



Recuerda marcar tu asistencia

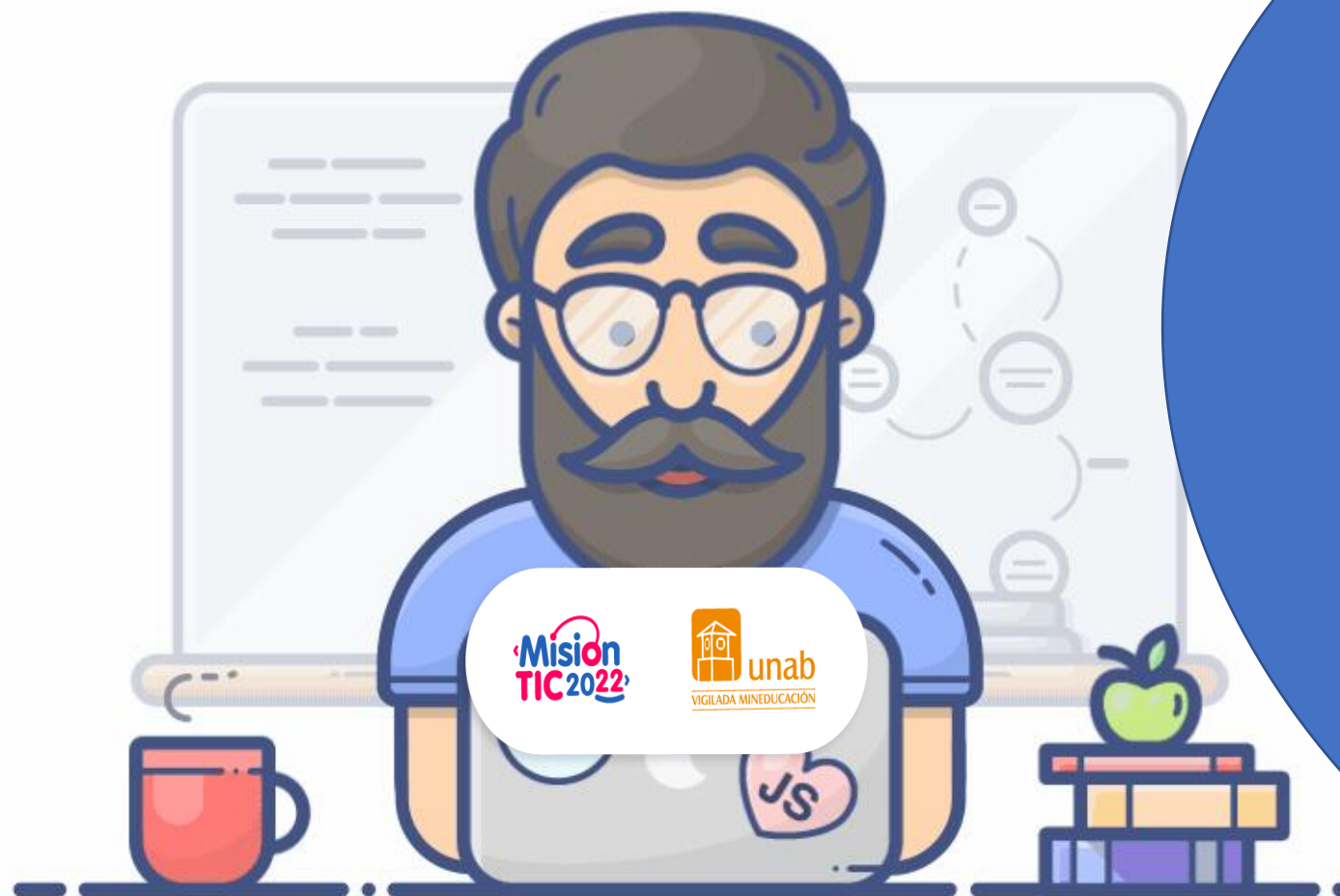


Asistencia en LMS



Semana 2

Node JS



Ing Edwin Garcia



El futuro digital
es de todos

MinTIC

Hechos

QUE

CONECTAN

Misión
TIC 2022

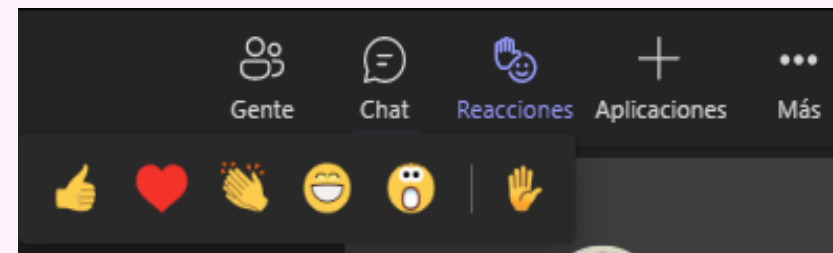
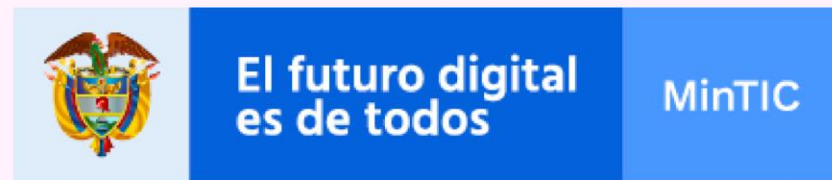
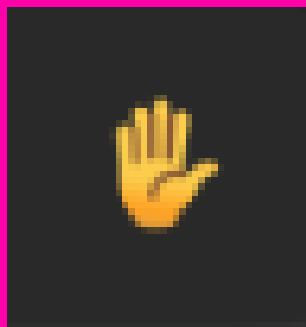
unab
VIGILADA MINEDUCACIÓN



Recuerden marcar su asistencia

Asistencia en LMS

**Pedir la
palabra para
realizar una
pregunta o
una
intervención**





La creciente popularidad de JavaScript ha traído consigo varios cambios, incluyendo la superficie del desarrollo web, ya que hoy en día es radicalmente diferente. Las cosas que podemos hacer en la web hoy, con JavaScript ejecutando en el servidor, como también en el navegador, eran difíciles de imaginar hace varios años



Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web.





Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables.



- En el siguiente ejemplo de "hola mundo", 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.



