

Objetivo principal

Recolectar los datos de registro a través de un formulario, validarlos y enviarlos a una base de datos local y remota.(Ver video “objetivoActividad”)

Pasos a seguir

1. **Instalar NodeJS.** Hasta la creación de Node.js, allá por el año 2009, el lenguaje de programación JavaScript únicamente podía ejecutarse del lado del navegador o cliente, pero... ¿Qué ocurría si queríamos ejecutar este lenguaje fuera del navegador o del lado del servidor?

Node js es un entorno de desarrollo que permite ejecutar JavaScript en nuestra máquina.

Para instalar Node JS nos dirigimos a la dirección <https://nodejs.org/en> y descargamos el instalador. Ver video.

2. Creamos una carpeta llamada “server” en nuestro proyecto, nos ubicamos en ella y creamos un proyecto nuevo en NodeJS como lo indica el video.
3. Ejecutamos los siguientes comandos:

npm install ejs Este comando sirve para instalar un motor de plantillas que permiten generar vistas que contengan información y estructuras dinámicas.

npm install --save mysql2 Nos permite trabajar con una BD tipo SQL

4. Dentro de la carpeta “server” creamos el archivo “configBD.js” el cual va a contener la configuración necesaria para conectar a la BD. En este archivo escribimos el siguiente código.



```
1  const mysql = require('mysql2')
2
3  //conexión local
4  const connection = mysql.createConnection({
5      host: "localhost",
6      user: "root",
7      password: "",
8      database: "bd2711993",
9      port: 3306,
10 })
11
12 connection.connect((err)=>{
13     if(!err){console.log("DataBase connected succesfully")}
14     else{
15         console.log("DataBase coneccction failed")
16     }
17 })
18 module.exports = connection
```

5. En el archivo "index.js" que es nuestro servidor de backend escribimos el siguiente código. Para iniciar el servidor escribimos "node index.js"

```

1  const express = require('express')
2  const app = express()
3  const cors = require("cors")
4  const bodyParser = require('body-parser')
5  const path = require("path")
6
7
8  //Requiriendo la conexión a BD
9  const connection = require("../configBD")
10
11 app.use(cors())
12 app.use(bodyParser.urlencoded({ extended: true }))
13 app.use(bodyParser.json())
14 app.use(express.urlencoded());
15
16 app.use("/public", express.static(path.join(__dirname, "public")))
17
18 app.set("view engine", "ejs")
19 app.set("views", path.join(__dirname, "views"))
20
21 app.get("/", (req, res) => {
22     res.send("saludando desde el backend")
23 })
24
25 app.get("/form-registro", (req, res) => {
26     res.render("formulario")
27 })
28
29 app.post('/registrarse', (req, res) => {
30     // res.send("Saludando desde el servidor Backend")
31     for (const campo in req.body) {
32         if (!req.body[campo]) {
33             res.send(`Error: el campo ${campo} está vacío`)
34             return
35         }
36     }
37     //Desestructurando los datos que vienen en el body
38     const { documento, nombre, apellido, usuario, email, pass } = JSON.parse(JSON.stringify(req.body)); //para que el body no llegue como Undefined
39
40     try {
41         const query = "INSERT INTO usuario(id,nombres,apellidos,usuario,correo,password,fechaCreacion) VALUES (?, ?, ?, ?, ?, ?)";
42         connection.execute(query, [documento, nombre, apellido, usuario, email, pass, new Date()]);
43         res.send("Registro exitoso")
44     }
45     catch (error) {
46         console.error("Error al insertar en la BD ", error)
47         res.send("Error al procesar el formulario")
48     }
49 })
50
51 const port = 3001
52 app.listen(port, () => {
53     console.log("Server is listening in port ", port)
54 })

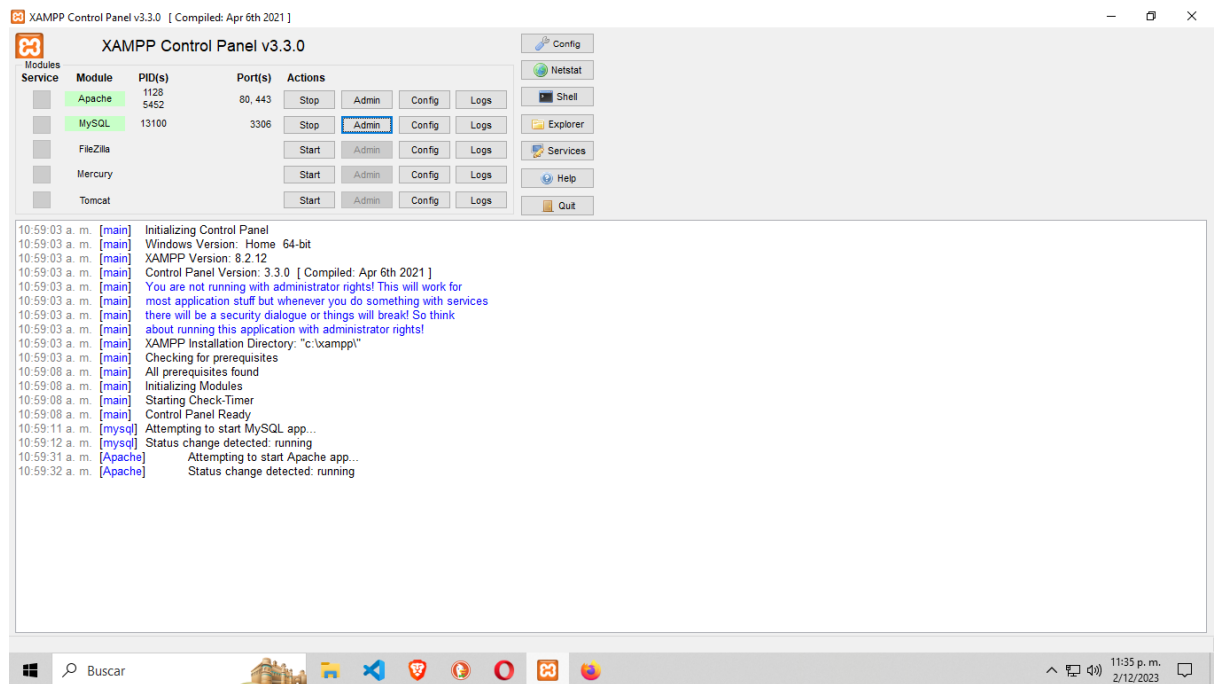
```

6. Dentro de la carpeta server creamos otra llamada “views”. Aquí es donde nuestro motor de plantillas buscará las vistas a renderizar. Por ello, las vistas deben estar al interior de esta carpeta. Por ejemplo, la vista formulario.html debe estar alojada allí y con la extensión .ejs
7. Instalamos XAMPP. Xampp es un servidor web local multiplataforma que permite la creación y prueba de páginas web u otros elementos de programación. Sin embargo, Xampp integra una serie de herramientas que potencian y facilitan la experiencia al desarrollador.

<https://www.apachefriends.org/es/index.html>

Una vez instalado XAMPP lo abrimos e iniciamos la BD clickeando el botón “start” . Luego hacemos click en el botón “admin” para cargar phpmyadmin y allí crear la BD.

Nota: si no abre "<http://localhost/phpmyadmin/>" entonces iniciamos el servidor Apache clickeando en el botón "start"



8. Creamos la BD en "phpMyadmin". Ver video "creacionBDphpMyadmin"

Árbol de carpetas del proyecto

