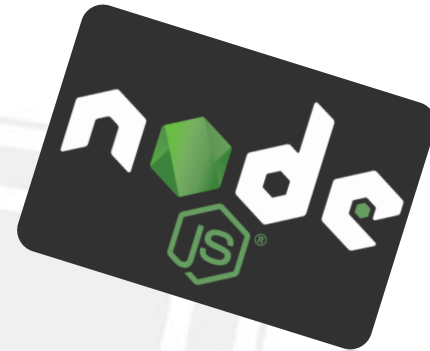




PROFE VIDERMID



Back-End con NodeJS



<https://nodejs.org/es>



@ingenieriadigitalsc

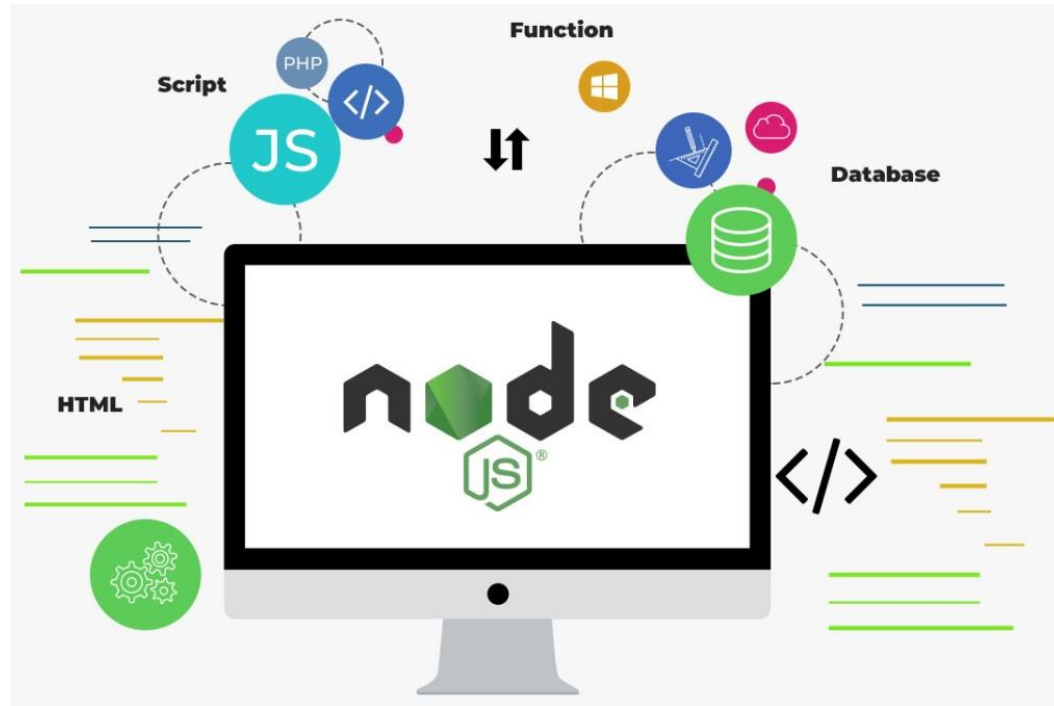


+584147464801

FACILITADOR: ING. ESP. VIDERMID SÁNCHEZ



¿Qué es NodeJS?



Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables. Pueden atenderse muchas conexiones simultáneamente. Por cada conexión, se activa la devolución de llamada o callback, pero si no hay trabajo que hacer, Node.js se dormirá.



¿Qué es NodeJS?

Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables. Pueden atenderse muchas conexiones simultáneamente. Por cada conexión, se activa la devolución de llamada o callback, pero si no hay trabajo que hacer, Node.js se dormirá.

Éste funciona mediante la incorporación de módulos que facilitan el desarrollo de aplicaciones Cliente – Servidor, conexión a diferentes SMD como MariaDB, MySQL, SQLite, PostgreSQL, entre otros. Además la compatibilidad con diferentes Frameworks para optimizar los desarrollos, entre ellos: Bootstrap, ExpressJS. Asimismo, es de fácil integración con librerías como AngularJS, bibliotecas como ReactJS o Embedded JavaScript (EJS).



