

wawiwa

NODE.JS

Introducción

Sobre node.js:

- Gratis
- Marco de código abierto
- Se ejecuta en windows, linux, mac y more
- Vamos a escribir en JavaScript

Introducción

Usando node.js podemos:

- Crear páginas web dinámicas
- Abrir, cerrar, borrar y editar archivos en el servidor
- Recolectar datos de formularios
- Editar, crear, agregar y borrar datos de bases de datos

CAPÍTULO 1 - INTRODUCCIÓN E INSTALACIÓN

Instalación

Navega a nodejs.org e instala node.js

<https://nodejs.org/en/download/>

También necesitamos un editor de código. En este curso usaremos la plataforma Visual Studio Code.

<https://code.visualstudio.com/download>



wawiwa

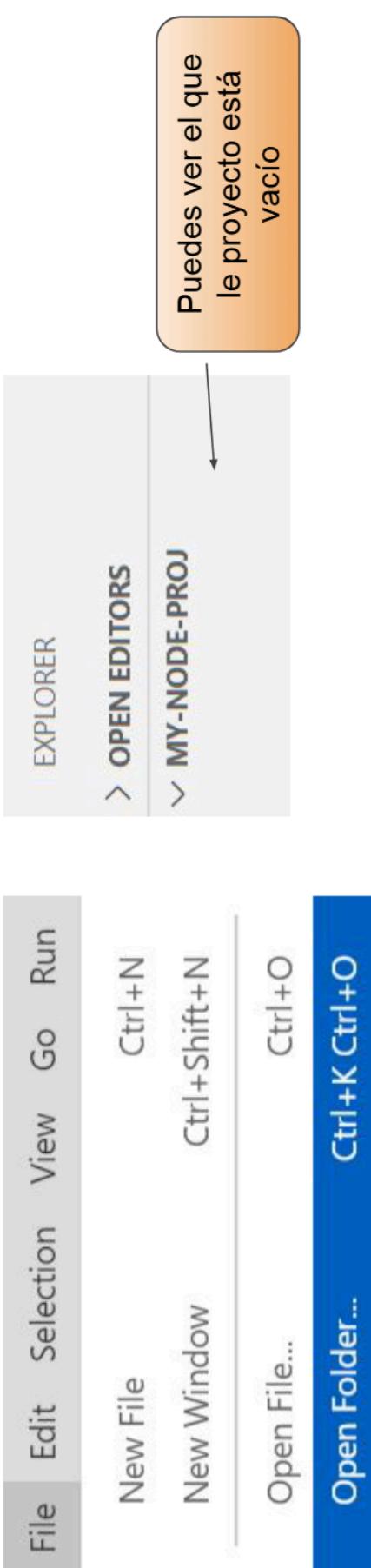
¿Preguntas?

CAPÍTULO 2 - PRIMER PROGRAMA

Primer Programa

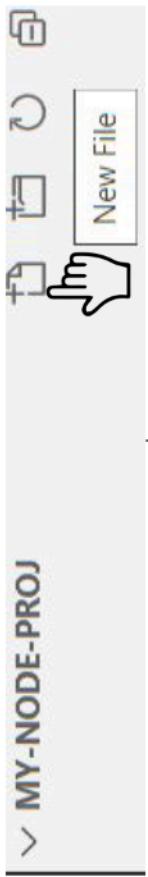
Crea una carpeta vacía en tu computadora, que después contendrá tu proyecto.

En VS Code - ve a **file** -> **open folder** -> **elige la carpeta que creaste**



Primer Programa

Crea un nuevo archivo js:



↗ MY-NODE-PROJ
JS a1.js

The extension
indicates the file type

Primer Programa

Empecemos con imprimir "hello world":

```
console.log("Hello World");
```

Presiona Ctrl + s para guardar, o ve a **file -> auto save**, para guardar automáticamente cada vez que editas uno de tus archivos.

Para correr el proyecto ve a **terminal -> new terminal**.

```
PS C:\Users\X5-i7\OneDrive\Desktop\Programming\Asaf\my-node-proj> node a1
Hello World
```

En la terminal ingresa "node a1", donde a1 es el nombre del archivo.

Primer Module

In node we can create a module, which contains functions and exports them to use.

Create another js file:

```
MY-NODE-PROJ
  JS  a1.js
  JS  mymodule.js
```

Primer Módulo

En "mymodule.js" escribiremos las funciones y las exportaremos:

```
exports.plus = function(x, y) {  
    return x + y;  
}  
  
exports.minus = function(x, y) {  
    return x - y;  
}
```

module.exports es un objeto especial que está incluido en todos los archivos JavaScript en las aplicaciones Node.js por defecto. El módulo es una variable que representa el módulo y exportaciones actuales como un objeto que va a ser expuesto como un módulo. Entonces, cualquier cosa que asigne a module.exports será expuesta como un módulo.

Primer Módulo

Usemos las funciones del módulo en otro archivo llamado "program.js":

En program.js:

```
const computeModule = require('./mymodule.js');

const sum = computeModule.plus(5, 6);
console.log(`sum is ${sum}`);

const sub = computeModule.minus(5, 6);
console.log(`sub is ${sub}`);
```

Una variable que contiene 'mymodule'

```
PS C:\Users\X5-i7\OneDrive\Desktop\Programming\Asaf\my-node-proj> node program
sum is 11
sub is -1
```

Hazlo tú mismo 1

Exporta un módulo “Name”. Tiene 2 funciones:

1. Una función que recibe **first** y **last name** y lo retorna con "hello".
2. Una función que recibe **sender name**, **message** y **subject** y los imprime.

Por ejemplo: "Jack", "Hello, how are you?", "Dan", el mensaje
retornado es: "**Dan, you got a new message from Jack: Hello, how are**
you?"

Crea un programa y usalo.



wawiwa

¿Preguntas?