

Facultad de telemática

Programación distribuida de servicios de internet

“Documentación”

Proyecto chat

José Armando Vázquez García

4°C

Mtro. MONTAÑO ARAUJO SERGIO ADRIAN

25/06/2021

**Índice**

**Introducción……………………………………………………………….3**

**Desarrollo  
 index.js…………………………………………………………….4-10**

**index.html…………………………………………………………11-23**

**Conclusión……………………………………………………………….23**

**Introducción**

El chat es una herramienta para que las personas puedan comunicarse con una persona o con varias personas en grupo como podrían ser las salas donde cierta cantidad de personas establecen una conversación entre ellos. El objetivo de este proyecto es crear un chat, para mostrarlo en un navegador web. Cuyas características o funciones del chat es ingresar por medio de login y seleccionando una sala, cumple con la función de que el usuario se registre y cambio de sala.

Para esto realizamos el chat usando HTML, con node.js y socket.io, en el mismo también incluimos algo de JavaScript, css para darle estilo y diseño al proyecto.

Para el desarrollo de este proyecto implementamos el siguiente código, en el index.js lleva la información de las conexión con la base de datos, sentencias sql para conseguir la información, validación de login y registros de los usuarios.

Index.js

En elindex.js se pone las partes donde se declaran las librerías o dependencias que utilizaremos durante todo el proyecto.

/\*se crean variables para las dependencias\*/

const express = require('express'), /\*para obtener express, en el caso de tener visual studio, en la terminal pones el comando npm install express\*/

socket = require('socket.io'), /\*para instalar socket.io utilizamos el comando npm install socket.io\*/

mysql = require('mysql'),/\*npm install mysql\*/

cookieParser = require('cookie-parser'),/\*npm install cookie-parser\*/

session = require('express-session');/\*npm install express-session\*/

var app = express(); /\*hacemos el uso de express, para la aplicacion\*/

var roomName = '';

const nameBot = "BotChat";

var server = app.listen(3030, function () {//corremos el servidor en el puerto 3030

  console.log("Servidor en marcha, port 3030.");

});

var io = socket(server);

var sessionMiddleware = session({

  secret: "keyUltraSecret",

  resave: true,

  saveUninitialized: true

});

io.use(function (socket, next) {

  sessionMiddleware(socket.request, socket.request.res, next);

});

app.use(sessionMiddleware);

app.use(cookieParser());

const config = {     /\*datos que se muestran en la terminal de visual studio\*/

  "host": "localhost",

  "user": "root",

  "password": "",

  "base": "chat"

};

var db = mysql.createConnection({  /\*funcion creada para la conexion a la base de datos\*/

  host     : 'localhost',

  user     : 'root',

  password : '',

  database : 'chat'

});

db.connect(function (err) {

  if (!!err)/\*el if nos sirve para validar o verificar que la conexion a la base no tuviera algun error \*/

  throw err;

    //de no haber ningun error semuestra un mensaje en la terminal, que se ha conectado a la base

  console.log('MySQL conectado: ' + config.host + ", usuario: " + config.user + ", Base de datos: " + config.base);

});

app.use(express.static('./'));//aqui nos direcciona al inicio en login

io.on('connection', function (socket) {//en esta parte es cuando se conecta el usuario al chat

  var req = socket.request;

  console.log(req.session);

    //en este if verificamos si el usuario esta dado de alta en la base de datos

    if(req.session.userID != null){

        db.query("SELECT \* FROM accounts WHERE id=?", [req.session.userID], function(err, rows, fields){

            console.log('Sesión iniciada con el UserID: ' + req.session.userID + ' Y nombre de usuario: ' + req.session.Username);

            socket.emit("logged\_in", {user: req.session.Username, email: req.session.correo});

        });//si el usuario esta registrado

    }else{

        console.log('No hay sesión iniciada');

    }

socket.on("login", function(data){

      const user = data.user,

      pass = data.pass;

      roomID = data.roomID;

      roomName = data.roomName;

      db.query("SELECT \* FROM accounts WHERE Username=?", [user], function(err, rows, fields){

          if(rows.length == 0){//por lo que hacemos una sentencia sql seleccionando los usuarios que estan registrados