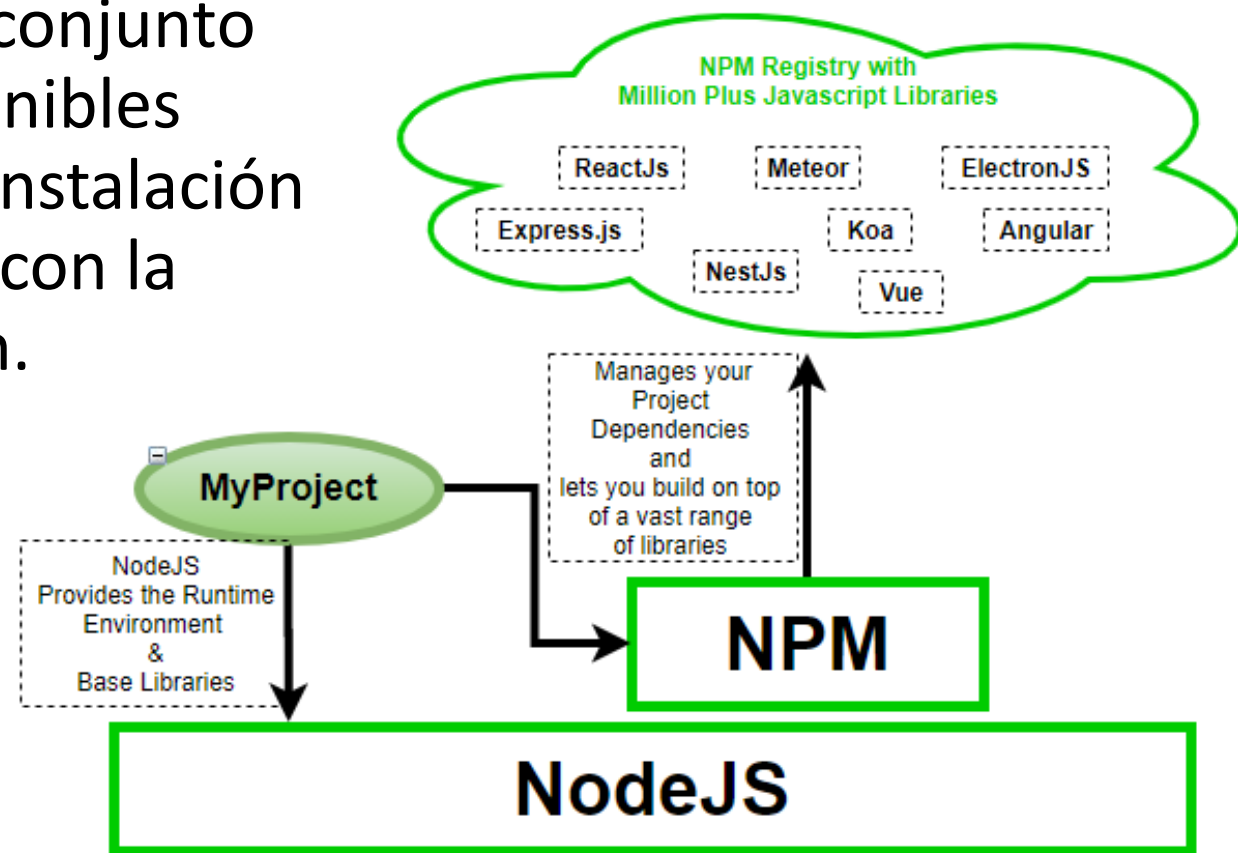
NPM: El Node Package Manager

Cuando hablamos de Node.js, una cosa que definitivamente no debe omitirse es integrarlo en el apoyo de la gestión de paquetes utilizando la herramienta NPM que viene por defecto con cada instalación de Node.js.





La idea de los módulos NPM es un conjunto de componentes reutilizables disponibles públicamente a través de una fácil instalación a través de un repositorio en línea, con la versión y la dependencia de gestión.





Requerimientos

- HTML
- Javascript



Funcionan sobre Node Js



Instalación

Descargar para Windows (x64)

16.17.0 LTS

Recomendado para la mayoría

