21/06/2021

ToDo API

Desarrollo del proyecto Final

JAVIER ANASTASIO BARRETO MARTINEZ
FACULTAD DE TELEMATICA

INTRODUCCIÓN

En este proyecto se desarrollará una pagina web la cual tenga las siguientes características:

- Vista para implementar tareas
  Como su nombre lo dice, en esta se podrá ingresar tareas y mostrarlas en tiempo real. Debe contener lo siguiente:
  - Un formulario para poder ingresar la información con los campos, nombre, descripción, fecha limite, [fata].

- Vista para revisar el progreso de las tareas.
- El uso de websockets para reflejar los cambios que realice un usuario al otro, en este caso si se crea una nueva actividad o se completa una, se debe de actualizar en tiempo real.

TECNOLOGÍAS

De tecnologías use varias dependencias de npm las cuales son:

- Cors
  Es una función el cual utiliza cabeceras HTTP para permitir el acceso a recursos seleccionados desde un servidor en un origen distinto (dominios).

- Body-parser
  Permite acceder a los datos, es decir, acceder al cuerpo y analizarlo en un objeto json para poder interactuar con este mismo.

- Express
  Permite el manejo de peticiones diferentes con rutas.

- WS (websockets)
  Permite el uso en vivo de un servidor y que los cambios que hagan los usuarios en una pagina se vean reflejadas en la otra.

- mysql
  Permite crear una o varias conexiones a varias bases de datos.

Tecnologías que instalé para ejecutar/comprobar mi avance en el proyecto.

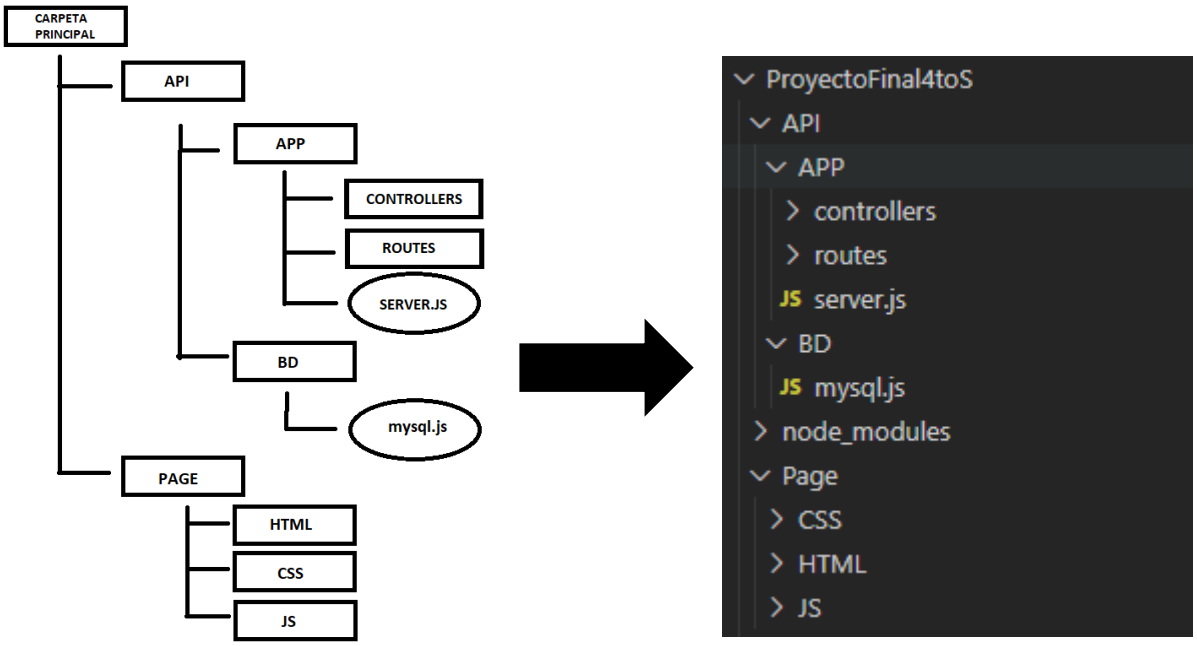
- Node.js
  Servicio que permite ejecutar códigos de javascript en tiempo real.