            //si no existe tendria que registrarse

            console.log("El usuario no existe, favor de registrarse!");

          }else{

                console.log(rows);

                const dataUser = rows[0].Username,

                dataPass = rows[0].Password,

                dataCorreo = rows[0].email;

                if(dataPass == null || dataUser == null){

                    socket.emit("error");

                }

                if(user == dataUser && pass == dataPass){

                    console.log("Usuario correcto!");

                    socket.emit("logged\_in", {user: user, email: dataCorreo, room: roomName, roomID: roomID});

                    req.session.userID = rows[0].id;

                    req.session.Username = dataUser;

                    req.session.correo = dataCorreo;

                    req.session.roomID = roomID;

                    req.session.roomName = roomName;

                    req.session.save();

                    socket.emit('historial');

                    socket.join(req.session.roomName);

                    bottxt('entroSala');

                }else{

                    socket.emit("invalido");

                }

          }

      });

    });

    socket.on('historial', function(){

        console.log('Buscando el historial de la sala: ' + req.session.roomName);

//declaramos una sentencia sql donde llamamos la información de los mensajes de la sala seleccionada, con su respectivo usuario

        db.query('SELECT s.nombre\_sala, u.Username, m.mensaje FROM mensajes = INNER JOIN salas s ON s.id = m.sala\_id INNER JOIN users u ON u.id=m.user\_id WHERE m.sala\_id = ' + req.session.roomID + 'ORDER BY m.id ASC', function(err,rows,fields){

            socket.emit('armadoHistorial', rows); //evento donde nos envia los datos

        });

    });

    socket.on('addUser', function(data){

        const user = data.user,

        pass = data.pass,

        email = data.email;

        if(user != "" && pass != "" && email != ""){//este if verifica que todos los campos esten llenos con la informacion correspondiente

            console.log("Registrando el usuario: " + user);

            //en esta parte insertamos el usuario con una sentencia sql, que el usuario quedara registrado en la base de datos

            db.query("INSERT INTO accounts(`Username`, `Password`, `email`) VALUES(?, ?, ?)", [user, pass, email], function(err, result){

              if(!!err)

              throw err;

              console.log(result);

//console.log se mostrara en la terminal

              console.log('Usuario ' + user + " se dio de alta correctamente!.");

              socket.emit('UsuarioOK');

            });

        }else{

            socket.emit('vacio');

        }

    });

    socket.on('cambiodesala', function(data){

        const idSala = data.idSala,

        nombreSala = data.nombreSala;

        socket.leave(req.session.roomName);

        req.session.roomID = idSala;

        req.session.roomName = nombreSala;

        socket.join(req.session.roomName);

        bottxt('cambioSala');

    });

    socket.on('mjsNuevo', function(data){ // Función para crear el mensaje nuevo.

            db.query("INSERT INTO mensajes(`mensaje`, `user\_id`, `sala\_id`, `fecha`) VALUES(?, ?, ?, CURDATE())", [data, req.session.userID, req.session.roomID], function(err, result){

              if(!!err)

              throw err;

              console.log(result);

              console.log('Mensaje dado de alta correctamente!.');//este mensaje solo se vera reflejado en la terminal

                    socket.broadcast.to(req.session.roomName).emit('mensaje', {

                        usuario: req.session.Username,

                        mensaje: data

                    });

                    socket.emit('mensaje', {

                        usuario: req.session.Username,

                        mensaje: data

                    });

            });

    });

    socket.on('getSalas', function(data){//en esta parte de la base de datos todas las salas que existen en la misma

        db.query('SELECT id, nombre\_sala from salas', function(err, result, fields){

            if(err) throw err;

            socket.emit('Salas', result);

        });

    });

    function bottxt(data){

        entroSala = 'Bienvenido a la sala ' + req.session.roomName;//se obtiene el nombre de la sala que el usuario seleciono

        cambioSala = 'cambiaste de sala a ' + req.session.roomName;//obtenemos el nombre de la sala de la que se cambio segun el usuario

        sefue = 'el usuario ' + req.session.Username + ' ha salido.';

        if(data =="entroSala"){

            socket.emit('mensaje', {//emite un mensaje dentro del chat, indicando a la sala que se selecciono

                usuario: nameBot,

                mensaje: entroSala

            });