18.9.0 Actual

Últimas características

[Otras Descargas](#) | [Cambios](#) | [Documentación de la API](#)

[Otras Descargas](#) | [Cambios](#) | [Documentación de la API](#)

```
C:\Users\LENOVO>node --version  
v14.15.4  
  
C:\Users\LENOVO>_
```




```
C:\Users\LENOVO>node -e "console.log('hola Mundo')  
hola Mundo  
C:\Users\LENOVO>_
```



Desarrollo de Aplicaciones Web



El futuro digital
es de todos

MinTIC

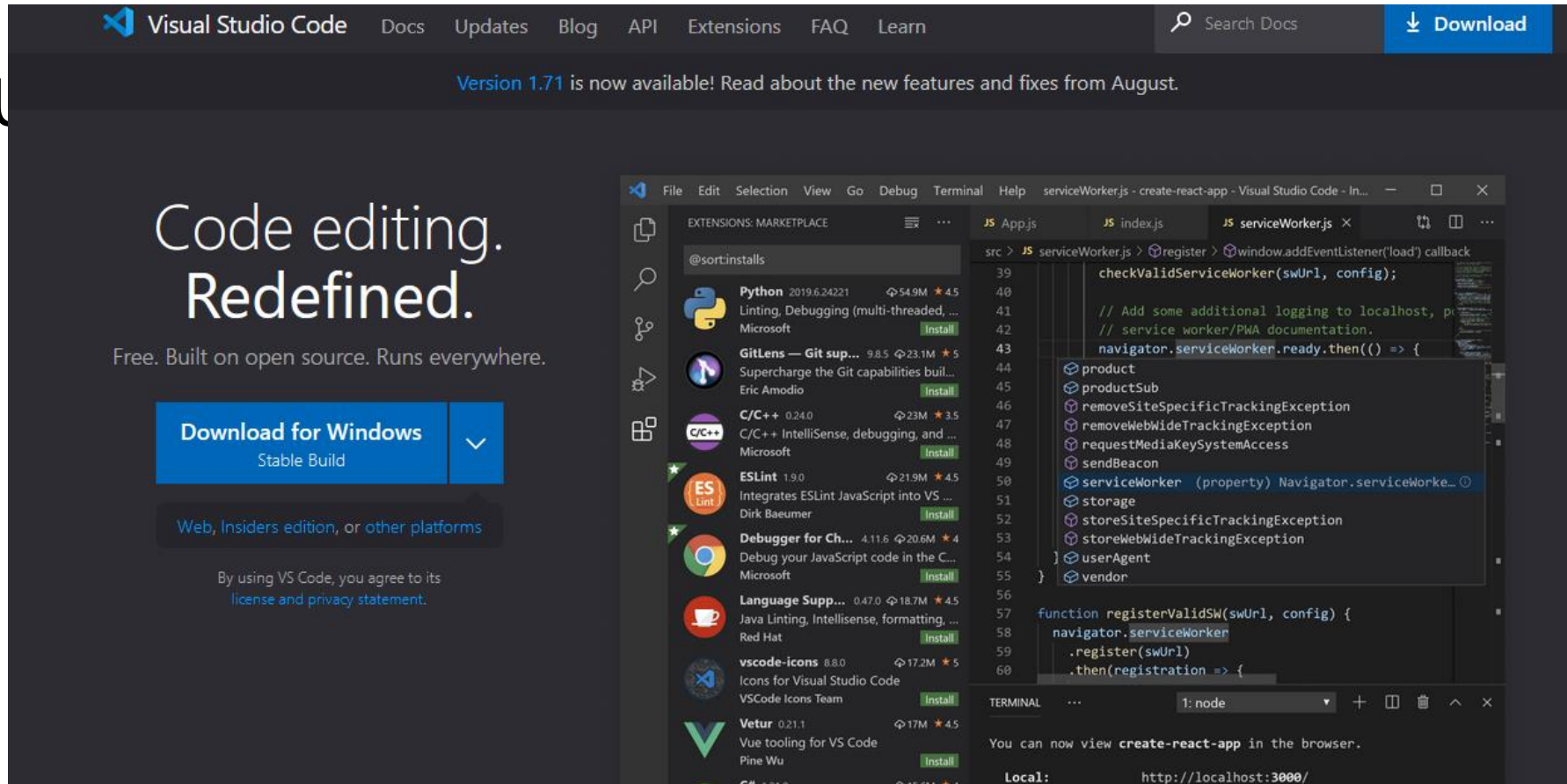
```
C:\Users\LENOVO>node
Welcome to Node.js v14.15.4.
Type ".help" for more information.
> var nombre = "edwin"
undefined
> console.log(nombre)
edwin
undefined
>
```