- Postman

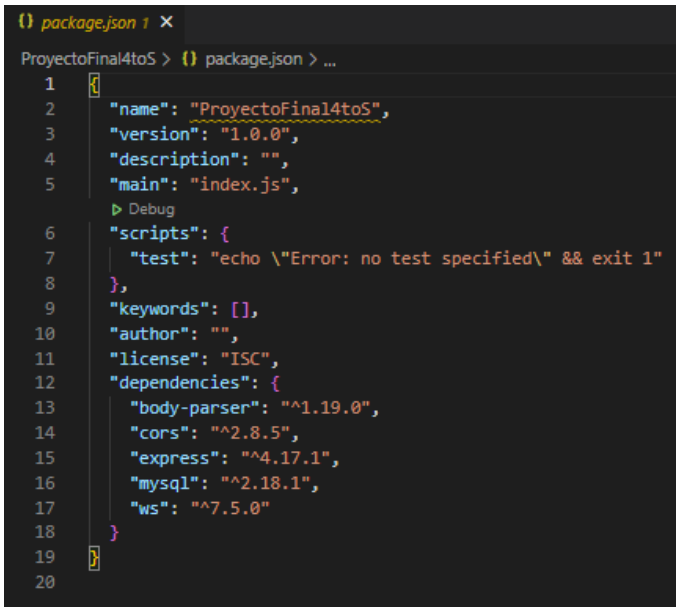
  Aplicación gratuita para comprobar el uso de rutas y ver como estas responden.

DESARROLLO

Primero cree la estructura de carpetas en la cual se encontrarán todos los archivos de mi proyecto.



Después instale las dependencias que iba a requerir para la creación de mi proyecto.



Configuré el archivo **mysql.js** para poder llamar la base de datos llamada dbactividades, y exportando la conexión para poder requerirla más tarde en el servidor.

```
mysql.js
ProyectoFinal4toS > API > BD > mysql.js > ...
1  const mysql = require('mysql');
2
3  const mysqlConnection=mysql.createConnection({
4      host:'localhost',
5      user:'root',
6      password:'@Jbderas12A',
7      database: 'dbactividades'
8  });
9
10 mysqlConnection.connect(function (err){
11     if(err){
12         console.log(err);
13         return;
14     }else{
15         console.log('DB is connected');
16     }
17 });
18
19 module.exports=mysqlConnection;
```

Una vez configurado el archivo **mysql.js**, empecé a configurar el archivo server.js, añadiendo las dependencias de cors, express y body-parser para así poder asignar puertos, asignar configuración de la información que se mandara, de la url y al final dándole una ruta principal. Luego, le dije a express en que puerto se ejecutara la app y al final imprimo mensajes para confirmar que el servidor se está ejecutando correctamente.

```
server.js
ProyectoFinal4toS > API > APP > server.js > ...
1  var express = require('express');
2  var app = express();
3  var cors=require('cors');
4  var bodyParser = require('body-parser');
5
6  var port = process.env.PORT || 1332;
7  app.use(cors());
8  app.use(bodyParser.urlencoded({ extended: true }));
9  app.use(bodyParser.json());
10
11 //ruta http://localhost:1332/api
12 var router = require('./routes');
13 app.use('/api', router);
14
15
16 //Info de aranque de servidor
17 app.listen(port);
18 console.log('API escuchando en el puerto ' + port);
```

Una vez creado el servidor, pensé en definir primero el controlador que se usara para las rutas de la API, por lo que creé el archivo **activitiescontroller.js** para que exportara unas funciones que usare más adelante y lo programé de la siguiente manera:

- NuevaA:
  Función para agregar una nueva actividad pendiente a la base de datos.

```
var mysql = require('../../BD/mysql');
//var activity = require('../models/activities');

module.exports = {
    NuevaA:(req,res)=>{// Crear nuevas actividades_pendientes
        mysql.query('INSERT INTO actividades_pendientes(nombre,descripcion,fecha,valor,seleccionada)  Values(?,?,?,?,?)',
        [req.body.nombre, req.body.dsc, req.body.fecha, req.body.valor, req.body.seleccionada], (err, rows, fields)=>{
            if(err)
            {
                res.json(err);
            }
            else
            {
                res.json(rows);
            }
        })
    },
```

- ListarAP:
  Función para obtener las actividades pendientes de la BD.

```
    ListarAP:(req,res)=>{// Obtener las actividades_pendientes
        mysql.query(`SELECT * FROM actividades_pendientes WHERE seleccionada = 'N'`, (err, rows, fields)=>
            if(err)
            {
                res.json(err);
            }
            else
            {
                res.json(rows);
            }
        })
    },
```

.

- ListarAC:
  Función para obtener las actividades completadas de la BD.

```
    ListarAC:(req,res)=>{// Obtener las actividades_completadas
        mysql.query('SELECT * FROM actividades_completadas', (err, rows, fields)=>{
            if(err)
            {
                res.json(err);
            }
            else
            {
                res.json(rows);
            }
        })
    },
```

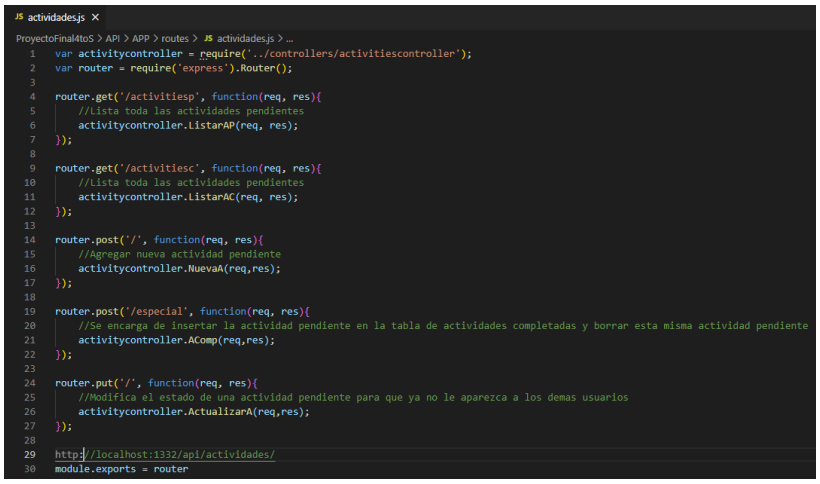
- ActualizarA:
  Se encarga de actualizar una actividad para que le desaparezca de las actividades pendientes.

```
ActualizarA:(req,res)=>{// Actualuzar una de las actividades_pendientes
    mysql.beginTransaction((err)=>{
        if(err)
        {
            rs.json(err);
        }
        else
        {
            mysql.query('UPDATE  actividades_pendientes SET seleccionada = ? WHERE id = ?',[req.body.selec, req.body.id] ,(err, rows, fields)=>{
                if(err)
                {
                    mysql.rollback(()=>{
                        res.json(err);
                     });
                }
                else
                {
                    mysql.commit(()=>{
                        if(err){
                           mysql.rollback(()=>{
                             res.json(err);
                           })
                        }
                    });
                }
            })
        }
    })
},
```

