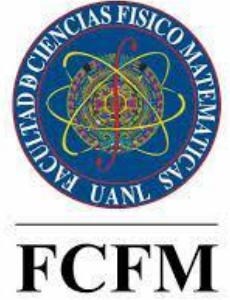




**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS**



PROGRAMACIÓN WEB DE CAPA INTERMEDIA (PWCI)
Docente: José Orihuela Sánchez

GAMELAND

GRUPO: 03

Alumno(s):

Pablo Andres Mariles Feregrino
Andrea Nahomi Rosas Sanchez 1803204

San Nicolás de los Garza, Nuevo León diciembre de 2020

GAMELAND

ELEMENTOS REQUERIDOS PARA LA ELABORACION DE NUESTRAS API'S

```
const express = require('express')
const app = express()
const port = 3000
var mysql = require("mysql");
var cors = require("cors");
app.use(express.json());
app.use(cors());
```

```
var connection = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "gameland",
});

connection.connect();

app.listen(port, () => {
  console.log(`Example app listening at http://localhost:${port}`);
});
```

USUARIOS

Apartado para explicar todo lo de la API relacionado con los usuarios

POST Registrarse

"http://localhost:3000/user"

Para registrar un usuario.

HEADERS

Content-Type

```
req.setRequestHeader("Content-Type", "application/json;charset=UTF-8");
```

BODY – Petición ejemplo

Nombre Usuario (NU): Andrea

Contraseña (Contra): Holasoyandrea

Correo (Correo): nahomiandrea@gmail.com

Público o privado (radioV): Público

PETICIÓN

```
var req = new XMLHttpRequest();
req.open("POST", "http://localhost:3000/user", true);
req.setRequestHeader("Content-Type", "application/json;charset=UTF-8");
req.send(
  JSON.stringify({
    NombreUsuario: NU,
    Contraseña: Contra,
    Correo: Correo,
    PoP: radioV
  })
);
```

RESPUESTA

200 OK

```
window.location.href='iniciarSesion.html';
```

GET Login

```
"GET","http://localhost:3000/user"
```

Tu ID de usuario será tomado gracias al llamado de arriba, el id servirá para identificar a cada usuario, será guardado de forma local.

HEADERS

Content-Type form-data

BODY

Ingresas los datos correspondientes en este caso

Nombre de usuario y contraseña

Al dar click al botón de iniciar sesión, se llamara al fetch, el cual hará un eventlistener del botón y llamara a la función hecha por nosotros la cual tiene la siguiente petición.

PETICIÓN

```
req.open ("GET","http://localhost:3000/user",true);

req.onreadystatechange = function (data) {
  if (req.readyState == 4 && req.status == 200) {
    var entro = false;
    var usuario = document.getElementById("formula1").value;
    var contra = document.getElementById("formula").value;
    var datos = data.target.response;
    console.log(datos);
    var parse = JSON.parse(datos);

    parse[0].forEach((k, v) => {
      console.log(`${k.NombreUsuario}`);
      console.log(`${v}`);
      if( (k.NombreUsuario == usuario) && (contra == k.Contraseña))
      {
        entro = true;
        localStorage.setItem("ID",k.IdUsuario);
      }
      else{
        if(entro == false){
          entro = false;
        }
      }
    });

    if(entro == false){
      alert("Usuario y Contraseña no coinciden");
    }
    else{
      window.open("inicio.html","_self");
    }
  }
}

req.send();
```

GET Perfil

```
"GET",`http://localhost:3000/user/${ID}`
```

PETICIÓN

```
var req = new XMLHttpRequest();
req.open("GET",`http://localhost:3000/user/${ID}`, true);
req.onreadystatechange = function (data) {
  if (req.readyState == 4 && req.status == 200) {
    var datos = data.target.response;
    console.log(datos);
    var parse = JSON.parse(datos);

    parse.forEach((k, v) => {
      console.log(`${k.NombreUsuario}`);
      console.log(`${v}`);
      document.getElementById("usuario").innerHTML = k.NombreUsuario;
      document.getElementById("Nombre").innerHTML = k.NombreUsuario;
      document.getElementById("Correo").innerHTML = k.Correo;
      if(k.PoP.data[0] == 1)
      {
        document.getElementById("PoP").innerHTML = "Publico";
      }
      else{
        document.getElementById("PoP").innerHTML = "Privado";
      }
    });
  }
  else{
    console.log("hola");
  }
}
req.send();
```

RESPUESTA

Se mostrará la información del usuario

POST Actualizar Usuario

```
"POST", "http://localhost:3000/editar"
```

HEADERS

Content-Type

```
req.setRequestHeader("Content-Type", "application/json;charset=UTF-8");
```

BODY – Petición ejemplo

Dentro del apartado `fetch()` se llamara con un `eventlistener` al click del botón para editar al perfil del usuario el cual llamara la información del perfil y una vez que le di click en aceptar se ira directamente a la función para guardar la información de nuevo