Mision
TIC 2022



Hechos
QUE CONECTAN ✓

Visu



Visual Studio Code

Docs Updates Blog API Extensions FAQ Learn

Search Docs Download

Version 1.71 is now available! Read about the new features and fixes from August.

Code editing. Redefined.

Free. Built on open source. Runs everywhere.

[Download for Windows](#)
Stable Build

[Web, Insiders edition, or other platforms](#)

By using VS Code, you agree to its [license and privacy statement](#).

EXTENSIONS: MARKETPLACE

@sortinstalls

- Python** 2019.6.24221 54.9M 4.5
Linting, Debugging (multi-threaded, ...
Microsoft [Install](#)
- GitLens — Git sup...** 9.8.5 23.1M 5
Supercharge the Git capabilities buil...
Eric Amodio [Install](#)
- C/C++** 0.24.0 23M 3.5
C/C++ + IntelliSense, debugging, and ...
Microsoft [Install](#)
- ESLint** 1.9.0 21.9M 4.5
Integrates ESLint JavaScript into VS ...
Dirk Baumer [Install](#)
- Debugger for Ch...** 4.11.6 20.6M 4
Debug your JavaScript code in the C...
Microsoft [Install](#)
- Language Supp...** 0.47.0 18.7M 4.5
Java Linting, IntelliSense, formatting, ...
Red Hat [Install](#)
- vscode-icons** 8.8.0 17.2M 5
Icons for Visual Studio Code
VSCode Icons Team [Install](#)
- Vetur** 0.21.1 17M 4.5
Vue tooling for VS Code
Pine Wu [Install](#)
- C#** 1.21.0 15.6M 4.4
C# IntelliSense, debugging, and ...
Microsoft [Install](#)

