



# DESARROLLO DE APLICACIONES WEB AVANZADO

## Laboratorio N° 01 Instalacion de NodeJs



**DOCENTE**  
Jorge Castañeda



**CURSO**  
Desarrollo de Aplicaciones Web Avanzado

## **Laboratorio 01: Instalación de la plataforma node.js**

**ALUMNO: LENNARD IDONE**

### **Objetivos:**

Al finalizar el laboratorio el estudiante será capaz de:

- Identificar las principales características del node.js
- Instalar node.js
- Identificar las características de utilizar JavaScript en el lado del servidor.

### **Seguridad:**

- Ubicar maletines y/o mochilas en el gabinete del aula de Laboratorio.
- No ingresar con líquidos, ni comida al aula de Laboratorio.
- Al culminar la sesión de laboratorio apagar correctamente la computadora y la pantalla, y ordenar las sillas utilizadas.

### **Equipos y Materiales:**

- Una computadora con:
  - Windows 7 o superior
  - Conexión a la red del laboratorio
- Máquinas virtuales:
  - Windows 7 Pro 64bits Español - Plantilla
- Instalador de node.js

### **Procedimiento:**

#### **Instalación de node.js**

##### **1. Obtención del instalador**

1.1.El instalador de node.js se puede obtener de la página oficial:

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

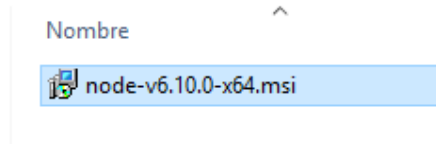
1.2.Veremos varias opciones, aparte de las plataformas de sistema operativo como son:

- **LTS** (recomendada para la mayoría de usuarios): es la versión de Node.js con Long Term Support (LTS), es decir a la que se le da soporte a largo plazo. Esta versión puede no tener disponibles las últimas tecnologías que todavía no se consideran estables.
- **Current**: esta es la versión más reciente de Node.js e incluye todas las funcionalidades, incluso aquellas más novedosas y que no se consideran estables.

