

GUIA #7

SOCKETS

TALLER DE PROGRAMACIÓN
OLGA LUCÍA ROA BOHÓRQUEZ

KAROL STEFANY ORDOÑEZ PEÑALOZA
YEFERSON PIÑEROS PEÑA

22/10/2024

PREGUNTAS ORIENTADORAS

¿Cuáles fueron los aprendizajes obtenidos al realizar esta guía?, liste como mínimo 3 aprendizajes y relaciónelos con su futuro quehacer profesional.

- Aprender a manejar los socket y su funcionalidad que nos permite conectar dos máquinas de diferentes “ip” para hacer un chat en tiempo real.
- La implementación de un chat en tiempo real para el proyecto puede ser aplicado a proyectos mucho más grandes y tendrá el mismo resultado de eficacia.
- Simular estos escenarios problemáticos o desafiantes de la vida real y darles resolución con los temas vistos en clase nos prepara para afrontar problemas de resolución lógica en diferentes áreas de trabajo del software que requieren modelados de estos escenarios.

¿Dónde presentó mayor dificultad resolviendo la guía? y ¿cómo lo resolvieron? ¿Cuáles fueron las estrategias de solución?

- Se presentó dificultad en la comprensión del uso de los “threads” y el manejo de los sockets, pero se pudo solucionar buscando información en diferentes fuentes y códigos que los implementan para tener una mayor claridad de su uso.

ACTIVIDAD DE TRABAJO AUTÓNOMO

What is distributed computing?

- “Distributed computing is a method of having multiple computers work together to solve a common problem. In this way, a network of computers forms a single powerful computer that provides large-scale resources to tackle complex challenges.”(aws. P1. P2024).

These systems provide many advantages over single-system computing. For example: scalability, transparency, consistency, availability, efficiency.

What are Sockets used for?

- Sockets are used to establish a connection between two devices over a network, allowing bi-directional communication. They are programming interfaces that allow

an application to send and receive data, either on a local network or on the Internet. For example, a Socket allows a web server and a browser to communicate, where the server sends the requested page and the browser receives it. In network application development, Sockets are essential for real-time data transmission.

What is the difference between UDP and TCP?

- TCP: Transmission Control Protocol is a connection-oriented protocol that guarantees delivery of data in the correct order. It ensures that all packets sent arrive at the destination without errors, so it is more reliable. TCP is used in applications where data integrity is critical, such as the web and email.
- UDP: User Datagram Protocol is a connectionless protocol and does not guarantee delivery of packets or the order in which they arrive. UDP is faster than TCP because it does not need to verify whether data arrived correctly or in order. It is used in applications where speed is more important than reliability, such as live video streaming or online gaming.

“So the main difference between them is that TCP is a connection-based protocol and UDP is connectionless. Although TCP is more reliable, it transfers data more slowly. UDP is less reliable but works faster.”(Academy. P1. 2024)

What is RMI and JNDI? And how do they relate to Sockets?

- RMI: Remote Method Invocation is a Java technology that allows an application to invoke methods of objects on other machines. With RMI, developers can build distributed applications where method calls can occur on different servers. RMI uses Sockets to establish network communication between systems.
- JNDI Java Naming and Directory Interface: Java Naming and Directory Interface This is an API that provides a way to look up and access network services or resources using names. JNDI is useful for accessing distributed services, such as databases or application services. Although JNDI does not use Sockets directly, it relies on them to establish network connections to distributed services.

Both RMI and JNDI rely on Sockets for information exchange and communication between different systems on a network.

What is a Web Service?

- A Web Service is an application that allows communication and data exchange between different applications over a network, such as the Internet. Web services are designed to be independent of the programming language and can be consumed by various applications and systems. They use standard protocols such as HTTP and formats such as XML or JSON, allowing interoperability between different platforms and devices. Web services are fundamental in application integration and are a key foundation for software architectures such as REST and SOAP.