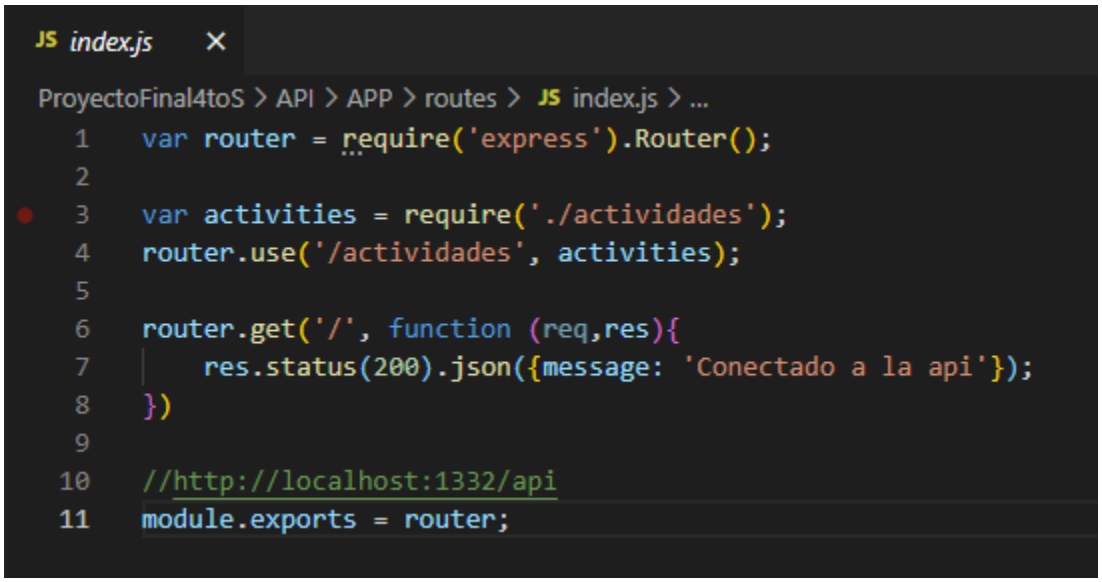
- Acomp:
  Se encarga de insertar una actividad pendiente en la tabla de actividades completadas y elimina esa misma actividad que estaba pendiente.

```
AComp:(req,res)=>{// Crear nuevas actividades_pendientes
    mysql.beginTransaction((err)=>{
        if(err)
        {
            res.json(err);
        }
        else
        {
            mysql.query('INSERT INTO actividades_completadas(nombre,descripcion,fecha,valor,info)  Values(?,?,?,?,?)',
            [req.body.nombre, req.body.descripcion, req.body.fecha, req.body.valor, req.body.info], (err, rows, fields)=>{
                if(err)
                {
                    mysql.rollback(()=>{
                        res.json(err);
                     });
                }
                else
                {
                    mysql.query('DELETE FROM actividades_pendientes WHERE id = ?', req.body.id, (err, rows, fields)=>{
                        if(err)
                        {
                            mysql.rollback(()=>{
                                res.json(err);
                             });
                        }
                        else
                        {
                            mysql.commit(()=>{
                                if(err){
                                   mysql.rollback(()=>{
                                     res.json(err);
                                   })
                                }
                            });
                        }
                    })
                }
            })
        }
    });
}
```

Después de eso creé el archivo **actividades.js** en la carpeta de *routes* y en el cual creé las rutas que estarán conectadas a las funciones para interactuar con la base de datos.



En la misma carpeta de *routes* creé el archivo de **index.js** en el cual defino cual será la ruta padre en la cual se podrán llamar las otras rutas que interactuaran con la base de datos.



Después probé las rutas con Postman para asegurarme de que las rutas funcionaran correctamente y en caso de detectar fallos, corregirlos.

Una vez creados todos los archivos de la base de datos, empecé a hacer la página web principal **index.js** la cual será de pura decoración para la página ToDo, ya que también cree dos paginas mas, una de estas páginas se llama **nactividades.js** donde estará el formulario para agregar una actividad nueva en tiempo real, y **actividades.js**, la cual permitirá a los usuarios ver las tareas completadas, las tareas pendientes, y ver en tiempo real cual tarea esta disponible, cual tarea esta completada y podrán en esa misma página mediante un popup entregar la actividad pendiente, y a su vez creé el estilo de cada página.

