Notas Node.js

Introducción a Node.js

- **Características clave**:
  - Ejecuta JavaScript en el servidor con el motor V8 de Google Chrome.
  - **Non-blocking I/O**: Manejo eficiente de operaciones de entrada/salida.
  - **Eventos asíncronos**: Usa el modelo de eventos para gestionar múltiples solicitudes.
- **Casos de uso**:
  - Aplicaciones web en tiempo real (ej.: chats y streaming).
  - APIs RESTful.
  - Automatización y herramientas de línea de comandos.

Instalación y Configuración

1. **Descargar Node.js**:
   - Página oficial: Node.js
   - Elegir versión LTS para mayor estabilidad.

Comprobación:

```
node -v  # Verifica la versión de Node.js instalada
npm -v   # Verifica la versión de npm instalada
```

2. **Node Version Manager (NVM)**:

Instalación:

```
curl -o-
https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh |
bash
```

Comandos básicos:

```
nvm install 16.20.0  # Instalar una versión específica
nvm use 16.20.0      # Usar una versión instalada
nvm list             # Listar versiones instaladas
```

Manejo de Paquetes con npm
Inicializar un proyecto:

```
npm init -y  # Crea un archivo package.json con valores
predeterminados
```

1. **Instalar dependencias**:

Globalmente:

```
npm install -g nodemon
```

Localmente:

```
npm install express
```

2. **Scripts personalizados**:

Definir en `package.json`:

```
"scripts": {
  "start": "node index.js",
  "dev": "nodemon index.js"
}
```

Ejecutar:

```
npm run start
npm run dev
```

Creación de un Servidor con Node.js
Servidor básico:

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

const server = http.createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('Hello, World!');
});

server.listen(3000, () => {
  console.log('Server is running on http://localhost:3000');
});
```

Servidor con rutas:

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

const server = http.createServer((req, res) => {
```

```
  if (req.url === '/about') {
    res.writeHead(200, { 'Content-Type': 'text/plain' });
    res.end('About Page');
  } else {
    res.writeHead(404, { 'Content-Type': 'text/plain' });
    res.end('Not Found');
  }
});

server.listen(3000, () => {
  console.log('Server is running on http://localhost:3000');
});
```

Manejo de Archivos con fs
Lectura de archivos:

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

fs.readFile('example.txt', 'utf8', (err, data) => {
  if (err) throw err;
  console.log(data);
});
```

Escritura de archivos:

```
fs.writeFile('example.txt', 'Hello Node.js!', (err) => {
  if (err) throw err;
  console.log('File has been saved!');
});
```

Usando promesas:

```
const fs = require('fs').promises;

async function readFile() {
  try {
    const data = await fs.readFile('example.txt', 'utf8');
    console.log(data);
  } catch (err) {
    console.error(err);
  }
}
```

```
readFile();
```

Manejo del Event Loop
Modelo asíncrono:

```
setTimeout(() => console.log('Timeout'), 0);
console.log('Synchronous log');
// Output: "Synchronous log" -> "Timeout"
```

Evitar bloqueos:

```
const fs = require('fs');
fs.readFile('largeFile.txt', 'utf8', (err, data) => {
  if (err) throw err;
  console.log('File read asynchronously');
});
console.log('This will log first');
```

Express.js: Framework para Servidores Web
Instalación:

```
npm install express
```

Servidor básico con Express:

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

app.get('/', (req, res) => {
  res.send('Hello Express!');
});

app.listen(3000, () => {
  console.log('Server is running on http://localhost:3000');
});
```

Middleware personalizado:

```
app.use((req, res, next) => {
  console.log(`${req.method} request to ${req.url}`);
  next();
```

```
});
```

Socket.IO: Comunicación en Tiempo Real
Instalación:

```
npm install socket.io
```

Servidor básico de chat:

```
const { Server } = require('socket.io');
const io = new Server(3000);

io.on('connection', (socket) => {
  console.log('User connected');
  socket.on('message', (msg) => {
    console.log(`Message: ${msg}`);
    io.emit('message', msg);
  });
});
```

Recursos Útiles

1. **Documentación**:
    - Node.js Docs
    - Express API
    - Socket.IO Docs
2. **Tutoriales**:
    - Node.js Handbook
    - Learn Node.js
    - FreeCodeCamp Node.js Guide