“In simple terms, a web server is a computer that stores, processes, and delivers website files to users from a browser.”(Hostinger tutoriales. P1. 2024)

```
1 <!DOCTYPE html>
2 <!--
3 Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
4 Click nbfs://nbhost/SystemFileSystem/Templates/JSP\_Servlet/Html.html to edit this template
5 -->
6 <!DOCTYPE html>
7 <html lang="es">
8 <head>
9     <meta charset="UTF-8">
10    <meta name="viewport" content="width=device-width, initial-scale=1.0">
11    <title>Magic Travel - Soporte</title>
12    <style>
13        body {
14            font-family: 'Courier New', Courier, monospace;
15            display: flex;
16            justify-content: center;
17            align-items: center;
18            height: 100vh;
19            margin: 0;
20            background-color: #f4f4f4;
21        }
22        .container {
23            background-color: #fff;
24            padding: 10px;
25            border-radius: 10px;
26            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
27            width: 300px;
28            height: 500px;
29            display: flex;
30            flex-direction: column;
31            justify-content: space-between;
32        }
33        .header {
34            display: flex;
35            align-items: center;
36            justify-content: center;
37            padding: 10px;
38        }
39        .header img {
40            width: 70px;
41            margin-right: 20px;
42        }
43        .header h1 {
44            font-size: 20px;
45            margin: 0;
46        }
47        hr {
48            border: 0;
```

```
48     border: 0;
49     height: 1px;
50     background: #ccc;
51     margin: 0 0 10px 0;
52 }
53 .chat-window {
54     flex-grow: 1;
55     background-color: #f9f9f9;
56     border: 1px solid #ccc;
57     border-radius: 5px;
58     padding: 10px;
59     overflow-y: auto;
60 }
61 .input-container {
62     display: flex;
63     align-items: center;
64     padding: 10px;
65     border-top: 1px solid #ccc;
66 }
67 .input-container input[type="text"] {
68     flex-grow: 1;
69     padding: 10px;
70     border: 1px solid #ccc;
71     border-radius: 5px;
72     font-size: 16px;
73 }
74 .input-container button {
75     background-color: #007bff;
76     border: none;
77     color: #fff;
78     padding: 10px 15px;
79     margin-left: 10px;
80     border-radius: 5px;
81     cursor: pointer;
82 }
83 .input-container button:hover {
84     background-color: #0056b3;
85 }
86 .message {
87     margin: 5px 0;
88     padding: 8px;
89     background-color: #elf5fe;
90     border-radius: 5px;
91 }
92 .app-message {
93     margin: 5px 0;
94     padding: 8px;
95     background-color: #dcedc8;
96     border-radius: 5px;
```

```

97 |     }
98 |   </style>
99 | </head>
100 | <body>
101 |   <div class="container" id="chatContainer">
102 |     <div class="header">
103 |       
104 |       <h1>Soporte</h1>
105 |     </div>
106 |     <hr>
107 |     <div class="chat-window" id="chatWindow">
108 |
109 |     </div>
110 |     <div class="input-container">
111 |       <input type="text" id="messageInput" placeholder="Enviar un mensaje">
112 |       <button type="button" onclick="sendMessage()">>></button>
113 |     </div>
114 |   </div>
115 |
116 | <script>
117 |   let state = "mainMenu"; // Estado inicial
118 |   let solicitudRadicado = null;
119 |   let quejaRadicado = null;
120 |   let reclamoRadicado = null;
121 |
122 |   function sendMessage() {
123 |     const messageInput = document.getElementById('messageInput');
124 |     const messageText = messageInput.value.trim();
125 |     messageInput.value = ""; // Limpiar el campo de entrada
126 |
127 |     if (messageText.toLowerCase() === "quit") {
128 |       document.getElementById('chatWindow').innerHTML = '';
129 |       return;
130 |     }
131 |
132 |     if (messageText !== "") {
133 |       displayMessage("Tú: " + messageText, 'user');
134 |       handleUserInput(messageText);
135 |     }
136 |   }
137 |
138 |   function handleUserInput(input) {
139 |     switch (state) {
140 |       case "mainMenu":
141 |         displayMessage("MagicTravel: Hola!! Bienvenido a soporte. ¿Cómo puedo ayudarte el día de hoy?<br>1. Solicitud<br>2. Queja<br>3. Reclamo<br>4. Información", 'app');
142 |         state = "menuOptions";
143 |         displayMessage("Escribe quit para culminar", 'app');
144 |         break;