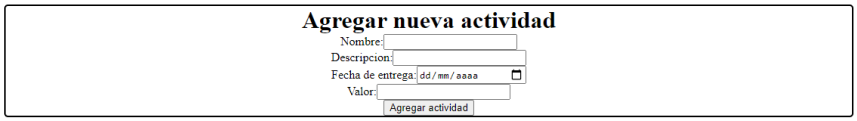
Index.html

index.html

ProyectoFinal4toS > Page > HTML > index.html > html > body > div.container > p > strong

```
<!DOCTYPE html>
<html>
    <head>
        <title>Escuela General Juan Benites</title>
        <link rel="stylesheet" href="../CSS/index.css">
    </head>
    <body>
        <header>
            <div class="headerbox">
                <ul>
                    <a href="./nactividades.html"><li>Agregar actividad</li></a>
                    <a href="./actividades.html"><li>Avance de las actividades</li></a>
                </ul>
            </div>
        </header>

        <div class="container">
            <h1>¡Bienvenid@ a la Escuela General Juan Benites!</h1>
            <p>Para agregar una nueva actividad, porfavor dirijase al menu superior y seleccione la opcion de <strong>Agregar actividad</strong>.
            Si quiere revisar el progreso de sus actividades, entregarlas o borrarlas, porfavor dirijase al menu superior y seleccione la opcion de
            <strong>Avance de las actividades</strong></div></p>
        </div>

    </body>
</html>
```

nactividades.html




```html
<!DOCTYPE html>
<html>
    <head>
        <title>Escuela General Juan Benites</title>
        <link rel="stylesheet" href="../CSS/nactividades.css">
    </head>
    <body>
        <header>
            <div class="headerbox">
                <ul>
                    <a href="./index.html"><li>Inicio</li></a>
                    <a href="./actividades.html"><li>Avance de las actividades</li></a>
                </ul>
            </div>
        </header>

        <div class="divnactividad">
            <h1>Agregar nueva actividad</h1>

            <form>
                <label for='txtname'>
                    Nombre:<input type="text" id="txtname">
                    <br>
                </label>

                <label for='txtdsc'>
                    Descripcion:<input type="text" id="txtdsc">
                    <br>
                </label>

                <label for='txtdate'>
                    Fecha de entrega:<input type="date" id="txtdate">
                    <br>
                </label>

                <label for='txtvalue'>
                    Valor:<input type="number" id="txtvalue">
                    <br>
                </label>
                <button type="button" id="AddActivity">Agregar actividad</button>
            </form>
        </div>
    <script src="../JS/nactividades.js"></script>
    </body>
</html>
```

actividades.html

Actividades Pendientes **Actividades Completadas**

actividades.html

ProyectoFinal4toS > Page > HTML > actividades.html > ...

```
<!DOCTYPE html>
<html>
    <head>
        <title>EGJB - Avance Actividades</title>
        <link rel="stylesheet" href="../CSS/actividades.css">
    </head>
    <body onload="alcargar()">
        <header>
            <div class="headerbox">
                <ul>
                    <a href="./index.html"><li>Inicio</li></a>
                    <a href="./nactividades.html"><li>Agregar actividad</li></a>
                </ul>
            </div>
        </header>

        <div class="divactividades">

            <div class="apendientesbox">
                <h3>Actividades Pendientes</h3>
                <div id="vap"></div>
            </div>

            <div class="acompletadasbox">
                <h3>Actividades Completadas</h3>
                <div id="vac"></div>
            </div>
        </div>

        <div class="overlay" id="overlay">
            <div class="popup" id="popup">
            </div>
        </div>
    </body>
    <script src="../JS/actividades.js"></script>
</html>
```

Después cree el archivo **nactividades.js** el cual se encargará de que el archivo **nactividades.html** pueda enviar la información a la base de datos.

