UNIVERSIDAD DE COLIMA

Facultad de Telemática
Ingeniería en Tecnologías de Internet
Semestre febrero -julio 2021

Colima Col., a 25 de junio de 2021

Asignatura: Programación distribuida en servicios de internet.

Profesor : Montaño Araujo Sergio Adrian.

“Documento Final del proyecto ”

Alumno:
Peralta Arciniega Rigoberto Missael.

Índice

Introducción:

En este documento se dará a conocer de manera breve explicación sobre la elaboración de este proyecto que fue desarrollar un chat interactivo e inicio de sesiones con Node.JS, JavaScript, MySQL y Express. La representación visual se realizó con el uso de HTML para dar una estructura básica y CSS para mejorar la apariencia del proyecto, además de agregar mi conclusión de lo que se ha aprendido de este proyecto.

Desarrollo:

Lo primero que se realizo fue crear los paquetes por NPM los cuales fueron:

- npm- init: Esto activará la inicialización de tu proyecto. Este comando funciona como una herramienta para crear el archivo package.json de un proyecto.

- npm express: Sirve para la escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas), la integración con motores de renderización de "vistas" para generar respuestas mediante la introducción de datos en plantillas, entre otras cosas.

- npm socket.io: Socket.IO es una biblioteca de JavaScript para aplicaciones web en tiempo real. Permite la comunicación bidireccional en tiempo real entre clientes y servidores web. Tiene dos partes: una biblioteca del lado del cliente que se ejecuta en el navegador y una biblioteca del lado del servidor para Node.js.

Después se crearon las instancias de socket, http y express, luego se desarrolló la interfaz del chat, con socket.emit enviamos información y con socket.on se recibe información o se crean usuarios u mensajes , para después crear la conexión con el cliente en el index.js y se puso en marcha el servicio web en el servidor.

```
var server = app.listen(3030, function () { //Conexion al servidor
  console.log("Servidor en marcha, port 3030.");
});

var io = socket(server);
```

Se puede observar que el servidor este puesto en marcha en el puerto 3030 y se declara la variable **io** para controlar el socket.io

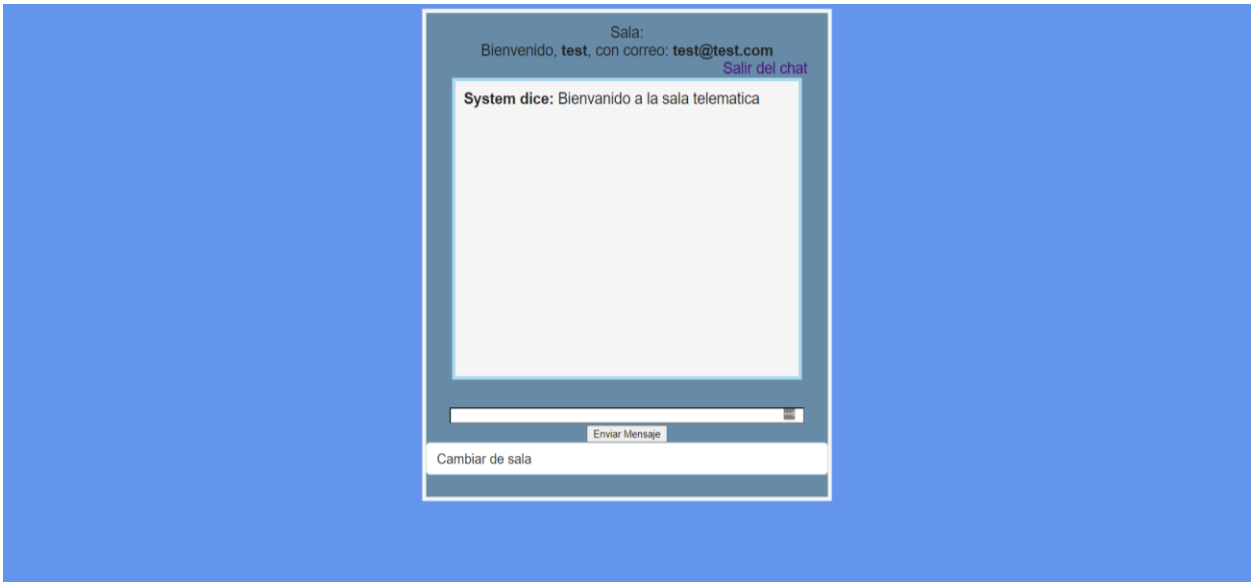

Figura 1.0. Se representa el primer paso del proyecto, la funcionalidad del chat.