La primera app en NodeJS

index.js

```
const http = require('http');
const hostname = '127.0.0.1';
const port = 5050;

const server = http.createServer((req, res) => {
  res.statusCode = 200; // indica que la solicitud ha tenido éxito
  res.setHeader('Content-Type', 'text/plain');
  res.write('Mi primera APP desde NodeJS \n');
  res.end('Hola Mundo');
});

server.listen(port, hostname, () => {
  console.log(`El servidor se está ejecutando en http://${hostname}:${port}/`);
});
```

Ejecutamos en la terminal: `node nombre.js` y se termina con `Ctrl+C`



Creando el Proyecto

Se crea el archivo de control de paquetes package.json, el cual tiene formato clave – valor, es decir, un script json.

npm init –y

Seguidamente, procedemos a instalar los módulos requeridos para el desarrollo de la aplicación:

npm install express ejs mysql nodemon

Posteriormente, se debe crear el archivo principal de la aplicación (index.js):

Para este caso le colocaremos app.js (el nombre puede ser cualquiera pero respetando la extensión js)

Ahora, en el script package.json, cambiamos el valor de la clave “main” de “index.js” por “app.js”

Seguidamente se agrega la siguiente clave – valor:

```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "start": "node app.js",  
  "dev": "nodemon app.js"  
},
```

Ahora ejecutamos:

```
npm run dev
```



App.js

Referenciamos a expressJS:

```
const express= require('express');
```

Invocamos mediante su clase:

```
const app=express();
```

Prueba básica:

```
app.listen(5050, ()=>{  
  console.log(5050,'Estamos runnnnn desde http://127.0.0.1:5050');  
});
```

Ejecutamos la aplicación:

```
node app
```

Saldrá el mensaje “Cannot GET /”, esto indica que no hay nada en la raíz

Lo siguiente después de realizar prueba básica

Enrutamos manualmente a la raíz ó “/”:

```
app.use('/', (require, response) => {  
  response.send('<h1>Activo el Server . . .</h1>'); //contenido de la web  
  response.end;  
});
```

Ahora ejecutamos:

```
npm run dev
```



App.js

Invocamos al motor de plantillas EJS:

```
app.set('view engine','ejs');
```

En el Proyecto hacer la siguiente estructura de archivos:

Crear en la raíz el archivo script para enrutar los componentes:

```
router.js
```

Crear las siguientes carpetas para organizar los components de la aplicación:

views:

archivos para las vistas del CRUD:

index.ejs edit.ejs create.ejs, para delete no se realiza vista.

database:

archivos para conexión:

```
db.js
```

controllers:

```
crud.js
```



router.js

Referenciamos a expressJS:

```
const express=require('express');
```

Invocamos al modulo de enrutamiento de expressJS:

```
const router=express.Router();
```

Creamos un nueva ruta:

Ejemplo:

```
router.get('/contacto',(request,response)=>{  
    response.send('CONTACTO');  
});
```

Exportamos el enrutador para poder utilizarlo desde la aplicación:

```
module.exports=router;
```

app.js

Referenciamos al enrutador:

```
app.use('/',require('./router'));
```




db.js

Referenciamos al modulo para el SGBD:

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

Creamos la conexión a la base de datos:

```
const conexion=mysql.createConnection({  
  host: 'localhost',  
  user: 'root',  
  password: '',  
  database: 'crud_nodejs_db'  
});
```

Ejecutamos la conexión:

```
conexion.connect((error)=>{  
  if(error){  
    console.error('El error de conexion es: '+error);  
    return  
  }  
  console.log('Conectado a la BD de forma correcta . . .');  
});
```

Exportamos la conexión para poder utilizarla desde la aplicación:

```
module.exports=conexion;
```



router.js

Agregamos en la cabecera la referencia a la conexión de la BD:

```
const conexion=require('./database/db');
```

Modificamos la ruta:

```
router.get('/',(request,response)=>{  
  conexion.query('SELECT * FROM users', (error, results)=>{  
    if(error){  
      throw error; //muestra error por consola  
    }else  
    {  
      response.send(results); //envia resultados consulta  
    }  
  })  
});
```