nactividades.js

ProyectoFinal4toS > Page > JS > nactividades.js > ...

```
document.getElementById("AddActivity").addEventListener("click", ()=>{
    let name, dsc, fecha, valor;
    name=document.getElementById("txtname").value;
    dsc=document.getElementById("txtdsc").value;
    fecha=document.getElementById("txtdate").value;
    valor=document.getElementById("txtvalue").value;

    if(name==""||dsc==""||valor==""||fecha=="")
    {
        alert("Porfavor asegurese de haber llenado todos los datos")
    }
    else
    {
        var data ={nombre:name, dsc:dsc, fecha:fecha, valor:valor, seleccionada:'N'}
        fetch('http://localhost:1332/api/actividades/',{
            method:'POST',
            body:JSON.stringify(data),
            headers:{
                'Content-type':'application/json'
            }
        })
        .then(response=>response.json())
        .catch(error => console.log(error))
        .then(json=>console.log(json))

        alert("Nueva actividad agregada correctamente.");
    }
});
```

Después creé el archivo **actividades.js** el cual se encargará de que el archivo **actividades.html** pueda extraer y colocar la información de las actividades para que los usuarios puedan ver cuales se completaron y las que no se han completado, de misma manera las que no se han completado se podrán completar al momento de darle click a una de las actividades pendientes.

```
function alcargar(){
    fetch('http://localhost:1332/api/actividades/activitiesp/')
    .then(response=>response.json())
    .catch(error => console.log(error))
    .then(json=>{
        let x = document.getElementById("vap");
        x.innerHTML="";
        test = "";
        json.forEach(element => {
            x.innerHTML+=`
            <button class="activiidadesp" id="btnactp" onclick="quitar('Y',${element.id}, '${element.nombre}', '${element.descripcion}', '${element.fecha}',${element.valor})">${element.nombre}</button>
            <br>`;
        });

    })


    fetch('http://localhost:1332/api/actividades/activitiesc/')
    .then(response=>response.json())
    .catch(error => console.log(error))
    .then(json=>{
        let x = document.getElementById("vac");

        x.innerHTML="";
        json.forEach(element => {
            x.innerHTML+=`
            <button class="activiidadesc">${element.nombre}</button>
            <br>`;
        });
    })
}
```

```
function quitar(selec, id, nombre, dsc, fecha, valor){
    let data = {selec:selec, id:id}
    let data2 = {id:id, nombre:nombre, dsc:dsc, fecha:fecha, valor:valor}
    fetch('http://localhost:1332/api/actividades/',{
        method:'PUT',
        body:JSON.stringify(data),
        headers:{
            'Content-type':'application/json'
        }
    })
    .then(response=>response.json())
    .catch(error => console.log(error))
    .then(json=>console.log(json));


    let popup = document.getElementById("popup");

    popup.innerHTML=`
    <form>
        <h3>${nombre}</h3>
        <h4>${dsc}</h4>
        <div class="contenedor-input">
            <textarea id="txtinfo" cols="60" rows="10"></textarea>
        </div>
        <br>
        <button type="button" id="btnentregrar" class="btnentregrar" onclick="Entregar(${data2.id}, '${data2.nombre}', '${data2.dsc}', '${data2.fecha}',${data2.valor})">Entregar</button>
    </form>
    `;
    document.getElementById("overlay").classList.add("active");
}
```

```
function Entregar(id, nomb, dsc, fecha, valor){
    let info = document.getElementById("txtinfo").value;
    if(info=="")
    {
        alert("Porfavor, escriba algo en el campo de texto");
    }
    else
    {
        fecha = fecha.split('T',1);
        let data = {id:id, nombre:nomb, descripcion:dsc, fecha:fecha, valor:valor, info:info}
        console.log(data)
        fetch('http://localhost:1332/api/actividades/especial',{
            method:'POST',
            body:JSON.stringify(data),
            headers:{
                'Content-type':'application/json'
            }
        })
        .then(response=>response.json())
        .catch(error => console.log(error))
        .then(json=>console.log(json));
        document.getElementById("overlay").classList.remove("active");

    }
}


function onlyactpend(){
    fetch('http://localhost:1332/api/actividades/activitiesp/')
    .then(response=>response.json())
    .catch(error => console.log(error))
    .then(json=>{
        let x = document.getElementById("vap");
        x.innerHTML="";
        test = "";
        json.forEach(element => {
            x.innerHTML+=`
            <button class="activiidadesp" id="btnactp" onclick="quitar('Y',${element.id}, '${element.nombre}', '${element.descripcion}', '${element.fecha}',${element.
            <br>`;
        });

    })
}
```

