta> de destino
<servidor:porta> de origem
3. **Enviar datagrama**
4. **Bloqueia num receive**
 - ...
 - ...
 - ...
5. **Recebe a resposta**
6. **Trata a resposta**
7. **Volta ao item 2**

◇ Servidor

1. **Aguarda num receive**
 - ...
 - ...
 - ...
 - ...
 - ...
 - ...
 - ...
2. **Recebe a mensagem**
3. **Processa a mensagem**
4. **Envia resposta ao cliente**
5. **Volta ao item 1**

Comunicação por datagrama UDP

Resumo da Comunicação

Roteiro

Comunicação
entre processos
Datagrama x fluxo
Socket

Datagrama UDP

Características
Aplicações
Comandos Básicos
Resumo da
comunicação

Exemplo 1 -
JSockets (UDP)

Exemplo 2 -
JSockets (TCP)

Exercício em
SALA

Vamos trabalhar?



INSTITUTO
FEDERAL
Goiás

Sockets UDP:

- Operação **send** é não-bloqueante.
- Operação **receive** é **bloqueante** (Exceto nos casos em que é especificado um *timeout*).
- Um Socket UDP é **não conectado**:
 - Ao menos se a conexão for manual → `public void connect(InetAddress address, int port).`

Comunicação por datagrama UDP

Exemplo 1 - Servidor: Mensagem Eco

Roteiro

Comunicação entre processos

Datagrama x fluxo
Socket

Datagrama UDP

Características
Aplicações
Comandos Básicos
Resumo da comunicação

Exemplo 1 - JSockets (UDP)

Exemplo 2 - JSockets (TCP)

Exercício em SALA

Vamos trabalhar?



```
public class Servidor {  
    public static void main(String args[]) {  
        DatagramSocket s = null;  
  
        try {  
            s = new DatagramSocket(6789); // cria um socket UDP  
  
            byte[] buffer = new byte[1000];  
  
            System.out.println("\n\n*** Servidor aguardando request");  
  
            // cria datagrama para receber solicitação do cliente  
            DatagramPacket req = new DatagramPacket(buffer, buffer.length);  
  
            s.receive(req);  
  
            System.out.println("*** Request recebido de: " + req.getAddress()+":"+req.getPort());  
  
            // envia resposta  
            DatagramPacket resp = new DatagramPacket(req.getData(), req.getLength(), req.getAddress(), req.getPort());  
            s.send(resp);  
  
        } catch (SocketException e) {  
            System.out.println("Erro de socket: " + e.getMessage());  
        } catch (IOException e) {  
            System.out.println("Erro envio/recepcao pacote: " + e.getMessage());  
        } finally {  
            if (s != null) s.close();  
        }  
    }  
}
```

Comunicação por datagrama UDP

Exemplo 1 - Cliente: Mensagem Eco

```
public class Cliente {
    public static void main(String args[]) {
        DatagramSocket s = null;

        try {
            String servidor = "localhost";
            int porta = 6789;
            String msg = "MENSAGEM TESTE";

            if (args.length > 0) servidor = args[0];
            if (args.length > 1) porta = Integer.parseInt(args[1]);
            if (args.length > 2) msg = args[2];

            byte[] m = msg.getBytes(); // transforma arg em bytes
            InetAddress serv = InetAddress.getByName(servidor);
            s = new DatagramSocket(); // cria um socket UDP

            System.out.println("* Socket criado na porta: " + s.getLocalPort());
            DatagramPacket req = new DatagramPacket(m, msg.length(), serv, porta);
            s.send(req); // envia datagrama contendo a mensagem m
            System.out.println("* Datagrama enviado...: " + msg);

            byte[] buffer = new byte[1000];
            DatagramPacket resp = new DatagramPacket(buffer, buffer.length);
            s.setSoTimeout(10000); // timeout em ms
            s.receive(resp); // aguarda resposta do servidor - bloqueante
            System.out.println("* Resposta do servidor: " + new String(resp.getData(), 0, resp.getLength()));
        } catch (SocketException e) {
            // timeout, erro na criação
            System.out.println("! Erro socket: " + e.getMessage());
        } catch (IOException e) {
            System.out.println("! Erro envio/recepcao do pacote: " + e.getMessage());
        } finally {
            if (s != null) s.close();
        }
    }
}
```

Roteiro

Comunicação
entre processos

Datagrama x fluxo
Socket

Datagrama UDP

Características

Aplicações

Comandos Básicos

Resumo da
comunicação

Exemplo 1 -
JSockets (UDP)

Exemplo 2 -
JSockets (TCP)

Exercício em
SALA

Vamos trabalhar?