PETICIÓN:

```
var req = new XMLHttpRequest();
req.open("POST", "http://localhost:3000/editar", true);
req.setRequestHeader("Content-Type", "application/json;charset=UTF-8");
req.send(
  JSON.stringify({
    NombreUsuario: NU,
    Contraseña: Contra,
    Correo: Correo,
    PoP: radioPop,
    IdUsuario: IDedit
  })
);
onclick = location.reload();
```

GET Búsqueda por consola

```
"GET", `http://localhost:3000/busqueda/${busca}`
```

BODY – Petición ejemplo

Buscar: play

Aun sin ingresar el nombre completo de la consola, se pondrán las listas que están hechas para ella

PETICIÓN

```
var req = new XMLHttpRequest();
req.open("GET", `http://localhost:3000/busqueda/${busca}`, true);
req.onreadystatechange = function (data) {
  if (req.readyState == 4 && req.status == 200) {
    var datos = data.target.response;
    console.log(datos);
    var parse = JSON.parse(datos);
    let listas = 1;
    parse[0].forEach((k, v) => {
      $( "#listas" ).prepend( `<div class="card mb-3" style="max-width: 840px;" id="carta1" >
<div class="row no-gutters">
<div class="col-md-4">

</div>
<div class="col-md-8">
<div class="card-body">
<a href="#" id="verLista${listas}">${k.NombreLista}</a>
<p class="card-text">${k.Descripcion}</p>
<p class="card-text"><small id="Autor">Autor: ${k.NombreUsuario}</small></p>
</div>
</div>
</div>
</div>` );
      listas ++;
    });
  }
  else{
    console.log("hola");
  }
}
req.send();
```

RESPUESTA

```
parse[0].forEach((k, v) => {
  $( "#listas" ).prepend( `<div class="card mb-3" style="max-width: 840px;" id="carta1" >
<div class="row no-gutters">
<div class="col-md-4">

</div>
<div class="col-md-8">
<div class="card-body">
<a href="#" id="verLista${listas}">${k.NombreLista}</a>
<p class="card-text">${k.Descripcion}</p>
<p class="card-text"><small id="Autor">Autor: ${k.NombreUsuario}</small></p>
</div>
</div>
</div>
</div>` );
  listas ++;
});
```

LISTAS

GET Mostrar Listas de perfil de usuario

```
"GET", `http://localhost:3000/verListaUsuario/${ID}`
```

PETICIÓN

```
var req2 = new XMLHttpRequest();
req2.open("GET", `http://localhost:3000/verListaUsuario/${ID}`, true);
req2.onreadystatechange = function (data) {
  if (req2.readyState == 4 && req2.status == 200) {
    var datos = data.target.response;
    console.log(datos);
    var parse = JSON.parse(datos);

    parse[0].forEach((k, v) => {
      if(k.PrivPub.data[0] == 1)
      {
        var pop = "Publico";
      }
      else{
        var pop = "Privado";
      }
    })
  }
}
```

RESPUESTA

Todas las listas hechas por el perfil que este logueado

```
$("#ListasUsuario").prepend(`<div class="card mb-3" style="max-width: 840px;" id="carta">
<div class="row no-gutters">
  <div class="col-md-4">
    
  </div>
  <div class="col-md-8">
    <div class="card-body">
      <a href="#" >${k.NombreLista}</a>
      <p class="card-text">${k.Descripcion}</p>
      <p class="card-text"><small class="text-muted">${pop}</small></p>
      <p class="card-text"><small class="text-muted" id="id1${k.IdLista}">${k.IdLista}</small></p>
    </div>
    <button class="btn btn-warning" id="edita${k.IdLista}">Editar</button>
    <button class="btn btn-danger" id="elimina${k.IdLista}">Eliminar</button>
  </div>
</div>
</div>`)
```

GET Mostrar Listas en el inicio

```
"GET", `http://localhost:3000/listas`
```

Para mostrar listas en la página principal. Tiene un límite de 5 en la página principal, por lo cual se mostrarán las más recientes.

PETICIÓN

```
var req2 = new XMLHttpRequest();
req2.open("GET", `http://localhost:3000/listas`, true);
req2.onreadystatechange = function (data) {
  if (req2.readyState == 4 && req2.status == 200) {
    var datos = data.target.response;
    console.log(datos);
    var parse = JSON.parse(datos);
    var valor = parse.length;
    let listas = 1;
```

RESPUESTA

```
if(valor <= 5){
    $('#Listas').prepend(`<div class="card mb-3" style="max-width: 840px;" id="carta1" >
    <div class="row no-gutters">
    <div class="col-md-4">
    
    </div>
    <div class="col-md-8">
    <div class="card-body">
    <a href="#" id="verLista${listas}">${k.NombreLista}</a>
    <p class="card-text">${k.Descripcion}</p>
    <p class="card-text"><small id="Autor">Autor: ${k.NombreUsuario}</small></p>
    </div>
    </div>
    </div>
    </div>` );
    listas++;
}
```

POST Crear Lista

```
"POST", "http://localhost:3000/agregarLista"
```

HEADERS

Content-Type

```
req.setRequestHeader("Content-Type", "application/json; charset=UTF-8");
```

