

**KELVIN SEVERINO**

**ABNER CARLUCCI**

***TRABALHO 2***

***COMUNICAÇÃO VIA SOCKET JAVA NO LINUX***

**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**5º ADS**

**DISCIPLINA:** Redes de Computadores

*Prof. Me. Jean*

**São José dos Campos – SP**

**2018**

O Conteúdo abaixo é referente ao trabalho 2 – SOCKET da matéria de Redes de Computadores.

Para a realizar deste trabalho, foi necessário a utilização de 2 máquinas virtuais (VM):

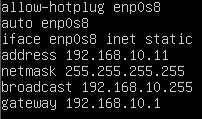
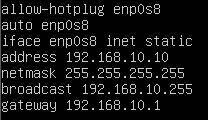
1 – Ubuntu Server

2 – Ubuntu Cliente

**Configuração Inicial de Redes**

Para a realizar a configuração de ambas as máquinas, deve-se primeiro configurar a rede no qual irão funcionar. Para isso deve-se modificar o arquivo Interfaces contido em /etc/network/interfaces

**# vi /etc/network/interfaces** (Comando para acessar o arquivo utilizando o Editor VI)

**Máquina Cliente Máquina Server**

Alguns detalhes para se mencionar, são de que a placa enp0s8 é referente a placa responsável pela REDE INTERNA entre as máquinas, no qual irão se comunicar.

Logo após configurado a rede, deve-se reiniciar o serviço para que as modificações tenham efeito na rede.

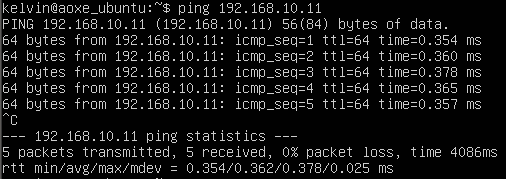
**# service networking restart**

ou

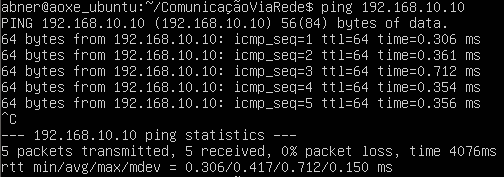
**# /etc/init.d/networking/ restart**

Para testar se ambas as máquinas estão realmente na mesma rede, basta fazer um ping entre as máquinas.

Ping 192.168.10.11 🡪 Comunicação da máquina server com a máquina cliente



Ping 192.168.10.10 🡪 Comunicação da máquina cliente com a máquina server



**Código Java do Socket**

Para que possamos realizar a comunicação via socket, precisamos elaborar um código em java que realize a comunicação entra as máquinas e envia uma mensagem. Segue abaixo o código em JAVA de ambas as máquinas:

**Máquina Server**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.ServerSocket;

import java.net.Socket;

public class Servidor

{

public static void main(String args[]) throws Exception

{

//Abaixo o código que ativa a Classe Cliente

//Cliente c = new Cliente();

//c.conectarEnviar();

//Abaixo o código que ativa a Classe Servidor

Servidor server = new Servidor();

server.iniciar();

}

private ServerSocket socketServidor;

public Servidor() throws IOException

{

System.out.println("Startando Servidor");

socketServidor = new ServerSocket(1234); //Nesta linha é inserido a porta, EX:1234

}

public void iniciar() throws Exception

{

while(true)

{

Socket socketEscuta = socketServidor.accept();

InputStreamReader streamReader =

new InputStreamReader(socketEscuta.getInputStream());

BufferedReader reader = new BufferedReader(streamReader);

String textoEnviado = reader.readLine();

System.out.println(textoEnviado);

reader.close();

}

}

}

**Máquina Cliente**

import java.io.IOException;

import java.io.PrintWriter;

import java.net.Socket;

public class Cliente {

public static void main(String args[]) throws Exception {

//Abaixo o codigo que ativa a Classe Cliente

Cliente c = new Cliente();

c.conectarEnviar();

}

private Socket socketCliente;

public Cliente() throws Exception

{

try {

System.out.println("Realizando Conexão");

//Abaixo se insere o IP do servidor e a porta que ira mandar a mensagem

socketCliente = new Socket("192.168.10.10", 1234);

}

catch(IOException e) {

System.out.println(e);

}

}

public void conectarEnviar() throws Exception

{

PrintWriter escritor =

new PrintWriter(socketCliente.getOutputStream());

System.out.println("Enviando...");

escritor.write("MENSAGEM VIA SOCKET!!!");

escritor.close();

}

}

Deve-se jogar os códigos em suas respectivas máquinas de nome. Logo após será necessário inserir o comando **javac** para criar a classe java e ser compilado posteriormente, e após a criação da classe, para que se possa compilar, apenas mande o comando **java**.

**Máquina Server**

**# javac /caminho/do/diretório/Servidor.java**

**# java /caminho/do/diretório/Servidor**

**Máquina Cliente**

**# javac /caminho/do/diretório/Cliente.java**

**# java /caminho/do/diretório/Cliente**