**CRIAÇÃO DE UM CHAT UTILIZANDO O SOCKET IO**

**CONFIGURAÇÃO DO PROJETO**

1. Vamos utilizar a mesma base do código da sessão 25 e dar prosseguimento para a criação do chat.
2. No frontend no arquivo index.ejs, fazer as alterações como abaixo.
3. <!DOCTYPE html>
4. <html lang="en">
5. <head>
6. <meta charset="UTF-8">
7. <meta http-equiv="X-UA-Compatible" content="IE=edge">
8. <meta name="viewport" content="width=device-width, initial-scale=1.0">
9. <title>Document</title>
10. </head>
11. <body>
12. <h1> Chat do Guia </h1>
13. <div id="chat">
14. </div>
15. <input type="text" name="" id="msg"><br>
16. <input type="text" name="" id="username" placeholder="Seu nome de usuário">
17. <button>Enviar</button>
19. </body>
20. <script src="https://cdn.socket.io/4.4.1/socket.io.min.js"
21. integrity="sha384-fKnu0iswBIqkjxrhQCTZ7qlLHOFEgNkRmK2vaO/LbTZSXdJfAu6ewRBdwHPhBo/H"
22. crossorigin="anonymous">
23. </script>
24. <script>
25. *var* clienteFrontEnd = io("http://localhost:3000");
26. clienteFrontEnd.on("disconnect", ()*=>*{
27. console.log("Desconectado")
28. })
29. </script>
30. </html>
31. No backend faremos as alterações como abaixo;
32. *var* express = require("express");
33. *var* app = express();
34. //colocando a aplicação express para rodar dentro do servidor http nativo do node
35. *var* http = require("http").createServer(app);
36. *var* io = require("socket.io")(http);
37. //evento de conexão do socket.io
38. io.on("connection",(*clienteBackEnd*) *=>* {
39. *clienteBackEnd*.on("disconnect",() *=>* {
40. console.log("Cliente X desconectou "+ *clienteBackEnd*.id);
41. })


45. })
46. //configuando a view do projeto para utilizar o ejs
47. app.set("view engine","ejs");
48. //criando rota principal que irá renderizar o arquvio index.ejs
49. app.get("/", (*req*, *res*) *=>* {
50. *res*.render('index')
51. })
52. http.listen(3000, () *=>* {
53. console.log("Servidor rodando");
54. })

**ENVIANDO MENSAGEM**

1. No frontend criamos a função enviar que irá capturar os dados dos inputs e através do método emit do socket clientBackEnd vamos emitir esses dados em um json.
2. *function* enviar(){
3. *var* msgField = document.getElementById("msg");
4. *var* usernameField = document.getElementById("username");
5. *var* msg = msgField.value;
6. *var* username = usernameField.value;
7. clienteFrontEnd.emit("msg",{
8. msg: msg,
9. username: username
10. })
11. }
12. No botão enviar vamos chamar a função enviar no evento onclick.
13. <button onclick="enviar()">Enviar</button>

**RECEBENDO A MENSAGEM NO BACKEND**

1. No banckend utilizamos o método on do socket clienteBackEnd vamos ouvir os eventos enviados do frontEnd pelo método ‘msg’. vamos ainda configurar o backend para enviar um emit utilizando o método ‘showmsg’ que será configurado no frontend.
2. *clienteBackEnd*.on("msg",(*data*)*=>*{
3. *clienteBackEnd*.emit('showmsg',*data*);
4. console.log(*data*);
5. })