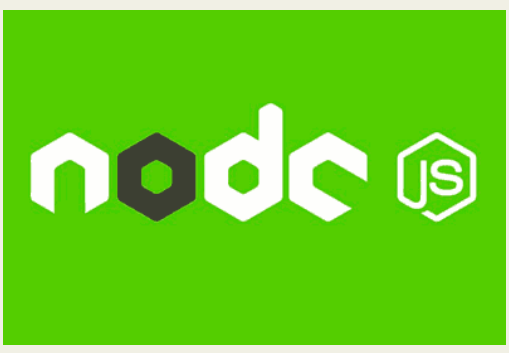
INTERFAZ DE PROGRAMACIÓN DE APLICACIONES(API)

¿QUE ES?

UNA INTERFAZ DE PROGRAMACIÓN DE APLICACIONES (API) ES UN CÓDIGO QUE PERMITE QUE DOS PROGRAMAS DE SOFTWARE SE COMUNIQUEN ENTRE SÍ.
LA API DEFINE LA FORMA CORRECTA PARA QUE UN DESARROLLADOR ESCRIBA UN PROGRAMA QUE SOLICITE SERVICIOS DE UN SISTEMA OPERATIVO (SO) U OTRA APLICACIÓN. LAS API SE IMPLEMENTAN MEDIANTE LLAMADAS A FUNCIONES COMPUESTAS DE VERBOS Y SUSTANTIVOS. LA SINTAXIS REQUERIDA SE DESCRIBE EN LA DOCUMENTACIÓN DE LA APLICACIÓN A LA QUE SE LLAMA.

¿QUE ES NODE JS?

NODE.JS, ES UN ENTORNO EN TIEMPO DE EJECUCIÓN MULTIPLATAFORMA PARA LA CAPA DEL SERVIDOR (EN EL LADO DEL SERVIDOR) BASADO EN JAVASCRIPT.
NODE.JS ES UN ENTORNO CONTROLADO POR EVENTOS DISEÑADO PARA CREAR APLICACIONES ESCALABLES, PERMITIÉNDOTE ESTABLECER Y GESTIONAR MÚLTIPLES CONEXIONES AL MISMO TIEMPO. GRACIAS A ESTA CARACTERÍSTICA, NO TIENES QUE PREOCUPARTE CON EL BLOQUEO DE PROCESOS, PUES NO HAY BLOQUEOS.



CARACTERISTICAS PRINCIPALES

VELOCIDAD

ASÍNCRONO Y CONTROLADO POR EVENTOS

SIN BÚFER

¿Cómo funciona Node.js?

EL DISEÑO DE NODE.JS ESTÁ INSPIRADO EN SISTEMAS COMO EL EVENT MACHINE DE RUBY O EL TWISTED DE PYTHON. SIN EMBARGO, NODE.JS PRESENTA UN BUCLE DE EVENTOS COMO UNA CONSTRUCCIÓN EN TIEMPO DE EJECUCIÓN EN LUGAR DE UNA BIBLIOTECA. ESTE BUCLE DE EVENTOS ES INVISIBLE PARA EL USUARIO.

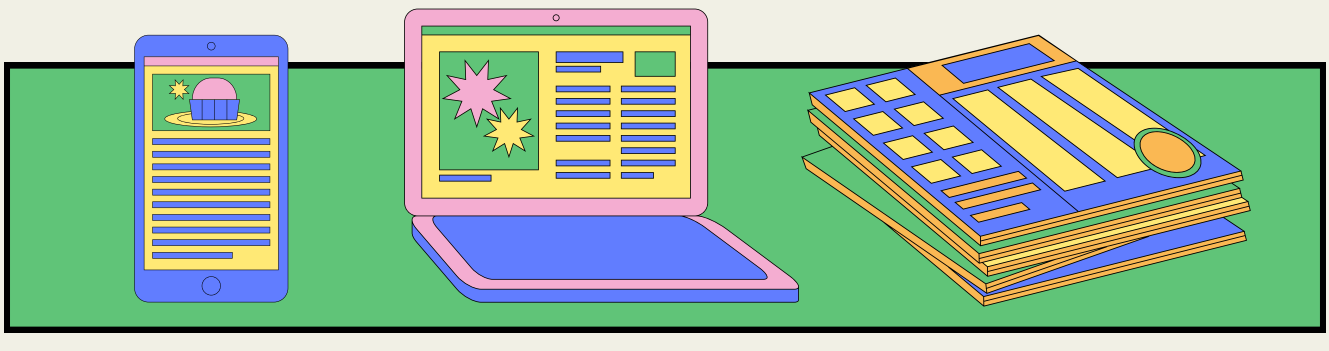
¿PARA QUE SIRVE NODE JS?