```

```

145 |
146 |       case "menuOptions":
147 |         processMenuOption(input);
148 |         break;
149 |
150 |       case "solicitud":
151 |         if (input) {
152 |           solicitudRadicado = Math.floor(1000 + Math.random() * 9000);
153 |           displayMessage("MagicTravel: Entiendo. Tu solicitud será radicada con número ${solicitudRadicado} y será atendida lo antes posible.<br>Necesitas algo más?<br>1. Nueva solicitud<br>2. Volver al menú anterior", 'app');
154 |           state = "solicitudSubMenu";
155 |         }
156 |         break;
157 |
158 |       case "queja":
159 |         if (input) {
160 |           quejaRadicado = Math.floor(1000 + Math.random() * 9000);
161 |           displayMessage("MagicTravel: Tu queja será radicada con número ${quejaRadicado} y será atendida lo antes posible.<br>Necesitas algo más?<br>1. Nueva queja<br>2. Volver al menú anterior", 'app');
162 |           state = "quejaSubMenu";
163 |         }
164 |         break;
165 |
166 |       case "reclamo":
167 |         if (input) {
168 |           reclamoRadicado = Math.floor(1000 + Math.random() * 9000);
169 |           displayMessage("MagicTravel: Tu reclamo será radicado con número ${reclamoRadicado} y será atendido lo antes posible.<br>Necesitas algo más?<br>1. Nuevo reclamo<br>2. Volver al menú anterior", 'app');
170 |           state = "reclamoSubMenu";
171 |         }
172 |         break;
173 |
174 |       case "informacion":
175 |         displayMessage("MagicTravel: Claro, dime qué quieres conocer:<br>1. Planes de viaje<br>2. Beneficios<br>3. Ser miembro<br>0. Volver al menú anterior", 'app');
176 |         state = "informacionSubMenu";
177 |         break;
178 |
179 |       case "informacionSubMenu":
180 |         processInformacionOption(input);
181 |         break;
182 |
183 |       case "informacionPlanes":
184 |       case "informacionBeneficios":
185 |       case "informacionMiembro":
186 |         displayMessage("MagicTravel: Muy bien aquí tienes información sobre ${state === "informacionPlanes" ? "Planes de viaje" : state === "informacionBeneficios" ? "Beneficios" : "Ser miembro"}", 'app');
187 |         state = "informacionDetail";
188 |         break;
189 |
190 |       case "informacionDetail":
191 |         if (input === "0") {
192 |           state = "informacion";
193 |           displayMessage("Volviste al menú principal", 'app');

