

## [Node.js](#)

Node es un entorno para ejecutar todo lo que tenga que ver con java script en el backend Normalmente usado para web, apis y aplicaciones en tiempo real, es asincro donde no se espera a que las otras tareas se hagan para poder ejecutar una tarea, en nodejs se usa npm para poder descargar paquetes casi igual con pip install o gestionar paquetes

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

fs.readFile('texto.txt', 'utf8', (err, data) => {
  if (err) {
    console.error('Error al leer el archivo:', err);
    return;
  }

  console.log('Contenido del archivo:', data);
});
```

Aquí se muestra un código simple donde se lee un texto de manera asíncrona

-Const se usa para definir una constante como dice su nombre y require para importar el módulo file system

-El file system lee el archivo y si ocurre algún error (err) se muestra por la consola el mensaje

-En el console log si no ocurre ningún error se muestra el contenido del archivo

-El utf8 se usa para que se soporten los tildes y emojis pero no es de node

Aunque normalmente se usa en paginas con http y servers como podría ser

```

const http = require('http');
const server = http.createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/html; charset=utf-8' });

  if (req.url === '/') {
    res.write('<h1>hola</h1>');
    res.write('<p>estas en la página de inicio</p>');
  } else if (req.url === '/about') {
    res.write('<h1>acerca de</h1>');
    res.write('<p>hola</p>');
  } else if (req.url === '/contacto') {
    res.write('<h1>Contacto</h1>');
    res.write('<p>este es mi contacto contacto@1234.com</p>');
  } else {
    res.writeHead(404, { 'Content-Type': 'text/html; charset=utf-8'
  });
    res.write('<h1>404 - Página no encontrada</h1>');
  }
  res.end();
});
server.listen(3000, () => {
  console.log('Servidor corriendo en http://localhost:3000');
});

```

Aquí con `createserver` se crea el server al igual que el request y response son definidos despues en `res.writedHead` se responde con un html siempre que se entra a la página. En el if dependiendo de la página o ruta (`localhost/contacto`) se mostrará alguna de las url y se termina la respuesta. Finalmente en el server listen dice que puerto escuchará (<http://localhost:3000> )

En general casi todos los lenguajes de programación de alto nivel se parecen solo que las terminologías son diferentes para definir una constante se usa `const` aunque python no lo use todo lo que se define como `jose = 1` es una constante, `file system` o `fs` sirve para leer o escribir en archivos como `fs.writeFileSync` o `fs.readFileSync`, siento que en lo que mas destaca es en poder levantar servidores sin necesidad de apache o Nginx con `createserver` y el administrador de paquetes `npm`, en general el lenguaje sirve porque en nuestro proyecto todo se va basar en enviar y recibir datos de tipo json junto con el framework `express` para alivianar el trabajo de nodejs y la comunicación con la base de datos de `mysql` que se ve relativamente fácil.

No se que podría aportar diferente a los otros documentos de mis compañeros más que una pequeña demostración de como es la lógica.