

Laboratorio III

Servidor Node.js (parte 1)

Clase 4

Maximiliano Neiner

Temas a Tratar

- Servidor Node.js

Temas a Tratar

- Servidor Node.js

Temas a Tratar

- Servidor Node.js
 - Introducción.
 - Crear servidor.

Introducción a Node.js (1/3)

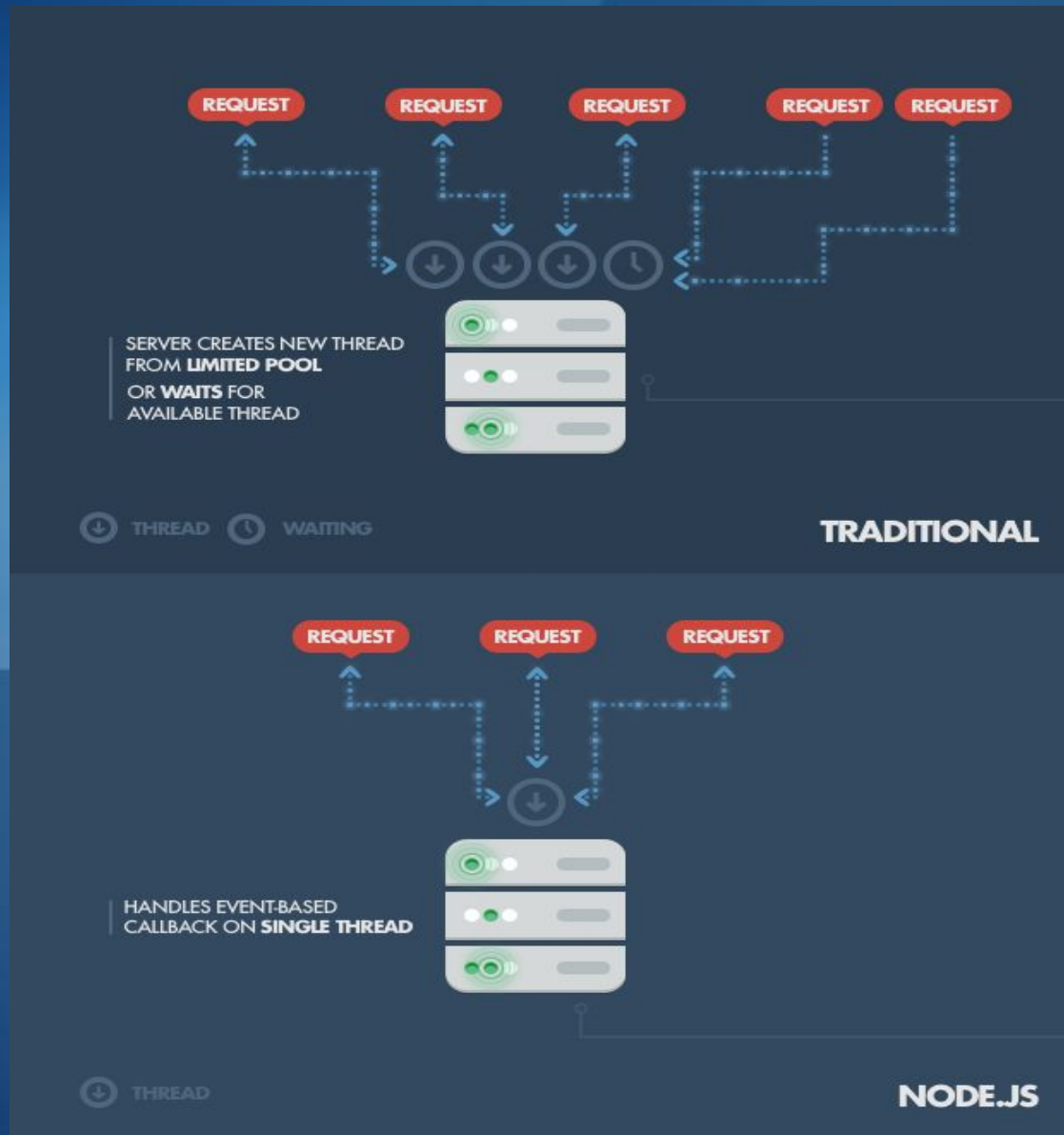
- **Node.js** es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación *JavaScript*, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor **V8** de Google.
- Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web.