Una vez que ya tenia toda la pagina con su funcionamiento con mysql, la probe y corregí errores para que esta funcionara correctamente. Después de que ya corroborara de que la página funcionara correctamente, comencé a añadir websockets por lo que primero agregue los websockets al servidor, dándole un puerto y configurándola para que hacer en caso de recibir mensajes.

server.js

ProyectoFinal4toS > API > APP > server.js > ...

```
var express = require('express');
var app = express();
var cors=require('cors');
var bodyParser = require('body-parser');
const ws = require('ws');
const wsc = new ws.Server({port:1331});

var port = process.env.PORT || 1332;
app.use(cors());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());

//ruta http://localhost:1332/api
var router = require('./routes');
app.use('/api', router);


//Info de arranque de servidor
app.listen(port);
console.log('API escuchando en el puerto ' + port);
console.log('Websocket activo en el puerto 1331');

//Websocket
var conexions = new Array();

wsc.on('connection', ws=>{
    conexions.push(ws);

    ws.on("message", event=>{
        if(event=="nac")
        {
            conexions.forEach(env =>{
                env.send(event);
            })
        }
        else
        {
            if(event=="nap")
            {
                conexions.forEach(env =>{
                    env.send(event);
                })
            }
        }
    })
});
```

Después de configurar el servidor para que use también websockets, también configure el archivo de **actividades.js** para que este envíe información al servidor websocket y pueda actualizar la información en tiempo real.