        }

        if(data =="cambioSala"){//emite un mensaje con el nombre de la sala, a la que se cambio el usuario

            socket.emit('mensaje', {

                usuario: nameBot,

                mensaje: cambioSala

            });

        }

        if(data =="salioUsuario"){

            socket.emit('mensaje', {

                usuario: nameBot,

                mensaje: sefue

            });

        }

    }

    socket.on('salir', function(request, response){

        bottxt('salioUsuario');//tambien se destruye la sesion de la sala en la que estaba el usuario

        socket.leave(req.session.roomName);

        req.session.destroy();//destruye la sesion iniciada

    });

});

Y el index.html este que lleva toda la parte de la interfaz grafica del chat, parte del diseño del chat

Index.html

<!doctype html>

<html lang="es">

  <head>

    <meta charset="utf-8">

    <link rel="shortcut icon" href="icono.jpg"> <!--este link nos pone el icono en la pestaña del navegador web-->

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <meta name="description" content="">

    <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors">

    <meta name="generator" content="Hugo 0.83.1">

    <title>Chat-NodeJs & Socket.IO</title>

    <link rel="canonical" href="https://getbootstrap.com/docs/5.0/examples/sign-in/">

    <!-- Bootstrap core CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOOl7+AMvyTG2x" crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-gtEjrD/SeCtmISkJkNUaaKMoLD0//ElJ19smozuHV6z3Iehds+3Ulb9Bn9Plx0x4" crossorigin="anonymous"></script>

<meta name="theme-color" content="#7952b3">

<!--Aqui comienza es apartado de estilos de css-->

    <style>

      html,

body {

  height: 100%;

}

body {

  display: flex;

  align-items: center;

  padding-top: 40px;

  padding-bottom: 40px;

  background-color: #24479250;

}

.form-signin { /\*diseño del formulario\*/

  width: 100%;

  max-width: 330px;

  padding: 15px;

  margin: auto;

}

body { /\*el ttamaño de la fuente y diseño del body\*/

    font:12px arial;

    color: #222;

    text-align:center;

    padding:35px; }

form, p, span {

    margin:0;

    padding:0; }

input { font:12px arial; }

a {

    color:#1515cc;

    text-decoration:none; }

    a:hover { text-decoration:underline; }

#wrapper, #loginform {

    margin:0 auto;

    padding-bottom:25px;

    background:#EBF4FB;

    width:504px;

    border:1px solid #ACD8F0; }

#loginform { padding-top:18px; }

#loginform p { margin: 5px; }

#chatbox {/\*diseño de la caja de texxto donde se mostraran los mensajes\*/

    text-align:left;

    margin:0 auto;

    margin-bottom:25px;

    padding:10px;

    background:#fff;

    height:270px;

    width:430px;

    border:1px solid #ACD8F0;

    overflow:auto; }

#usermsg {

    width:395px;

    border:1px solid #ACD8F0; }

#submit { width: 60px; }

.error { color: #ff0000; }

#menu { padding:12.5px 25px 12.5px 25px; }

.welcome { float:left; }

.logout { float:right; }

.msgln { margin:0 0 2px 0; }

#Login{

  background-color: rgb(90, 104, 211);

}

.encabezado{

  background-color: rgb(28, 29, 31);

  border-radius: 10px;

  width: 100%;

}

.imagen{/\*diseño de la imagen del chat\*/

  height: 90px;

  width: 80px;

}

.inicio\_login{

  margin-top: 15px;

  text-align: center;

  height: 100px;

  width: 300px;

  background-color: #222222;

  border-radius: 10px;

}

/\*neon effect\*/

h1,#tchat{

  font-size:6em;

  margin:20px 0 20px 0;

}

#tchat,h1{

  text-decoration:none;

  -webkit-transition: all 0.5s;

  -moz-transition: all 0.5s;

  transition: all 0.5s;

