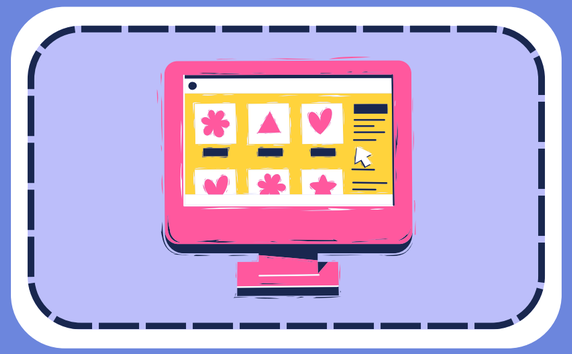


SOCKET

Sistemas Distribuídos

SOCKET

Definição dos Sockets

- Os sockets possibilitam a comunicação entre dois processos, permitindo a troca de informações. Na internet, por exemplo, são usados para estabelecer conexões entre usuários e sites.

Popularidade e Uso Diário

- O uso de sockets é frequente devido à sua aplicação comum em páginas da web, contribuindo para a melhoria da experiência do usuário.

APLICAÇÕES DIVERSAS

Além das páginas da web, os sockets são utilizados em configurações como a criação de conexões SSH em servidores, aumentando a eficiência entre processos.

Origens Históricas

Os sockets remontam a 1983, quando foram introduzidos na versão 4.2 do Berkeley Software Distribution (BSD), derivado do Unix. Desde então, são uma característica proeminente em sistemas Unix-like, como as distribuições Linux.

SOCKET

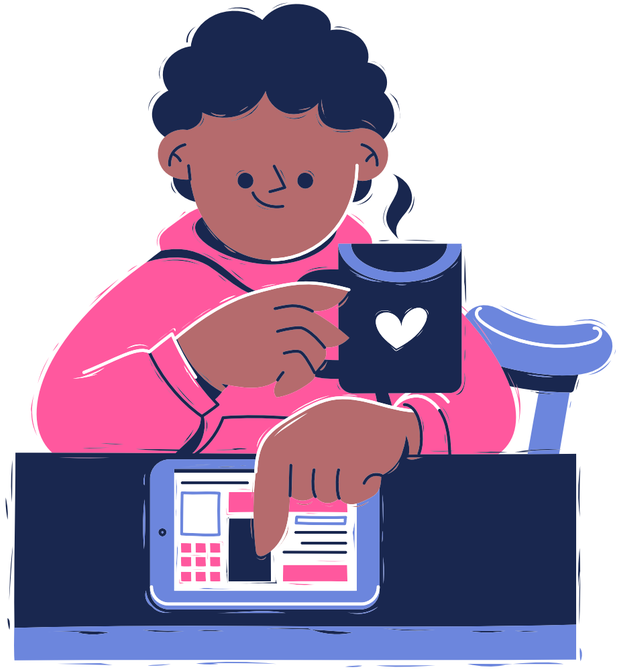
EVOLUÇÃO PARA APIS

- Atualmente, os sockets são frequentemente encontrados na forma de APIs, proporcionando maior independência em relação à rede

Centralização da Comunicação

- A interface entre os processos ocorre em portas distintas, centralizando o projeto e limitando a comunicação entre processos diferentes, embora não seja completamente independente devido à necessidade de transporte pelo protocolo(TCP/UDP).

implementação



Nessa etapa foi implementado na linguagem java o socket entre cliente e servidor.





Imports Servidor

```
package chatserversocket;

import java.awt.Image;
import java.awt.Toolkit;
import java.net.URL;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.ImageIcon;
```

CODIGO SERVER

```
public class chat_server extends javax.swing.JFrame {
    static ServerSocket ss;
    static Socket s;
    static DataInputStream din;
    static DataOutputStream dout;

    public chat_server() {
        initComponents();

      try
          {
        Image i = new ImageIcon(this.getClass().getResource("
/servers.png")).getImage();
        this.setIconImage(i);
       }catch(Exception ex){
           System.out.println("Sem imagem");
           System.out.println(ex.toString());
           }


       }

    private void msg_sendActionPerformed(java.awt.event.
ActionEvent evt) {
         try {
            String msgout = "";
            msgout = msg_text.getText().trim();
            dout.writeUTF(msgout);
//Entrega a mensagem do sever ao cliente.

        } catch (IOException ex) {
             }

                }
```

CODIGO SERVER

```
private void SeverActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
                                                }
     @param args the command line arguments

   public static void main(String args[]) {
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new chat_server().setVisible(true);
                   }
        });

      String msgin = "";
      try{
          ss = new ServerSocket(1201); //Porta de número 1201
          s = ss.accept(); //agora o servidor aceitará a conexão
          din  = new DataInputStream(s.getInputStream());
          dout = new DataOutputStream(s.getOutputStream());


          while(!msgin.equals("exit")){
              msgin  = din.readUTF();
              msg_area.setText(msg_area.getText().trim()+"\nCliente: "+msgin);
// display onde aparece a mensagem do cliente para o servidos
                 }
      }catch(Exception e){

          }


       }
    // Variables declaration - do not modify
    private javax.swing.JTextField Sever;
    private javax.swing.JScrollPane jScrollPane1;
    private static javax.swing.JTextArea msg_area;
    private javax.swing.JToggleButton msg_send;
    private javax.swing.JTextField msg_text;
    // End of variables declaration
}
```