JS App.js JS index.js JS serviceWorker.js

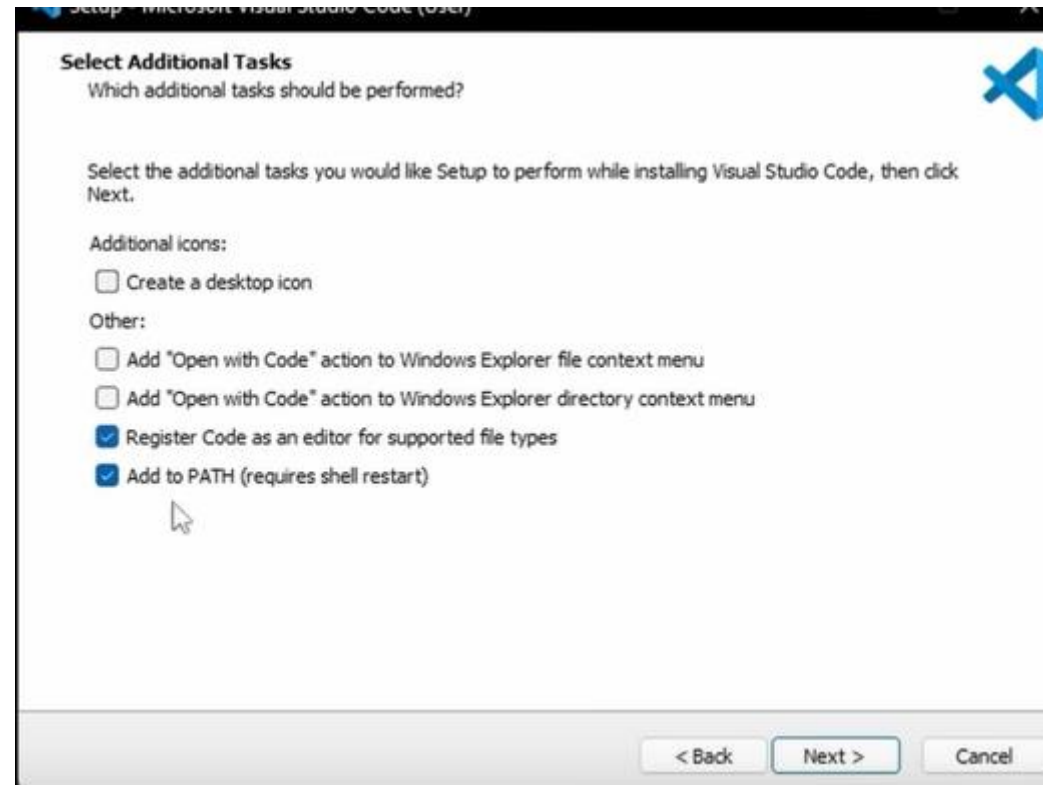
```
src > JS serviceWorker.js > register > window.addEventListener('load') callback
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
function registerValidSW(swUrl, config) {
  navigator.serviceWorker
    .register(swUrl)
    .then(registration => {
```

product
productSub
removeSiteSpecificTrackingException
removeWebWideTrackingException
requestMediaKeySystemAccess
sendBeacon
serviceWorker (property) Navigator.serviceWorke...
storage
storeSiteSpecificTrackingException
storeWebWideTrackingException
userAgent
vendor

TERMINAL 1: node

You can now view create-react-app in the browser.

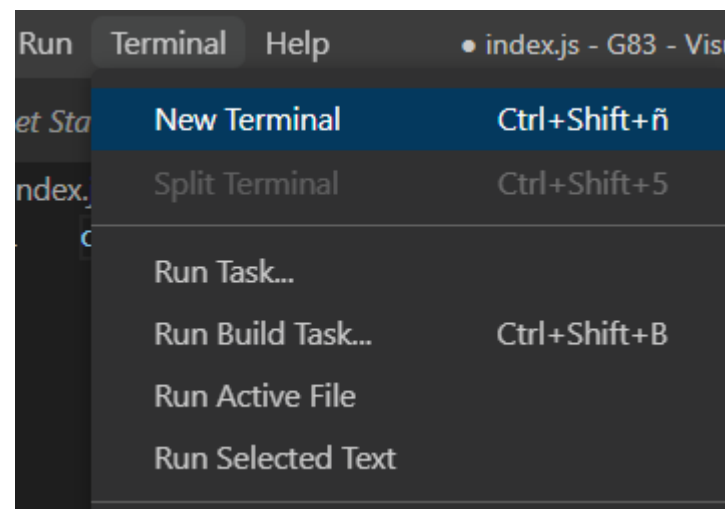
Local: http://localhost:3000/





G83

```
JS index.js
```



Desarrollo de Aplicaciones Web



El futuro digital
es de todos

MinTIC

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER  > po

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\LENOVO\Desktop\node\G83> node index.js
PS C:\Users\LENOVO\Desktop\node\G83> node index.js
hola mundo
PS C:\Users\LENOVO\Desktop\node\G83> 
```



Hechos
QUE CONECTAN



Var - Let - Const



Si declaramos la variable con **var** también dentro de la función, se crea una variable local cuya visibilidad se reduce a la propia función donde ha sido declarada



Si una variable es declarada con **let** en el ámbito global o en el de una función, la variable pertenecerá al ámbito global o al ámbito de la función respectivamente, de forma similar a como ocurría con **var**.

Pero si declaramos una variable con **let** dentro un bloque que a su vez está dentro de una función, la variable pertenece solo a ese bloque



El ámbito o scope para una variable declarada con **const** es, al igual que con **let**, el bloque, pero si la declaración con **let** previene la sobreescritura de variables, **const** directamente prohíbe la reasignación de valores.



Ejemplos variables

Ejemplos if else



Función flecha

Una **expresión de función flecha** es una alternativa compacta a una expresión de función tradicional, pero es limitada y no se puede utilizar en todas las situaciones.



Diferencias y limitaciones:

- No tiene sus propios enlaces a `this` o `super` y no se debe usar como [métodos](#).
- No tiene argumentos o palabras clave `new.target`.
- No apta para los métodos `call`, `apply` y `bind`, que generalmente se basan en establecer un [ámbito o alcance](#).
- No se puede utilizar como [constructor](#).
- No se puede utilizar `yield` dentro de su cuerpo.



// Función tradicional