}

h1,#tchat:nth-child(1) {

  color:#0052cc;

  font-family:comic sans;

}

#tchat,h1:nth-child(1):hover{

  -webkit-animation: neon1 1.5s ease-in-out infinite alternate;

  -moz-animation: neon1 1.5s ease-in-out infinite alternate;

  animation: neon1 1.5s ease-in-out infinite alternate;

}

#tchat,h1:hover{

color:#ffffff;

}

/\*glow for webkit-ket-frames\*/

@-webkit-keyframes neon1 {

  from {

    text-shadow: 0 0 10px #fff,

               0 0 20px  #fff,

               0 0 30px  #fff,

               0 0 40px  #0052cc,

               0 0 70px  #0052cc,

               0 0 80px  #0052cc,

               0 0 100px #0052cc,

               0 0 150px #0052cc;

  }

  to {

    text-shadow: 0 0 5px #fff,

               0 0 10px #fff,

               0 0 15px #fff,

               0 0 20px #0052cc,

               0 0 35px #0052cc,

               0 0 40px #0052cc,

               0 0 50px #0052cc,

               0 0 75px #0052cc;

  }

}

/\*glow for mozilla-keyframes\*/

@-moz-keyframes neon1 {

  from {

    text-shadow: 0 0 10px #fff,

               0 0 20px  #fff,

               0 0 30px  #fff,

               0 0 40px  #0052cc,

               0 0 70px  #0052cc,

               0 0 80px  #0052cc,

               0 0 100px #0052cc,

               0 0 150px #0052cc;

  }

  to {

    text-shadow: 0 0 5px #fff,

               0 0 10px #fff,

               0 0 15px #fff,

               0 0 20px #0052cc,

               0 0 35px #0052cc,

               0 0 40px #0052cc,

               0 0 50px #0052cc,

               0 0 75px #0052cc;

  }

}

/\*glow\*/

@keyframes neon1 {

  from {

    text-shadow: 0 0 10px #fff,

               0 0 20px  #fff,

               0 0 30px  #fff,

               0 0 40px  #0052cc,

               0 0 70px  #0052cc,

               0 0 80px  #0052cc,

               0 0 100px #0052cc,

               0 0 150px #0052cc;

  }

  to {

    text-shadow: 0 0 5px #fff,

               0 0 10px #fff,

               0 0 15px #fff,

               0 0 20px #af2442,

               0 0 35px #cc0000,

               0 0 40px #cc000a,

               0 0 50px #cc0000,

               0 0 75px #e21436;

  }

}

.form-signin .checkbox {

  font-weight: 400;

}

.form-signin .form-floating:focus-within {

  z-index: 2;

}

.form-signin input[type="email"] {

  margin-bottom: -1px;

  border-bottom-right-radius: 0;

  border-bottom-left-radius: 0;

}

.form-signin input[type="password"] {

  margin-bottom: 10px;

  border-top-left-radius: 0;

  border-top-right-radius: 0;

}

      .bd-placeholder-img {

        font-size: 1.125rem;

        text-anchor: middle;

        -webkit-user-select: none;

        -moz-user-select: none;

        user-select: none;

      }

      @media (min-width: 768px) {

        .bd-placeholder-img-lg {

          font-size: 3.5rem;

        }

      }

    </style>

    <!-- Custom styles for this template -->

    <link href="signin.css" rel="stylesheet">

  </head>

  <body class="text-center">

    <!--parte principal del chat, formulario de login-->

<main class="form-signin">

  <div class="encabezado">

    <img src="icono3.png" class="imagen">

    <p id="tchat">CHAT</p>

  </div>

  <div class="inicio\_login">

    <h1 class="h3 mb-3 fw-normal" >Iniciar sesión</h1>

  </div>

    <div class="form-floating">

      <input type="text" class="form-control" id="userName" placeholder="name@example.com" name="username">

      <label for="floatingInput">Nombre de usuario</label>

    </div>

    <div class="form-floating">

      <input type="password" class="form-control" id="Password" placeholder="Password" name="password">