PETICIÓN

```
var req = new XMLHttpRequest();
req.open("POST", "http://localhost:3000/agregarLista", true);
req.setRequestHeader("Content-Type", "application/json; charset=UTF-8");
req.send(
    JSON.stringify({
        _NombreLista: tituloL,
        _PrivPub: PoP,
        _Autor: ID,
        _Descripcion: desc
    })
);
```

Después de agregar esa lista a la base de datos, dentro de la misma función se realizara el agregado de cada elemento de lista con el ID de la lista, y también el agregado de cada lista con su sección.

```
var req3 = new XMLHttpRequest();
req3.open("GET", "http://localhost:3000/IdLista/${tituloL}", true);
req3.onreadystatechange = function (data) {
    if (req3.readyState == 4 && req3.status == 200) {
        var datos = data.target.response;
        console.log(datos);
        var parse = JSON.parse(datos);

        parse[0].forEach((k, v) => {
            console.log(`${k.IdLista}`);

            for( var i = 0; i< secciones; i++)
            {
                if(xbox == 1 && xboxYA == false)
                {
                    var req2 = new XMLHttpRequest();
                    req2.open("POST", "http://localhost:3000/agregarSeccionLista", true);
                    req2.setRequestHeader("Content-Type", "application/json; charset=UTF-8");
                    req2.send(
                        JSON.stringify({
                            ids: xbox,
                            idL: k.IdLista
                        })
                    );
                    xboxYA = true;
                }
            }
        });
    }
}
```

Eso se realizará con cada sección que se encuentre, preguntando que sección agrego el usuario.

Y al final los elementos hechos se agregarán en la lista.

```
for(var ll = 0; ll < cantListas; ll++)
{
    var tituliElemento = document.getElementById("pin${ll + 1}`).value;
    var descElemento = document.getElementById("desc${ll+1}`).value;
    if(tituliElemento==""){
        alert("Te falta el titulo del elemento"+ (ll+1));
    }else{
        //poner lo otro
    }
    var req5 = new XMLHttpRequest();
    req5.open("POST", "http://localhost:3000/agregarElemento", true);
    req5.setRequestHeader("Content-Type", "application/json;charset=UTF-8");
    req5.send(
        JSON.stringify({
            _titulo: tituliElemento,
            _Descripcion: descElemento,
            _IdLista: k.IdLista
        })
    );
}
```

RESPUESTA

```
GuardarTodo = true;
window.location = 'paginaPerfil.html';
```

POST Editar Lista

```
"POST", "http://localhost:3000/EditaLista"
```

Para editar la lista.

HEADERS

Content-Type

```
req.setRequestHeader("Content-Type", "application/json;charset=UTF-8");
```

PETICIÓN

```
var req = new XMLHttpRequest();
req.open("POST", "http://localhost:3000/EditaLista", true);
req.setRequestHeader("Content-Type", "application/json;charset=UTF-8");
req.send(
    JSON.stringify({
        _IdLista: id,
        _NombreLista: TIT,
        _Descripcion: DESC,
        _PrivPub: radioPop,
    })
);
```

POST Eliminar Lista

```
"POST", "http://localhost:3000/borrarLista",
```

Para eliminar listas.

HEADERS

Content-Type

```
req.setRequestHeader("Content-Type", "application/json;charset=UTF-8");
```


BODY – Petición ejemplo

Se crea el post por cada lista creada en la pagina de perfil del usuario.

PETICIÓN

```
var req = new XMLHttpRequest();
req.open("POST", "http://localhost:3000/borrarLista", true);
req.setRequestHeader("Content-Type", "application/json;charset=UTF-8");
req.send(
  JSON.stringify({
    _IdLista: id
  })
);
```

RESPUESTA

```
window.location.href='paginaPerfil.html';
```

GET Ver autor de lista seleccionada

```
"GET",`http://localhost:3000/verListaUsuarioP/${nombre}`
```

HEADERS

Content-Type form-data

BODY – Petición ejemplo

PETICIÓN

```
var IDautor = localStorage.getItem("AutorVerLista");
var req2 = new XMLHttpRequest();
req2.open("GET",`http://localhost:3000/verListaUsuarioP/${nombre}`, true);
req2.onreadystatechange = function (data) {
  if (req2.readyState == 4 && req2.status == 200) {
    var datos = data.target.response;
    console.log(datos);
    var parse = JSON.parse(datos);

    parse[0].forEach((k, v) => {
      if(k.PrivPub.data[0] == 1)
      {
        var pop = "Publico";
      }
      else{
        var pop = "Privado";
      }
    })
  }
}
```

RESPUESTA

```
`$("#ListasUsuario").prepend(`<div class="card mb-3" style="max-width: 840px;" id="carta">
<div class="row no-gutters">
  <div class="col-md-4">
    
  </div>
  <div class="col-md-8">
    <div class="card-body">
      <a href="#" >${k.NombreLista}</a>
      <p class="card-text">${k.Descripcion}</p>
      <p class="card-text"><small class="text-muted">${pop}</small></p>
    </div>
  </div>
</div>
</div>`)
```