PUEDES UTILIZAR NODE.JS PARA DIFERENTES TIPOS DE APLICACIONES. LOS SIGUIENTES SON ALGUNOS DE LOS EJEMPLOS:

- APLICACIONES DE TRANSMISIÓN DE DATOS (STREAMING)
- APLICACIONES INTENSIVAS DE DATOS EN TIEMPO REAL
- APLICACIONES VINCULADAS A E/S
- APLICACIONES BASADAS EN JSON:API
- APLICACIONES DE PÁGINA ÚNICA

¿QUÍEN USA NODE.JS?

- **GODADDY**
- **MICROSOFT**
- **EBAY**
- **GENERAL ELECTRIC**
- **PAYPAL**
- **UBER**
- **NASA**
- **NETFLIX**
- **LINKEDIN**
- **MEDIUM**

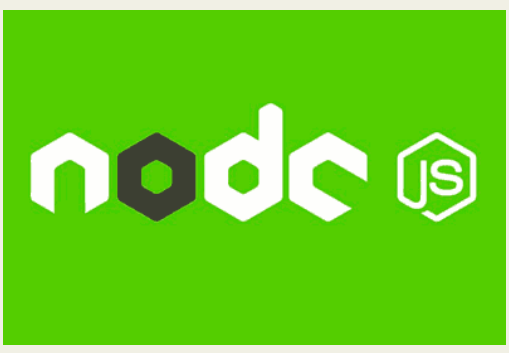


¿POR QUÉ UTILIZAR NODE.JS?

1. PORQUE PUEDE EJECUTARSE EN UNA VARIEDAD DE SERVIDORES, ENTRE LOS QUE DESTACAN MICROSOFT WINDOWS, MAC OS X Y UNIX.
2. PORQUE PLATAFORMAS COMO LINKEDIN, EBAY O PAYPAL FUERON CREADAS CON NODE.JS, LO QUE INDIRECTAMENTE ES UNA MUESTRA DE SU CALIDAD.
3. SU RENDIMIENTO HA SORPRENDIDO A PROGRAMADORES DE TODO EL MUNDO, PUES PERMITE CREAR TRABAJOS DE GRAN CALIDAD Y DISMINUYE EL MARGEN DE EXPERIMENTAR ERRORES TÉCNICOS.
4. SU PARECIDO CON JAVASCRIPT HACE QUE ESTE LENGUAJE SEA MÁS FÁCIL DE APRENDER.
5. NODE.JS ABRE TODO UN NUEVO MUNDO DE POSIBILIDADES PARA LOS PROGRAMADORES, A QUIENES PERMITIRÁ CREAR APLICACIONES ALTAMENTE ESCALABLES E INNOVADORAS, ESCRIBIENDO CÓDIGOS QUE PERMITAN DECENAS DE MILES DE CONEXIONES SIMULTÁNEAS EN UN ÚNICO SERVIDOR. HASTA EL MOMENTO LA MAYORÍA DE LOS PROGRAMAS DE SERVIDOR PERMITEN UN MÁXIMO DE APROXIMADAMENTE 4.000 USUARIOS CONECTADOS AL MISMO TIEMPO, POR LO QUE PARA AUMENTAR ESTA CIFRA LAS EMPRESAS DEBEN AGREGAR SERVIDORES, PROBLEMA QUE DESAPARECERÍA CON EL USO DE NODE.JS YA QUE ESTE PUEDE SOPORTAR DECENAS DE MILES DE CONEXIONES CONCURRENTES Y ASÍ DISMINUIR LOS COSTES DE INFRAESTRUCTURA.

¿CUÁNDO UTILIZAR NODE JS?

NODE.JS ES UNA BUENA SOLUCIÓN PARA REALIZAR TAREAS INTENSIVAS DE DATOS O ANÁLISIS EN TIEMPO REAL, YA QUE TIENE UNA ARQUITECTURA ASINCRÓNICA Y CARACTERÍSTICAS DE E/S SIN BLOQUEO. ALGUNOS CASOS DE USO POPULARES INCLUYEN:



POR EJEMPLO

CHAT EN TIEMPO REAL.

STREAMING DE DATOS.