```
function (a){ return a + 100; }
```

// Desglose de la función flecha

// 1. Elimina la palabra "function" y coloca la flecha entre el argumento y el corchete de apertura.

```
(a) => { return a + 100; }
```

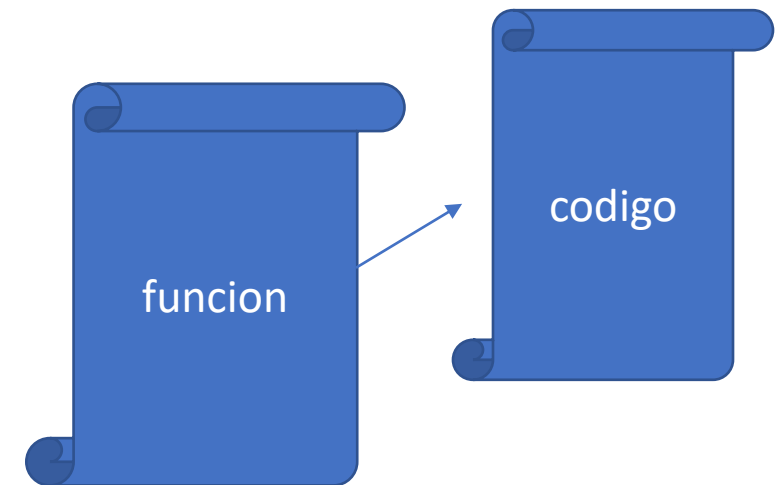


```
// Función tradicional  
function (a, b){  
  let chuck = 42;  
  return a + b + chuck;  
}  
  
// Función flecha  
(a, b) => {  
  let chuck = 42;  
  return a + b + chuck;  
}
```




Exports

La declaración **export** se utiliza al crear módulos de JavaScript para exportar funciones, objetos o tipos de dato primitivos del módulo para que puedan ser utilizados por otros programas con la sentencia [import](#).





Desarrollo de Aplicaciones Web



El futuro digital
es de todos

MinTIC

```
const ciclo = "Desarrollo Aplicaciones Web";  
  
module.exports= ciclo;  
  
console.log(module);|
```





```
index.js > ...  
1  //require('.modulos/datos')  
2  
3  const valor = require('.modulos/datos')  
4  
5  console.log(valor)
```



Desarrollo de Aplicaciones Web



El futuro digital
es de todos

MinTIC

```
modulos > JS datos.js > ...  
1  const ciclo = "Desarrollo Aplicaciones Web"  
2  
3  const objeto = {  
4      id : 1,  
5      equipo : "Equipo 1"  
6  }  
7  
8  
9  module.exports= objeto;  
***
```





Desarrollo de Aplicaciones Web



El futuro digital
es de todos

MinTIC

```
ulos > JS datos.js > [?] datos > 🔑 objeto
const ciclo = "Desarrollo Aplicaciones Web"