1.3. Por norma general, **seleccionar la versión LTS es la opción recomendada y estable.**

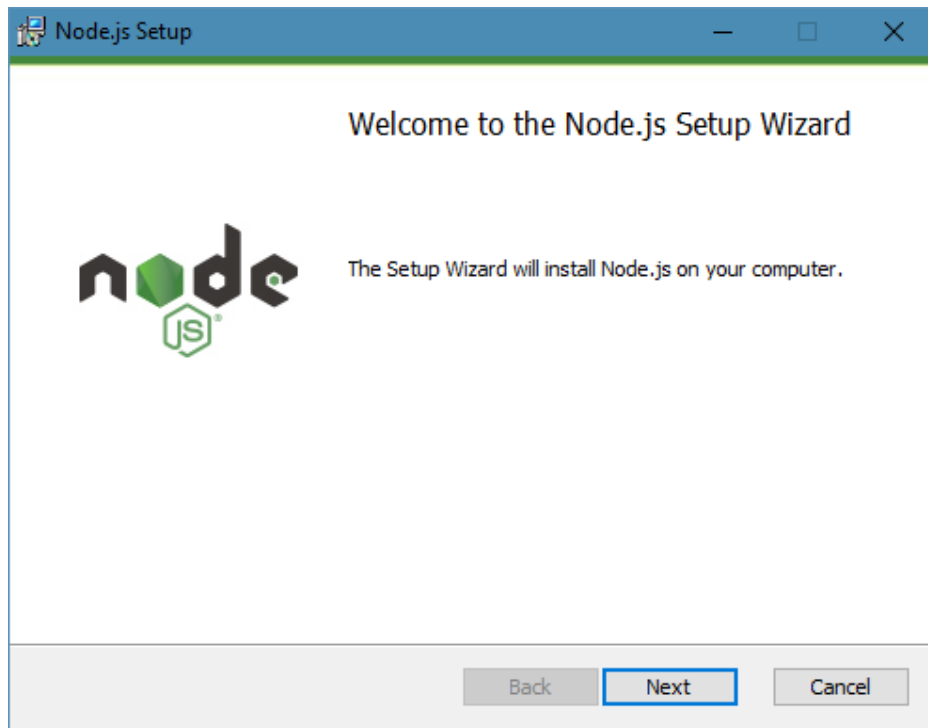
## 2. Proceso de instalación de node.js

2.1. Solicite al instructor, el archivo de instalación de node.js

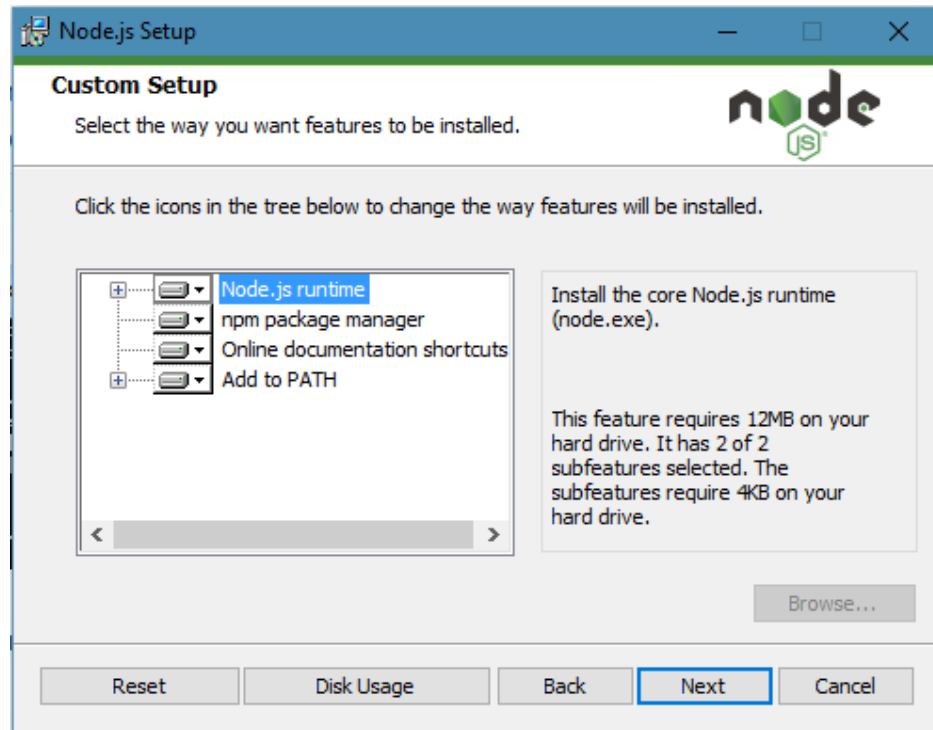


2.2. Copie el archivo instalador al escritorio del equipo virtual

2.3. Doble clic en el instalador para iniciar el proceso de instalación. Haga click en Next

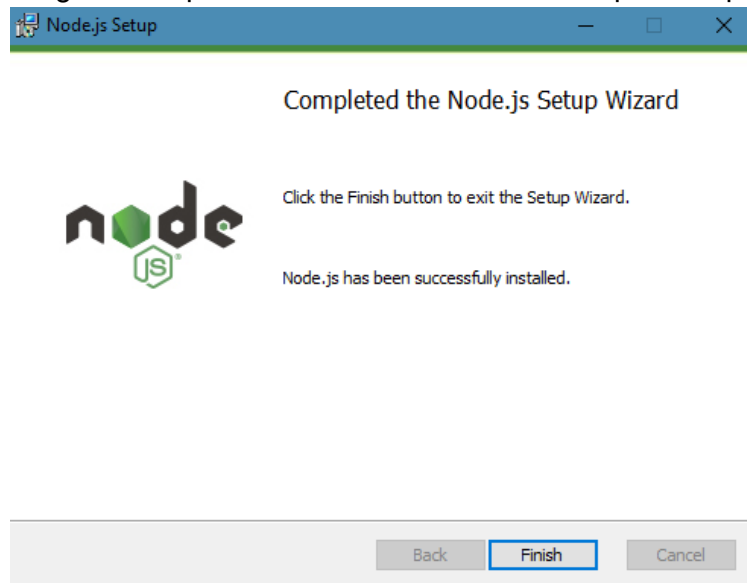


2.4. No deberá tener problemas en el instalador ya que se seleccionarán las opciones por defecto. Llegará usted a la siguiente pantalla:



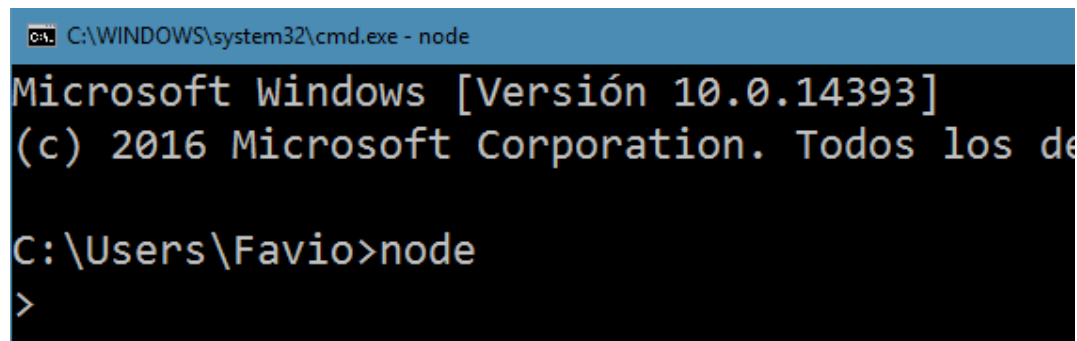
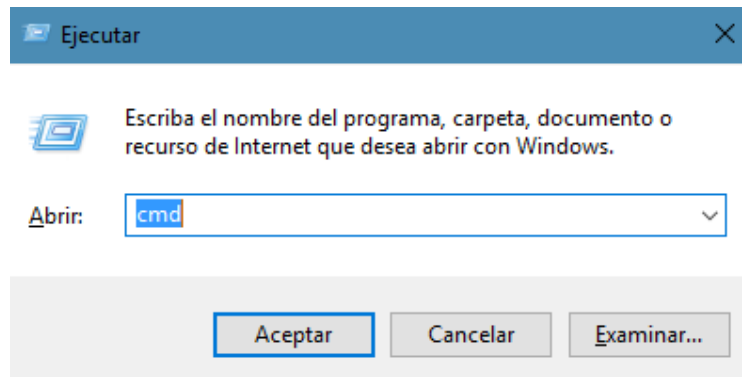
Haga click en cada una de las características a instalar e indique que es lo que realizarán al ser instaladas:

2.5. Siga con el proceso de instalación con las opciones por defecto.



### 3. Verificación de la instalación

3.1. Abra una consola de comandos y ejecutemos el comando **node**



#### 4. Hola mundo con node.js

4.1. Cree el archivo **hola.js** en la unidad **C:** con el siguiente contenido:

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

var manejador = function(solicitud, respuesta){
    console.log('Hola mundo!');
};

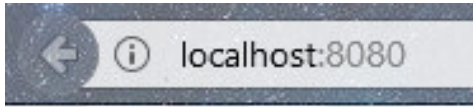
var servidor = http.createServer(manejador);

servidor.listen(8080);
```

4.2. En una ventana de comandos sitúese en la unidad **C:\** y escriba el siguiente comando:

```
>node hola.js
```

4.3. Abra un navegador de Internet, y escriba la siguiente URL:



- 4.4. Verificamos en la consola el resultado.
- 4.5. Anote sus conclusiones de lo que se acaba de realizar. Para finalizar la actividad del servidor, debe presionar **Ctrl + C**
- 4.6. A continuación, cree el archivo **hola.js**

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

var manejador = function(solicitud, respuesta){
  console.log('Conexion entrante');
  respuesta.end('Hola mundo!');
};

var servidor = http.createServer(manejador);

servidor.listen(8080);
```

- 4.7. Una vez más, ejecute el comando:

```
>node hola.js
```

- 4.8. Anote sus observaciones y mencione en que se diferencia de la versión anterior del archivo.
- QUE UNO EJECUTA EN EL SERVIDO Y OTRO EJECUTA EN EL NAVEGADOR

## 5. Leer archivo de servidor

- 5.1. Creamos el archivo **index.html** con el siguiente contenido:

```
<html>
<head>
  <title>Pagina de ejemplo</title>
</head>
<body>
  <h1>Hola mundo con HTML!</h1>
</body>
</html>
```

5.2. Creamos el archivo **web.js** con el siguiente contenido

```
var http = require('http'),
    fs = require('fs');

var html = fs.readFileSync('./index.html');

http.createServer(function(solicitud, respuesta){
    respuesta.write(html);
    respuesta.end();
}).listen(8080);
```

5.3. Ejecútelo y anote sus observaciones.

5.4. A continuación, modificaremos el archivo para que luzca de la siguiente manera.

```
var http = require('http'),
    fs = require('fs');

http.createServer(function(solicitud, respuesta){
    fs.readFile('./index.html', function(error, html){
        respuesta.write(html);
        respuesta.end();
    });
}).listen(8080);
```

5.5. Busque en Internet la diferencia entre los comandos **readFileSync** y **readFile** y explique cuál es la diferencia entre ambos.