APLICACIONES DE UNA SOLA PÁGINA

CÓMO CREAR UNA API UTILIZANDO NODE.JS

PRIMERO PASOS

- NPM CONFIGURADO Y GIT CONFIGURADO
- CREAR REPOSITORIO EN GITHUB
- CLONAR REPOSITORIO
- EJECUTAR NPM INIT
- CREAR LA ESTRUCTURA DE LA APLICACIÓN, POR EL MOMENTO UNA CARPETA APP DONDE IREMOS GUARDANDO EL CÓDIGO PROPIO DE LA APLICACIÓN

CONFIGURACIÓN DE ESLINT

SI TIENES LOS LINTERS PARA SUBLIME Y JS CONFIGURADOS, COMPRUEBA QUE LA CONSOLA ARROJA UN ERROR.
HAY QUE CONFIGURAR ESLINT PARA FORMATEAR NUESTRO CÓDIGO.

```
npm i -D eslint
```

PARA EJECUTARLO SE NECESITA EL PATH

./NODE_MODULES/.BIN/ESLINT --INIT
? HOW WOULD YOU LIKE TO CONFIGURE ESLINT? ANSWER QUESTIONS ABOUT YOUR STYLE
? ARE YOU USING ECMASCRIPT 6 FEATURES? NO
? WHERE WILL YOUR CODE RUN? NODE
? DO YOU USE JSX? NO
? WHAT STYLE OF INDENTATION DO YOU USE? SPACES
? WHAT QUOTES DO YOU USE FOR STRINGS? SINGLE
? WHAT LINE ENDINGS DO YOU USE? UNIX
? DO YOU REQUIRE SEMICOLONS? NO
? WHAT FORMAT DO YOU WANT YOUR CONFIG FILE TO BE IN? JSON
SUCCESSFULLY CREATED .ESLINTRC.JSON FILE IN /HOME/JUANDA/API_NODE_EXPRESS/EJERCICIO3-NODEMON-ESLINT

EXPRESS
UTILIZAREMOS EXPRESS PARA REALIZAR LA API
INSTALAR EXPRESS MEDIANTE UNO DE LOS COMANDOS SIGUIENTES:

```
npm install --save express

npm i -S express
```

CREAMOS EL FICHERO APP/SERVER.JS DONDE PONDREMOS EL CÓDIGO NECESARIO PARA TESTEAR UNA API MUY BÁSICA PARA PROBAR EXPRESS.
UTILIZA EL PLUGIN EXPRESSCOMPLETE (AGET, APUT...) DE SUBLIME PARA AUTOCOMPLETADO

```
var express = require('express') //llamamos a Express
var app = express()

var port = process.env.PORT || 8080  // establecemos nuestro puerto

app.get('/', function(req, res) {
  res.json({ mensaje: '¡Hola Mundo!' })
})

app.get('/cervezas', function(req, res) {
  res.json({ mensaje: '¡A beber cerveza!' })
})

app.post('/', function(req, res) {
  res.json({ mensaje: 'Método post' })
})

app.del('/', function(req, res) {
  res.json({ mensaje: 'Método delete' })
})

// iniciamos nuestro servidor
app.listen(port)
console.log('API escuchando en el puerto ' + port)
```

CONFIGURACIÓN DE UN SERVIDOR EXPRESO
VAMOS A CREAR UN NUEVO ARCHIVO APP.JS, QUE SERÁ EL PUNTO DE ENTRADA A NUESTRO PROYECTO ACTUAL. AL IGUAL QUE CON EL SERVIDOR DE HTTP ORIGINAL, NOS REQUIEREN UN MÓDULO Y CONFIGURAR UN PUERTO PARA EMPEZAR.

CREAR UN ARCHIVO APP.JS Y PONER EL SIGUIENTE CÓDIGO EN ÉL.

```
// Require packages and set the port
const express = require('express');
const port = 3002;
const app = express();
```

```
app.get('/', (request, response) => {
    console.log(`URL: ${request.url}`);
    response.send('Hello, Server!');
});
```

```
// Start the server
const server = app.listen(port, (error) => {
    if (error) return console.log(`Error: ${error}`);

    console.log(`Server listening on port ${server.address().port}`);
});
```

PODEMOS INICIAR EL SERVIDOR CON NODE APP.JS COMO HICIMOS ANTES, PERO TAMBIÉN PODEMOS MODIFICAR LA PROPIEDAD DE SCRIPTS EN NUESTRO ARCHIVO PACKAGE.JSON EJECUTAR AUTOMÁTICAMENTE ESTE COMANDO ESPECÍFICO.

```
"scripts": {
  "start": "node app.js"
},
```

AHORA PODEMOS USAR NPM START PARA INICIAR EL SERVIDOR, Y VAMOS A VER NUESTRO MENSAJE DE SERVIDOR EN EL TERMINAL.

```
Server listening on port 3002
```

Si ejecuto un [illegible] en la URL, vemos a ver que es gestionado por express ahora y hay algunos