const objeto = {
  id : 1,
  equipo : "Equipo 1"
}

const datos = {
  ciclo,
  objeto
}

module.exports= datos;
...
```

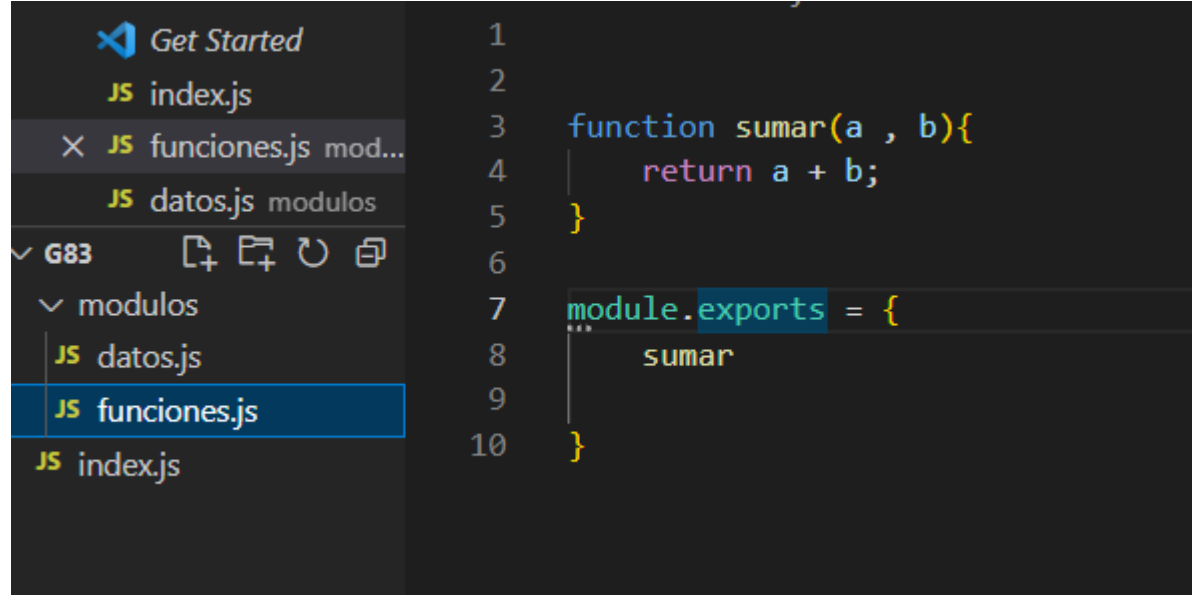
```
PS C:\Users\LENOVO\Desktop\node\G83> node index
{
  ciclo: 'Desarrollo Aplicaciones Web',
  objeto: { id: 1, equipo: 'Equipo 1' }
}
```



```
//require('.modulos/datos')  
  
const {objeto} = require('./modulos/datos')  
  
console.log(objeto)
```

```
PS C:\Users\LENOVO\Desktop\node\G83> node index  
{ id: 1, equipo: 'Equipo 1' }  
PS C:\Users\LENOVO\Desktop\node\G83> █
```


Podemos también exportar funciones...



```
1
2
3 function sumar(a , b){
4     return a + b;
5 }
6
7 module.exports = {
8     sumar
9 }
10
```

```
6
7 const fun = require('./modulos/funciones');
8
9 console.log(fun.sumar(10,20))
```



Leer Archivos

Del las siglas fs (fileSystem), al requerir fs , nos permite cargar desde nuestro código, archivos para ser leídos internamente.

```
const fs = require('fs')  
  
const arch = fs.readFileSync('./modulos/archivo1.js', 'utf-8')  
console.log(arch);
```

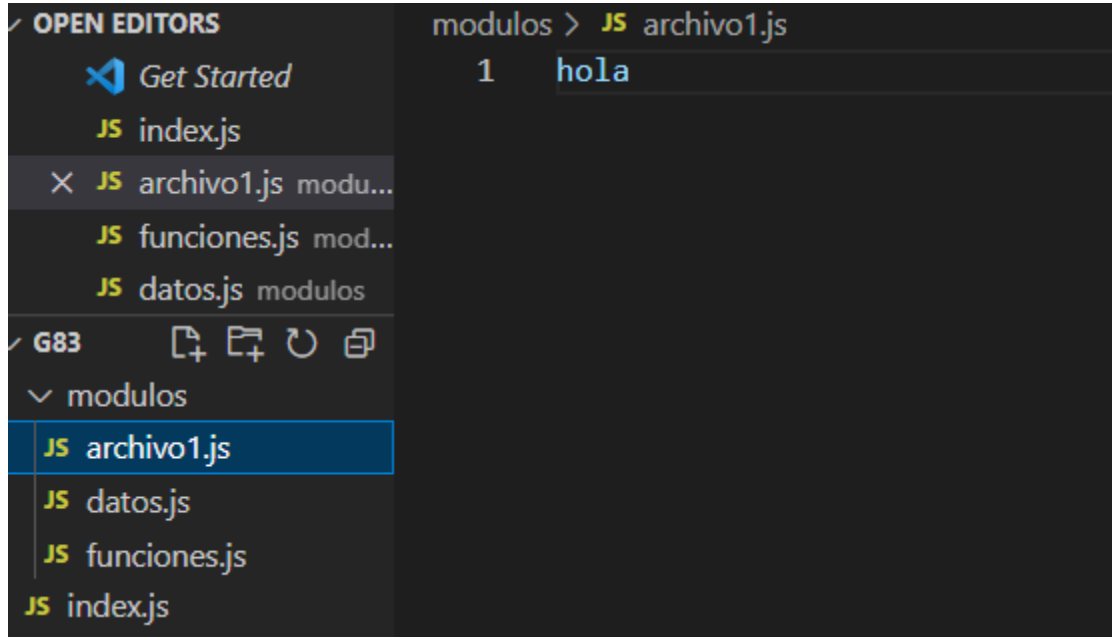


Desarrollo de Aplicaciones Web



El futuro digital
es de todos

MinTIC



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

const arch = fs.readFileSync('./modulos/archivo1.js', 'utf-8')
console.log(arch);
```

```
PS C:\Users\LENOVO\Desktop\node\G83> node index
hola
```