```

```

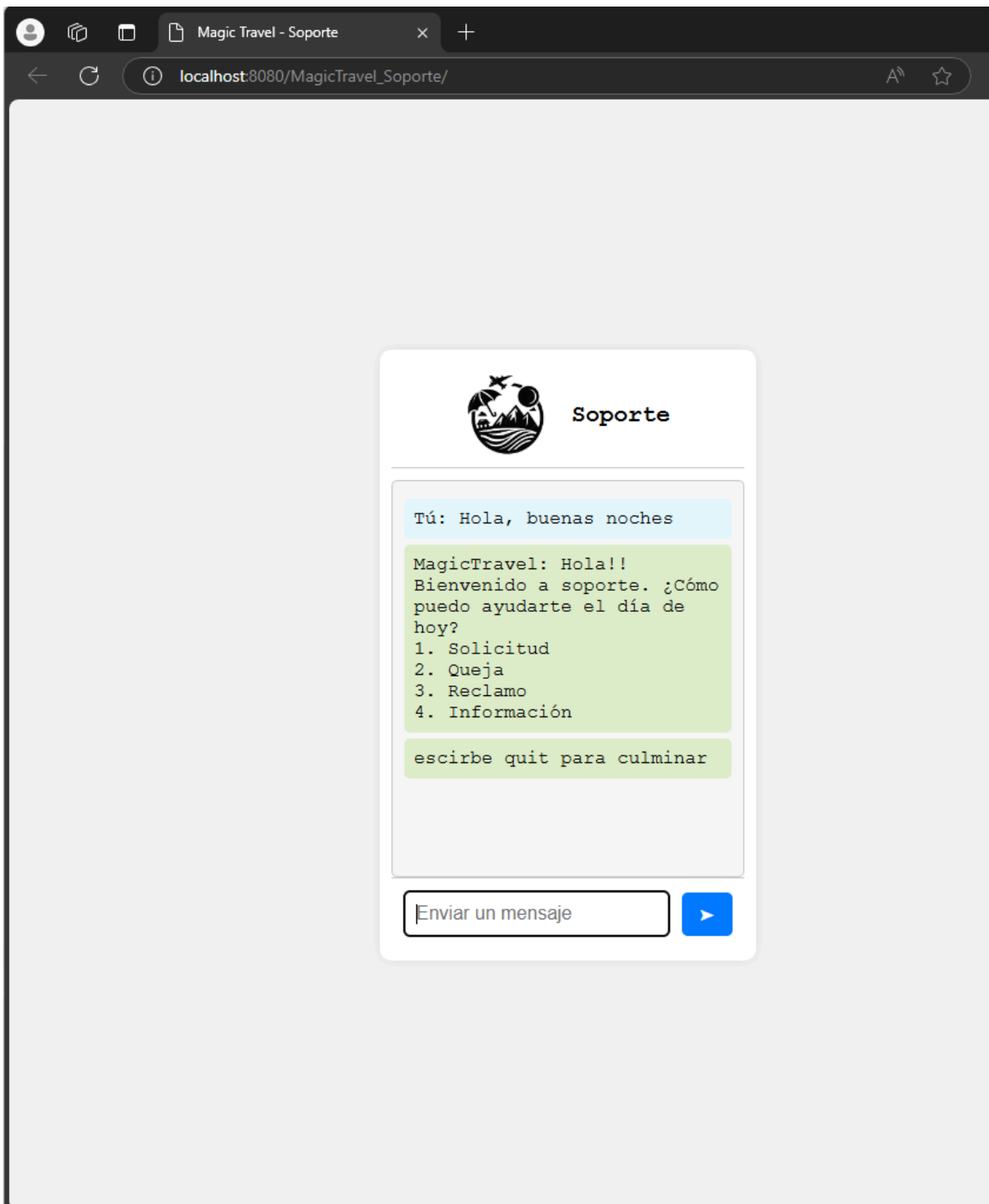
192         state = "informacion";
193         handleUserInput("");
194     }
195     break;
196
197     case "solicitudSubMenu":
198     case "quejaSubMenu":
199     case "reclamoSubMenu":
200         processSubMenuOption(input);
201         break;
202     }
203 }
204
205 function processMenuOption(input) {
206     if (input === "1") {
207         state = "solicitud";
208         displayMessage("MagicTravel: Dime exactamente qué necesitas", 'app');
209     } else if (input === "2") {
210         state = "queja";
211         displayMessage("MagicTravel: Por favor, escribe claramente tu inconformidad", 'app');
212     } else if (input === "3") {
213         state = "reclamo";
214         displayMessage("MagicTravel: Dime exactamente cuál es tu caso (recuerda conocer nuestras políticas y condiciones para tener más conocimiento de tu caso)", 'app');
215     } else if (input === "4") {
216         state = "informacion";
217         handleUserInput("");
218     }
219 }
220
221 }
222
223 function processInformationOption(input) {
224     if (input === "1") {
225         state = "informacionPlanes";
226         handleUserInput("");
227     } else if (input === "2") {
228         state = "informacionBeneficios";
229         handleUserInput("");
230     } else if (input === "3") {
231         state = "informacionMiembro";
232         handleUserInput("");
233     } else if (input === "4") {
234         state = "mainMenu";
235         handleUserInput("");
236     }
237 }
238
239 function processSubMenuOption(input) {