      <label for="floatingPassword">Contraseña</label>

    </div>

    <!--div para la seleccion de salas-->

    <div class="form-floating">

      <select class="form-select" name="rooms" id="rooms">

        <option selected >Selecciona la sala a ingresar</option>

        <option value="sala 1" id="1">Sala 1</option>

        <option value="sala 2" id="2">Sala 2</option>

        <option value="sala 3" id="3">Sala 3</option>

      </select>

    </div>

    <button class="w-100 btn btn-lg btn-primary" type="button" id="Login">Entrar</button>

    <button class="w-100 btn btn-lg btn-warning" type="button" id="registrar" data-toggle="modal" data-target="#registro">Registrar</button>

    <p class="mt-5 mb-3 text-muted">&copy; 2021</p>

</main>

<!--interfaz del chat donde se encuentran la caja de texto de los mensajes -->

    <div id="wrapper" style="display: none;">

      <div id="menu">

        <p class="bienvenido"> Bienvenido, <b id="usernameTag"></b>, con correo: <b id="emailUser"></b></p>

        <p class="logout"><a id="exit" href="/">Salir del chat</a></p>

      </div>

      <div id="chatbox">

<!--    div de la caja del chat que contendrá todos los mensajes. -->

      </div>

        <input name="usermsg" type="text" id="mensaje" size="63"/>

        <input type="button" name="submitmsg" id="enviarMensaje" value="Enviar Mensaje"/>

      <!--div donde se encuentran las salas a seleccionar-->

        <div class="form-floating">

          <select class="form-select" name="roomsCambio" id="roomsCambio">

            <option selected >Cambiar de sala</option>

            <option value="sala 1" id="1">Sala 1</option>

            <option value="sala 2" id="2">Sala 2</option>

            <option value="sala 3" id="3">Sala 3</option>

          </select>

        </div>

    </div>

<!-- Modal  de registro de usuario-->

<div class="modal fade" id="registro" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

  <div class="modal-dialog" role="document">

    <div class="modal-content">

      <div class="modal-header">

        <h5 class="modal-title" id="exampleModalLabel">Registro</h5>

        <button type="button" class="close" data-dismiss="modal" aria-label="Close">

          <span aria-hidden="true">&times;</span>

        </button>

      </div>

      <div class="modal-body">

        <div class="form-floating">

        <input type="text" class="form-control" id="userNameR" placeholder="name@example.com" name="username" required>

        <label for="floatingInput">Nombre de usuario</label>

      </div>

      <div class="form-floating">

        <input type="password" class="form-control" id="PasswordR" placeholder="Password" name="password" required>

        <label for="floatingPassword">Contraseña</label>

      </div>

      <div class="form-floating">

        <input type="email" class="form-control" id="correo" placeholder="correo" name="correo" required>

        <label for="floatingPassword">Correo</label>

      </div>

      </div>

      <div class="modal-footer">

        <button type="button" class="btn btn-secondary" data-dismiss="modal">Cerrar</button>

        <button type="button" class="btn btn-primary" id="sendResgistro">Registrar</button>

      </div>

    </div>

  </div>

</div>

<!--scripts que forman parte del diseño del chat obtenidos de bootstrap-->

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

<script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>

<script src="/socket.io/socket.io.js"></script>

<!--script donde tiene funciones y eventos del chat-->

<script>

    $(document).ready(function(){

      var socket = io();

      let salas = []; //declaracion de una variable tipo array

      socket.emit('getSalas');

      socket.on('Salas', function(data){

       // $.each(data, function(id, val){

         // salas.push(data[id].nombre\_sala);

        //});

        getSalas(salas);

      });

      $('#roomsCambio').change(function(){//funcion para hacer el cambio de sala

        roomID = $(this).find('option: selected').attr('id');

        roomName = $(this).find('option:selected').text();

        //roomID = roomID + 1;

        $('#SalaNombre').text(roomName);

        $('#chatbox').empty();

        socket.emit('cambiodesala', { //evento para el cambio de sala

          idSala: roomID,

          nombreSala: roomName

        });

        socket.emit('historial');

        console.log('cambio select a ID: ' + roomID + 'con nombre ' + roomName);

      });