Introducción a Node.js (2/3)

- El funcionamiento interno es bastante interesante.
- Mientras que las tradicionales técnicas de servicio web generan por cada conexión (solicitud) un nuevo subproceso (hilo), Node.js opera en un único subproceso, no utiliza el bloqueo de llamadas de E/S, lo que le permite admitir decenas de miles de conexiones simultáneas.

Introducción a Node.js (3/3)



Temas a Tratar

- Servidor Node.js
 - Introducción.
 - **Crear servidor.**

Servidor Node.js (1/3)

- Inicializar un nuevo proyecto de Node.js.
 - En una carpeta vacía, escribir el comando:

```
npm init -y
```

- Agregar el archivo dónde se creará el servidor.
 - Para la creación se utilizará el framework *express*.

```
npm install express
```

- Opcionalmente, agregar las definiciones de tipos de Node.js.

```
npm install -g @types/node
```

```
npm install --save-dev @types/node
```

Servidor Node.js (2/3)

- Crear el servidor Node.js.

```
//SE REQUIERE EL MODULO DE 'EXPRESS'  
const express = require("express");  
  
//CREO EL SERVIDOR NODEJS  
const app = express();
```

- Establecer el puerto dónde el servidor 'escuchará'.

```
//ESTABLEZCO LA PROPIEDAD 'PUERTO'  
app.set("puerto", 8008);  
  
//INDICO QUE EL SERVIDOR 'ESCUCHE' UN DETERMINADO PUERTO  
app.listen(app.get("puerto"), ()=>{  
  console.log("servidor corriendo sobre puerto ", app.get("puerto"));  
});
```

Servidor Node.js (3/3)

- En la terminal, iniciar el servidor.

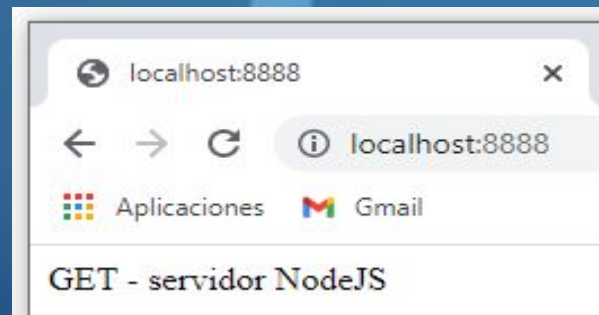
```
node nombre_archivo.extensión
```

```
servidor corriendo sobre puerto 8008
```

- Establecer los métodos y rutas al servidor.

```
app.get('/', (request:any, response:any)=>{  
  response.send('GET - servidor NodeJS');  
});
```

- Reiniciar el servidor y probarlo desde un navegador.



Demo

Apéndice (crear code-snipped)

- Dirigirse al menú:

Archivo → Preferencias → Configurar fragmentos de usuario.

- Seleccionar la opción:

Nuevo archivo de fragmentos globales.

Indicar el nombre del fragmento.

- Crear el code-snipped dentro de las llaves ({}).
- Guardar los cambios.

Apéndice (crear code-snipped)

```
"servidor_nodejs" :  
{  
  "scope": "typescript",  
  "prefix": "snjs",  
  "body": [  
    "const express = require('express');",  
    "",  
    "const app = express();",  
    "",  
    "app.set('puerto', $1);",  
    "",  
    "app.get('/', (request:any, response:any)=>{",  
    "\tresponse.send('GET - servidor NodeJS');",  
    "});",  
    "",  
    "",  
    "app.listen(app.get('puerto'), ()=>{",  
    "\tconsole.log('Servidor corriendo sobre puerto:', app.get('puerto'));",  
    "});",  
  ],  
  "description": "Creación de servidor NodeJS"  
}
```

Apéndice (nodemon)

- **nodemon** es una herramienta que ayuda a desarrollar aplicaciones basadas en Node.js reiniciando automáticamente la aplicación cuando se detectan cambios en los archivos.
- **nodemon** no requiere ningún cambio adicional en su código de desarrollo.
- Para usar nodemon, reemplace la palabra ***node*** en la línea de comando al ejecutar su script.

```
npm install -g nodemon
```

```
npm install --save-dev nodemon
```





Ejercitación