Luego se empezó a manejar la librería de Mysql “npm install mysql”, en cual nos ayudará a controlar node.js para mysql, después la conectamos con los parámetros correspondientes como el nombre de la base o la contraseña , después se creó una base de datos con sus respectivas tablas de usuarios y mensajes, para el almacenamiento de los usuarios o los mensajes, además , de usar CURDATE() es una función de MySQL para insertar la fecha del servidor.

Para verificar si la conexión es exitosa, podemos realizar la siguiente validación:

```
var mysql = require('mysql');

var conn = mysql.createConnection({
```

```
    host:'localhost',
    database:'chat_progra',
    user:'root',
    password:''
});

conn.connect(function(err){
    if(err){
        throw err;
    }else{
        console.log('conexión exitosa!');
    }

});
```

Figura 1.1. Ejemplo de la conexión hacia la base de datos.

Donde 'conn' es la variable que contiene los datos de conexión a la base de datos, 'connect' es la función de la librería de MySQL que nos ayudará a conectar la base de datos.

Para realizar un alta de usuarios, al ingresar el usuario al chat realizar la siguiente Query:

conn.query('INSERT INTO usuarios(nombre_usuario, fecha) VALUES ("'+ nombre_de_usuario +'", CURDATE())');

Después desarrollamos el inicio con sesiones para ello se instalaron los paquetes:

- npm install express-session: almacena los datos de sesión en el servidor; sólo guarda el ID de sesión en la propia cookie, no los datos de sesión. De forma predeterminada, utiliza el almacenamiento en memoria y no está diseñado para un entorno de producción.

- npm cookie-parser: nos permite configurar cookies dentro de nuestro servidor.

Se creó en la base de datos la tabla de ‘’users’’, ya luego el formulario tanto de inicio como de registro con Bootstrap y CSS, y por último en el index.js se agregaron los paquetes que se utilizara Express.

Se cifro la contraseña con:

```
var sessionMiddleware = session({//cifrado de contraseñas
  secret: "keyUltraSecret",
```

Ya que la seguridad es una de las cosas más importantes en los sitios web.

Aquí se puede observar la verificación para la validación si el usuario existe en la base de datos y no notifica a usuario nuevo que tiene que registrarse.

```
 db.query("SELECT * FROM users WHERE Username=?", [user], function(err, rows, fie
lds){
          if(rows.length == 0){//Se verifica que el susuario exista, si no se not
ifica
            console.log("El usuario no existe, favor de registrarse!");
```

En este apartado de código, si se encontró al usuario se mandan sus datos por el socket.io.

```
if(user == dataUser && pass == dataPass){//Se encontro al usuario y se mandan los
 datos encontrados al socket
                    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.join(req.session.roomName);
                    bottxt('entroSala');
                }else{
                    socket.emit("invalido");
```

```
}
```