```

```

239     function processSubMenuOption(input) {
240         if (input === "1") {
241             if (state === "solicitudSubMenu") {
242                 state = "solicitud";
243                 displayMessage("MagicTravel: Dime exactamente qué necesitas", 'app');
244             } else if (state === "quejaSubMenu") {
245                 state = "queja";
246                 displayMessage("MagicTravel: Por favor, escribe claramente tu inconformidad", 'app');
247             } else if (state === "reclamoSubMenu") {
248                 state = "reclamo";
249                 displayMessage("MagicTravel: Dime exactamente cuál es tu caso", 'app');
250             }
251         } else if (input === "2" || input === "0") {
252             state = "mainMenu";
253             handleUserInput("");
254         }
255     }
256
257     function displayMessage(text, sender) {
258         const messageElement = document.createElement('div');
259         messageElement.classList.add(sender === 'user' ? 'message' : 'app-message');
260         messageElement.innerHTML = text;
261         const chatWindow = document.getElementById('chatWindow');
262         chatWindow.appendChild(messageElement);
263         chatWindow.scrollTop = chatWindow.scrollHeight;
264     }
265
266     document.getElementById('messageInput').addEventListener('keypress', function (e) {
267         if (e.key === 'Enter') {
268             sendMessage();
269         }
270     });
271 </script>
272 </body>
273 </html>

```

DIALOGO DEL VIDEO:

Good afternoon, today I'm going to talk to you a little about the answers to the questions in the document about networks and distributed systems.

First, distributed computing. Basically, it's a method in which several computers collaborate to solve complex problems, as if they were a single powerful computer. This offers advantages such as greater scalability, availability and efficiency, something essential in modern applications.

To achieve this communication, sockets are used. A socket allows two devices to connect and send data in real time, something key for network applications, such as when a web server sends a page to a browser.

It's also important to know the difference between TCP and UDP. TCP is a connection protocol that guarantees the delivery of data in order and without errors, ideal for applications such as email or web browsing. UDP, on the other hand, is connectionless, which makes it faster but less reliable, and is used in live broadcasts or online games.

On the other hand, the document talks about how RMI allows a Java application to execute methods on other machines, while JNDI allows access to network services, such as databases. Both rely on sockets to communicate.

Finally, there are web services, which are applications designed to enable communication between different systems over a network. They are language-agnostic, allowing them to be used on a variety of platforms using protocols such as HTTP and formats such as XML or JSON, and are essential for architectures such as REST and SOAP.

LINK DEL VIDEO: 📺 WhatsApp Video 2024-11-10 at 9.54.48 PM.mp4

<https://drive.google.com/file/d/122-4Pyop1b2YruAs72rcWydP1Mhubcsd/view?usp=sharing>

BIBLIOGRAFÍA

1. <https://aws.amazon.com/es/what-is/distributed-computing/> - ¿Qué es la computación distribuida?. aws. 2024
2. <https://www.avast.com/es-es/c-tcp-vs-udp-difference#:~:text=La%20principal%20diferencia%20entre%20el,fiable%20pero%20funciona%20m%C3%A1s%20r%C3%A1pido.> - TCP frente a UDP: ¿En qué se diferencian y qué protocolo es mejor. Academy. 2024
3. <https://www.hostinger.co/tutoriales/que-es-un-servidor-web> - ¿Qué es un servidor web y cómo funciona?. Hostinger tutoriales. 2024