actividades.js

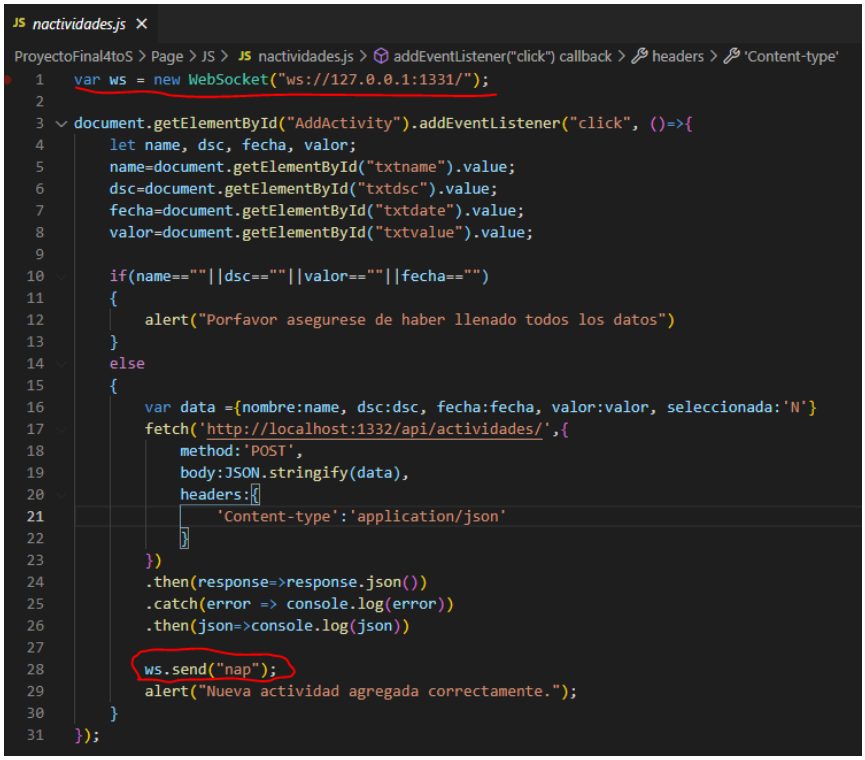
ProyectoFinal4toS > Page > JS > actividades.js > Entregar

```
var ws = new WebSocket("ws://127.0.0.1:1331/");

ws.onmessage = function(e) {
    console.log(e)
    if(e.data == "nap")
    {
        onlyactpend();
    }
    else
    {
        if(e.data=="nac")
        {
            alcargar();
        }
    }
};
```

```
function quitar(selec, id, nombre, dsc, fecha, valor){
    let data = {selec:selec, id:id}
    let data2 = {id:id, nombre:nombre, dsc:dsc, fecha:fecha, valor:valor}
    fetch('http://localhost:1332/api/actividades/',{
        method:'PUT',
        body:JSON.stringify(data),
        headers:{
            'Content-type':'application/json'
        }
    })
    .then(response=>response.json())
    .catch(error => console.log(error))
    .then(json->console.log(json));

    onlyactpend();
    ws.send("nap");

    let popup = document.getElementById("popup");

    popup.innerHTML=`
    <form>
        <h3>${nombre}</h3>
        <h4>${dsc}</h4>
        <div class="contenedor-input">
            <textarea id="txtinfo" cols="60" rows="10"></textarea>
        </div>
        <br>
        <button type="button" id="btnentregrar" class="btnentregrar" onclick="Entregar(${data2.id}, '${data2.nombre}', '${data2
    </form>
    `;
    document.getElementById("overlay").classList.add("active");
}
```

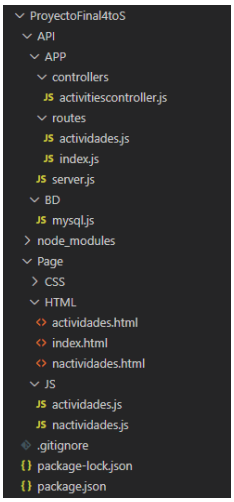
```
function Entregar(id, nomb, dsc, fecha, valor){
    let info = document.getElementById("txtinfo").value;
    fecha = fecha.split('T',1);
    let data = {id:id, nombre:nomb, descripcion:dsc, fecha:fecha, valor:valor, info:info}
    console.log(data)
    fetch('http://localhost:1332/api/actividades/especial',{
        method:'POST',
        body:JSON.stringify(data),
        headers:{
            'Content-type':'application/json'
        }
    })
    .then(response=>response.json())
    .catch(error => console.log(error))
    .then(json=>console.log(json));
    document.getElementById("overlay").classList.remove("active");
    ws.send("nac");
}
```

De igual manera, también agregue websockets en el archivo de **nactividades.js**.



Y al hacer eso y probar la pagina ya con todo y websockets, la probé y pude comprobar que todo funcionaba correctamente y que los cambios como agregar una nueva actividad o completar una, si se reflejaban en tiempo real.

Al final, así quedo la carpeta con todos los archivos:



CONCLUSIONES

En conclusión, gracias a este proyecto pude poner en practica todo lo que vi en este semestre y en semestres anteriores, pude reforzar lo aprendido de mysql con javascript, servidores locales, conexiones a base de datos, creación y funcionamiento de una API y pude aprender a cerca de cómo funcionan los websockets. También gracias a esta practica entiendo como es que se hace la actualización de elementos de un sitio web/aplicacion en tiempo real, como lo seria Messenger y WhatsApp al momento recibir mensajes, o como lo seria classroom que te avisa en tiempo real cuando se sube una actividad.