UNO EJECUTA DE MANERA ASINCRONA Y OTRO DE MANERA SINCRONA.

6. **Finalizar la sesión**

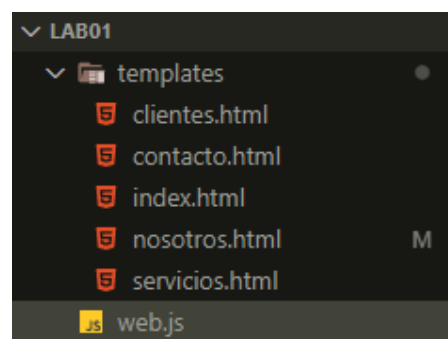


**Tarea:**

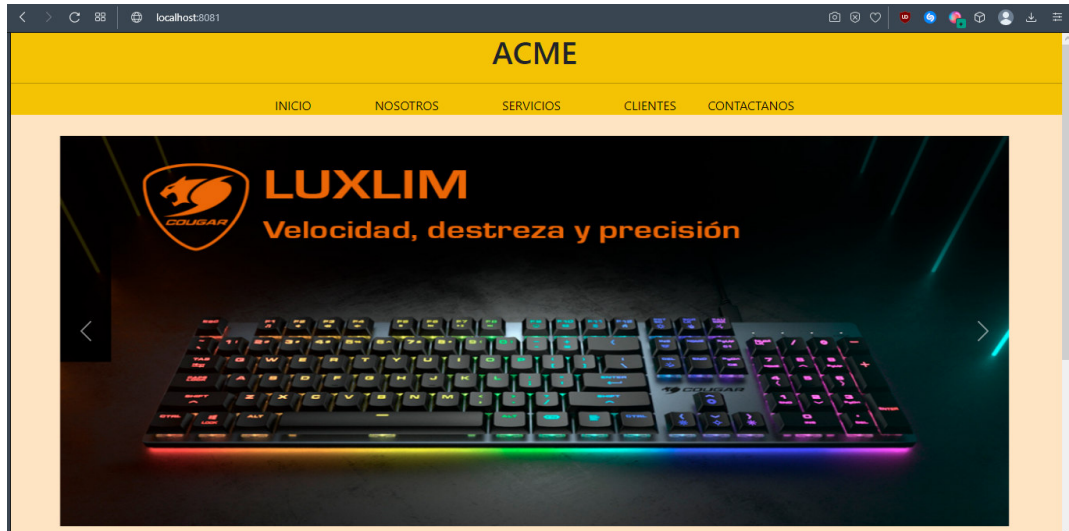
1. Desarrolle una página web para la empresa ACME (solicite el rubro de la empresa a su docente), con sus páginas básicas: Inicio, Nosotros, Nuestros Servicios, Catálogo de Clientes y Contáctenos. Esta última página deberá tener un formulario de contacto. Toda la web debe utilizar **node.js**

Enlace del GITHUB: <https://github.com/Lesandro/DAWAV-LAB1>

```
.js web.js > http.createServer() callback
1  var http = require('http');
2  fs = require('fs');
3
4  http.createServer(function(solicitud,respuesta){
5
6      const { url } = solicitud;
7      if (url === '/') {
8          template='index.html'
9      }
10     else if(url === '/nosotros') {
11         template='nosotros.html'
12     }
13     else if(url === '/servicios') {
14         template='servicios.html'
15     }
16     else if(url === '/clientes') {
17         template='nosotros.html'
18     }
19     else if(url === '/contacto') {
20         template='nosotros.html'
21     }
22
23     fs.readFile('templates/'+template,function(error,html) {
24         respuesta.write(html);
25         respuesta.end();
26     });
27     }).listen(8081);
28
```







### **Conclusiones:**

Indicar las conclusiones que llegó después de los temas tratados de manera práctica en este laboratorio.

- NodeJs nos permite programar de lado del servidor y así poder ejecutar mucho más rápido nuestras aplicaciones.
- NodeJs nos da la posibilidad de poder escoger el puerto en el cual deseamos ejecutar nuestro proyecto.
- Es rápido y eficiente gracias a que NodeJs está construido sobre el motor JavaScript V8 de Google, que es altamente eficiente en términos de rendimiento.