Comunicação por *stream* TCP

Exemplo 2 - Servidor: Recebe dados digitados no teclado do cliente

Roteiro

Comunicação entre processos

Datagrama x fluxo
Socket

Datagrama UDP

Características
Aplicações
Comandos Básicos
Resumo da
comunicação

Exemplo 1 - JSockets (UDP)

Exemplo 2 - JSockets (TCP)

Exercício em SALA

Vamos trabalhar?



```
public class Servidor {  
    public static void main(String[] args) throws IOException {  
        ServerSocket servidor = new ServerSocket(12345);  
        System.out.println("Porta 12345 aberta!");  
        Socket cliente = servidor.accept();  
        System.out.println("Nova conexão com o cliente " + cliente.getInetAddress().getHostAddress());  
        Scanner entrada = new Scanner(cliente.getInputStream());  
        while (entrada.hasNextLine()) {  
            System.out.println(entrada.nextLine());  
        }  
        entrada.close();  
        servidor.close();  
    }  
}
```

Comunicação por *stream* TCP

Exemplo 2 - Cliente: Envia dados digitados do teclado

Roteiro

Comunicação entre processos

- Datagrama x fluxo
- Socket

Datagrama UDP

- Características
- Aplicações
- Comandos Básicos
- Resumo da comunicação

Exemplo 1 - JSockets (UDP)

Exemplo 2 - JSockets (TCP)

Exercício em SALA

Vamos trabalhar?



```
public class Cliente {  
    public static void main(String[] args) throws UnknownHostException, IOException {  
        Socket cliente = new Socket("127.0.0.1", 12345);  
        System.out.println("O cliente se conectou ao servidor!");  
        Scanner teclado = new Scanner(System.in);  
        PrintStream saida = new PrintStream(cliente.getOutputStream());  
        while (teclado.hasNextLine()) {  
            saida.println(teclado.nextLine());  
        }  
        saida.close();  
        teclado.close();  
    }  
}
```

AVA - Programação com JSockets

Baseando nos códigos dos slides anteriores, faça:

- **[PROGRAMA 1]** Um servidor que atenda aos clientes invertendo a string recebida (peça para o cliente digitar a mensagem que será invertida) (Alterar código do exemplo 1). - **JSockets sobre UDP**
- **[PROGRAMA 2]** Um servidor que receba o conteúdo de um arquivo do computador do cliente para o servidor, ou seja, faça o cliente ler o conteúdo de um arquivo denominado "arquivo.txt", envie as informações para o servidor e o mesmo grave o conteúdo recebido em um arquivo denominado "recebido.txt". (Alterar código do exemplo 2) - **JSockets sobre TCP**.

Roteiro

Comunicação
entre processos

Datagrama x fluxo
Socket

Datagrama UDP

Características

Aplicações

Comandos Básicos

Resumo da
comunicação

Exemplo 1 -
JSockets (UDP)

Exemplo 2 -
JSockets (TCP)

Exercício em
SALA

Vamos trabalhar?



AVA - Programação com JSockets

Roteiro

Comunicação
entre processos
Datagrama x fluxo
Socket

Datagrama UDP

Características
Aplicações
Comandos Básicos
Resumo da
comunicação

Exemplo 1 - JSockets (UDP)

Exemplo 2 - JSockets (TCP)

Exercício em SALA

Vamos trabalhar?



INSTITUTO
FEDERAL
Goiás

- AVA - Programação distribuída com Sockets (*JSockets*) deve ser entregue, via moodle, até terça-feira (07/10/2025) às 23:59.
- Comente as linhas do seu programa de modo a torná-lo auto-explicativo.
- Semana que vem (07/10/2025) **não haverá aula**. A frequência da disciplina será aferida pela **entrega** da atividade.
- Atividade **individual** ou em **dupla**.

Vamos programar?

Roteiro

Comunicação entre processos

- Datagrama x fluxo
- Socket

Datagrama UDP

- Características
- Aplicações
- Comandos Básicos
- Resumo da
comunicação

Exemplo 1 - JSockets (UDP)

Exemplo 2 - JSockets (TCP)

Exercício em SALA

Vamos trabalhar?



**INSTITUTO
FEDERAL**
Goiás