      $("#Login").click(function(){ //evento donde toma los valores ingresados

        socket.emit("login", {

          user: $("#userName").val(),

          pass: $("#Password").val(),

          roomID: $('#rooms').val(),

          roomName: $('#rooms').find('option:selected').text()

        });

      });

      $("#sendResgistro").click(function(){//funcion donde se agrega un usuario

          socket.emit("addUser", {

          user: $("#userNameR").val(),

          pass: $("#PasswordR").val(),

          email: $("#correo").val(),

        });

      });

      $(".logout").click(function(){//funcion en la que el usuario sale del chat

        socket.emit("salir");

      });

      $('#registrar').click(function(){

        $("#userNameR").val("");

        $("#PasswordR").val("");

        $("#correo").val("");

      });

      $('#enviarMensaje').click(function(){

        if($("#mensaje").val().length <= 0){

          alert("Escribe el mensaje para poderlo enviar.");

        }else{

          var mensaje = $('#mensaje').val()

          socket.emit('mjsNuevo', mensaje); // Enviamos el nuevo mensaje a la función de mensaje nuevo.

        }

      });

      socket.on("logged\_in", function(data){

        console.log(data);

        $(".form-signin").hide();

        $("#wrapper").show();

        $('#usernameTag').text(data.user);

        $('#emailUser').text(data.email);

      });

      socket.on("invalido", function(){ //evento en el que el usuario no esta registrado o los datos

        //estan mal escritos manda un mensaje en pantalla

        alert("Usuario y/o contraseña incorrectos.");

      });

      socket.on("error", function(){

        alert("Error: Intenta de nuevo!");

      });

      socket.on("vacio", function(){//evento que manda mensaje en pantalla que no ha escrito la informacion en los campos

        alert("Error: Llena todos las campos.!");

      });

      socket.on("UsuarioOK", function(){//evento en el que señala que el usuario quedo registrdo

        $('#registro').modal('hide');

        alert("Dado de alta correctamente.");

      });

      socket.on('mensaje', function(data){ // Función que tiene de respuesta el nuevo mensaje, concatenamos e insertamos en la caja de chat.

          if(data.usuario == "BotChat"){

            var nuevoMensaje = '<small class="bot"><b>' + data.usuario +'</b>' + data.mensaje + '</small>';

            $('#chatbox').append(nuevoMensaje + '</br>');

            $('#mensaje').val("");

          }//var nuevoMensaje = '<b>' + data.usuario + ' dice:</b> ' + data.mensaje;

          else{

            var nuevoMensaje = '<p class="mensajeEnviado"><b>' + data.usuario + 'dice: </b>' + data.mensaje + '</p>';

            $('#chatbox').append(nuevoMensaje + '</br>');

            $('#mensaje').val("");

          }

       });

       socket.on('armadoHistorial', function(data){//evento en el que se va a juntar todos los mensajes del chat

         var historial = "";

         $.each(data, function(id, val){

           historial += '<p class="mensajeEnviado"><b>' + data[id]['Username'] + 'dijo: </b>' + data[id]['mensaje'] + '</p>';

         });

         historial += '<small class="bot"><b> BotChat -</b> Ultimos mensajes del historial de la sala </small>';

         $('#chatbox').append(historial +'</br>');

       });

    });

    function getSalas(data){

      $.each(data, function(id, val){

        $('#rooms').append($('<option>', {//funcion donde se selecciona la sala al login

          value: data[id].nombre\_sala,

          text: data[id].nombre\_sala,

          id: data[id].id

        }));

        $('#roomsCambio').append($('<option>', {//funcion deonde hace el cambio de sala

          value: data[id].nombre\_sala,

          text: data[id].nombre\_sala,

          id: data[id].id

        }));

      });

    };

</script>

  </body>

</html>

Conclusión:

En conclusión, con la realización de este proyecto sabemos que la estructura que lleva a la realización de un programa que beneficiara a personas estableciendo una comunicación entre ellos mismas.

Para así cumplir con el objetivo de crear un chat simple, solo con la funcione de enviar mensajes de texto, en el que utilizamos node.js y socket.io, así pues incluimos el lenguaje de HTML, CSS.