Imports Cliente

```
package chatclientsocket;

import java.awt.Image;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
```

CODIGO CLIENTE

```
public class chat_client extends javax.swing.JFrame {
    static Socket s;
    static DataInputStream din;
    static DataOutputStream dout;
    static String numeroIP ;
    static String porta ;
    static int portaInt;

    public chat_client() {
        initComponents();
         try
          {
        Image i = new ImageIcon(this.getClass().getResource("/client.png
")).getImage();
        this.setIconImage(i);
       }catch(Exception ex){
           System.out.println("Sem imagem");
           System.out.println(ex.toString());
           }
       }
    private void msg_sendActionPerformed(java.awt.event.ActionEvent evt
) {
        try {
            String msgout = "";
            msgout = msg_text.getText().trim();
            dout.writeUTF(msgout);
//Entrega a mensagem do cliente ao sever.
        } catch (IOException ex) {
             }
```

CODIGO CLIENTE

```
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {

            }
    });
    numeroIP = JOptionPane.showInputDialog( " Digite o número de IP " );;

    porta = JOptionPane.showInputDialog( " Digite o número de porta " );;
    portaInt = Integer.parseInt(porta);

    System.out.println(numeroIP);

    System.out.println(portaInt);
    new chat_client().setVisible(true);

   try{

      s = new Socket(numeroIP, portaInt);
//Isso é o endereço IP local addr. Devido ao motivo de rodarmos cliente e servidor no mesmo
computador.

      din = new DataInputStream(s.getInputStream());
      dout = new DataOutputStream(s.getOutputStream());
      String msgin = "";

      while(!msgin.equals("exit")){
          msgin  = din.readUTF();
          msg_area.setText(msg_area.getText().trim()+"\nSever: "+msgin);
// display onde aparece a mensagem do cliente para o servidos
          }

  }catch(Exception e){

      }
 }
// Variables declaration - do not modify
private javax.swing.JTextField Client;
private javax.swing.JScrollPane jScrollPane1;
private static javax.swing.JTextArea msg_area;
private javax.swing.JToggleButton msg_send;
private javax.swing.JTextField msg_text;
```

SOCKETS

Vantagens

- Facilidade de uso
- Portabilidade
- Ampla adoção
- Flexibilidade

Desvantagens

- Baixo nível de abstração
- Gerenciamento manual de conexões
- Segurança
- Escalabilidade limitada





Funcionamento ServerSocket

Construtor

Cria um socket de servidor, vinculado à porta especificada ou alocada automaticamente se colocado o valor 0. Este número de porta pode então ser obtido chamando getLocalPort.

SocketImplFactory

Analisando a documentação do java.net.ServerSocket, nota-se a implementação do padrão de projeto Factory, adicionando uma camada de segurança ao projeto.

SocketImpl

A classe ServerSocket gera na verdade uma instância de SocketImpl, que por sua vez é instanciada na Factory.





Funcionamento Socket

Construtor

No nosso exemplo, especificamos IP e porta do servidor a ser conectado ao construir o Socket do cliente, mas pode-se utilizar proxy, um socket já criado, entre outras configurações.

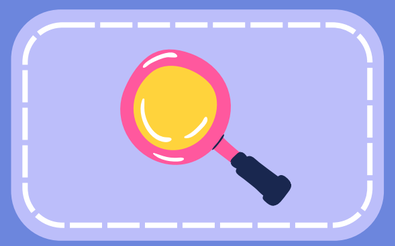
Input e Output

Utiliza de métodos getInputStream() e getOutputStream() para receber as mensagens do server.

java.lang.Object

Nota-se que é possível utilizar métodos comparativos da classe Object, como, equals, wait ou notify, já que o Socket herda tais características.




Exemplo de empresas que utilizam socket

References

- https://pt.linkedin.com/pulse/como-os-sockets-revolucionaram-comunica%C3%A7%C3%A3o-entre-matheus-almeida#:~:text=Hist%C3%B3ria%20dos%20Sockets,-A%20concep%C3%A7%C3%A3o%20de&text=J%C3%A1%20existiam%20movimenta%C3%A7%C3%B5es%20nesta%20parte,estabeleceu%20padr%C3%B5es%20para%20esta%20comunica%C3%A7%C3%A3o.
- https://www.youtube.com/watch?v=kqBmsLvWU14&pp=ygUOc29ja2V0IGVtIGphdmE%3D
- https://docs.oracle.com/javase/8/docs/api/java/net/Socket.html
- https://docs.oracle.com/javase/8/docs/api/java/net/ServerSocket.html

Obrigado!