Crear un archivo

```
const fs = require('fs')  
  
// const arch = fs.readFileSync('./modulos/archivo1.js', 'utf-8')  
// console.log(arch);  
💡  
fs.writeFile('./modulos/archivo2.js', 'Texto de ejemplo', (err, data) => console.log(err))
```

`\n` Salto linea



Desarrollo de Aplicaciones Web

Http



El futuro digital
es de todos

MinTIC

Misión
TIC 2022



Hechos
QUE CONECTAN ✓



- [https://es.wikipedia.org/wiki/V8_\(int%C3%A9rprete_de_JavaScript\)](https://es.wikipedia.org/wiki/V8_(int%C3%A9rprete_de_JavaScript))
- <https://nodejs.org/es/about/>
- <https://www.toptal.com/nodejs/por-que-demonios-usaria-node-js-un-tutorial-caso-por-caso>

Bibliografía





El futuro digital
es de todos

MinTIC

**Practicar los
temas tratados**

Mision
TIC2022



Hechos
QUE CONECTAN

Nos vemos en la siguiente sesión



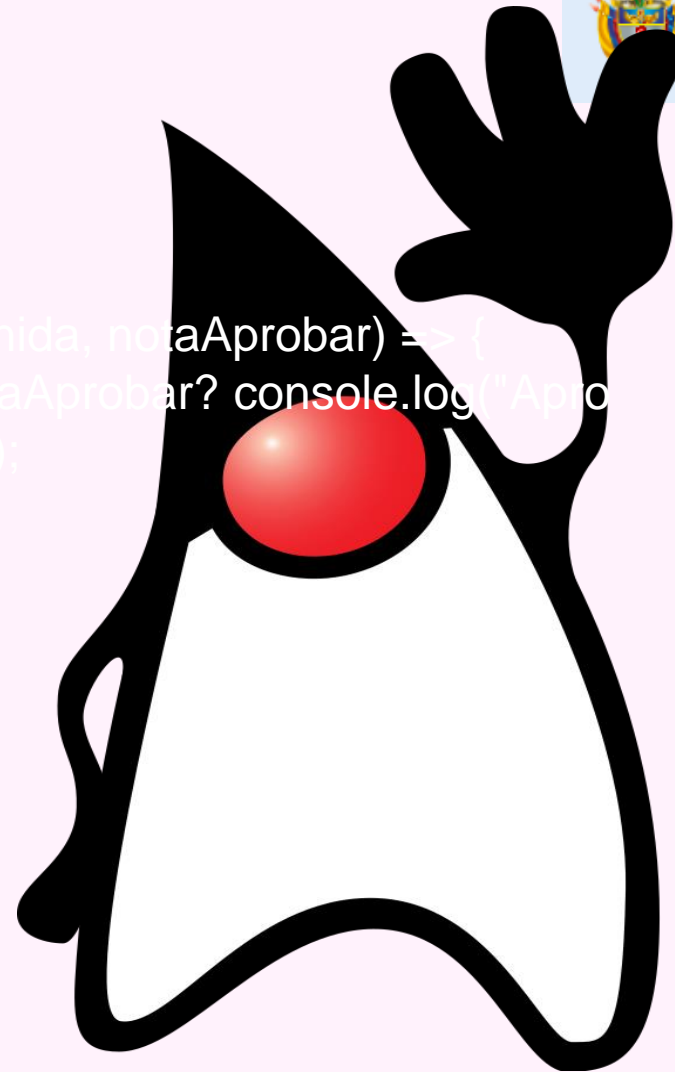
El futuro digital
es de todos

MinTIC

**Muchas
Gracias**

```
const promedio = (notaObtenida, notaAprobar) => {  
  return notaObtenida >= notaAprobar? console.log("Aprobo") : console.log("Recurso");  
}
```

```
promedio(4, 3)
```



www.mintic.gov.co

Misión
TIC2022

unab
VIGILADA MINEDUCACIÓN