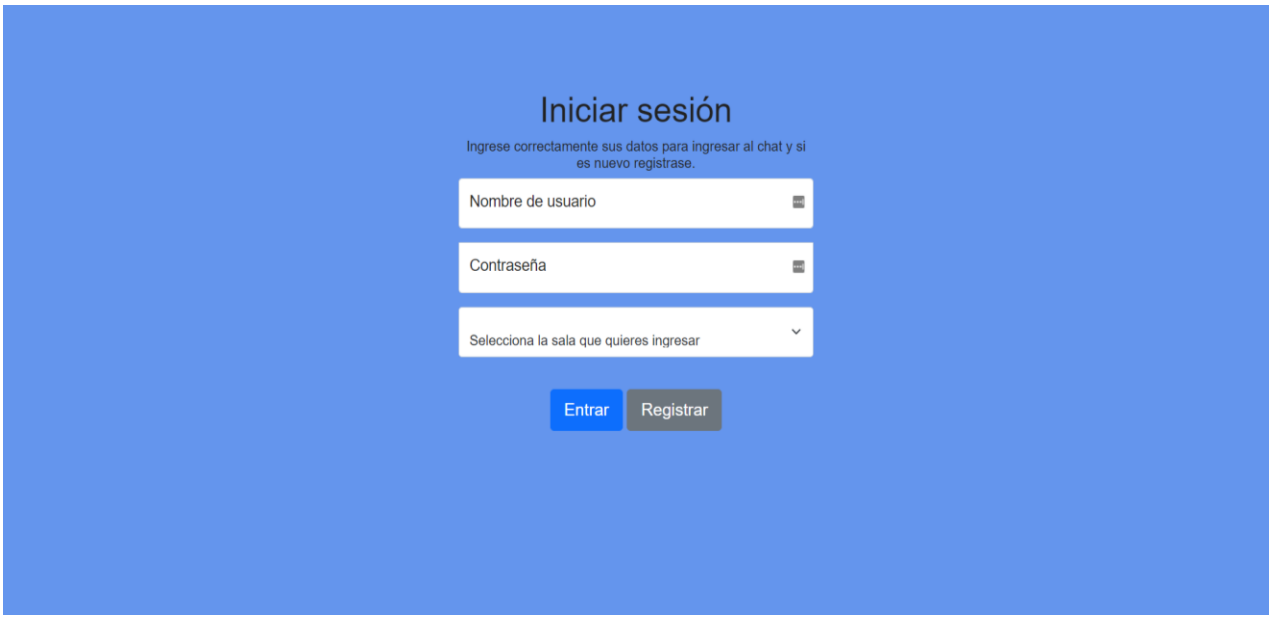

Figura 1.2. Se puede observar el login.

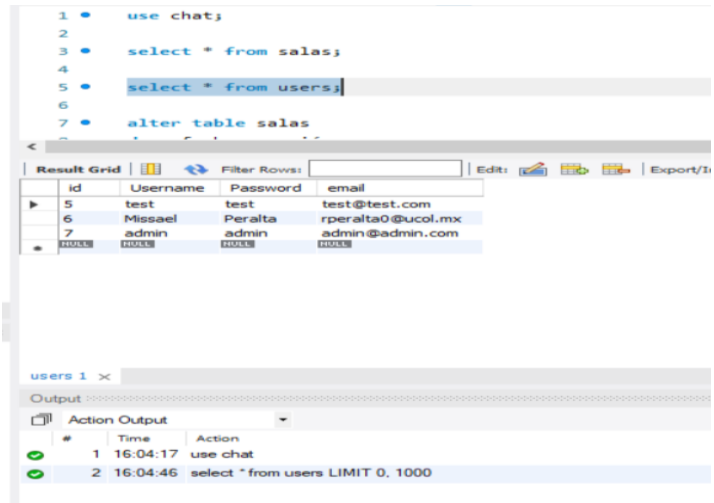

Figura 1.3. La base de datos, tabla users.

Ya por último se adicionamos para el usuario las “rooms” se pueden usar para transmitir eventos a un subconjunto de clientes. El chat cuenta con dos salas que son “software” y “telemática”. Para ello se añadieron las funciones de obtener los datos de la sala y para cambiar de sala como por ejemplo:

```
socket.on('cambiodesala', function(data){//Funcion para el cambio de sala dentro
del chat
        const idSala =data.idSala,
        nombreSala = data.nombreSala;
```

Glosario :

JavaScript: es un lenguaje de programación de scripts (secuencia de comandos) orientado a objetos.

HTML: es el lenguaje de marcado estándar para documentos diseñados para mostrarse en un navegador web.

Bidireccional: se establece cuando los protagonistas pueden enviar y recibir mensajes de modo simultáneo.

Repositorio de GitHub:

https://github.com/Missael-Peralta/Proyecto-final-sesiones-chat.git

conclusión :

En este proyecto he aprendido desarrollar un chat dinámico, utilizando diferentes librerías , permitiendo la comunicación simultanea entre varios usuarios con socket.io y un login de autenticación usando express y cookie-parser.