

Creación e instalación de servidor Express sobre NodeJs

1 Lo primero que se debe hacer es **instalar el *framework*** en el proyecto. Cabe resaltar que este está constituido de módulos.

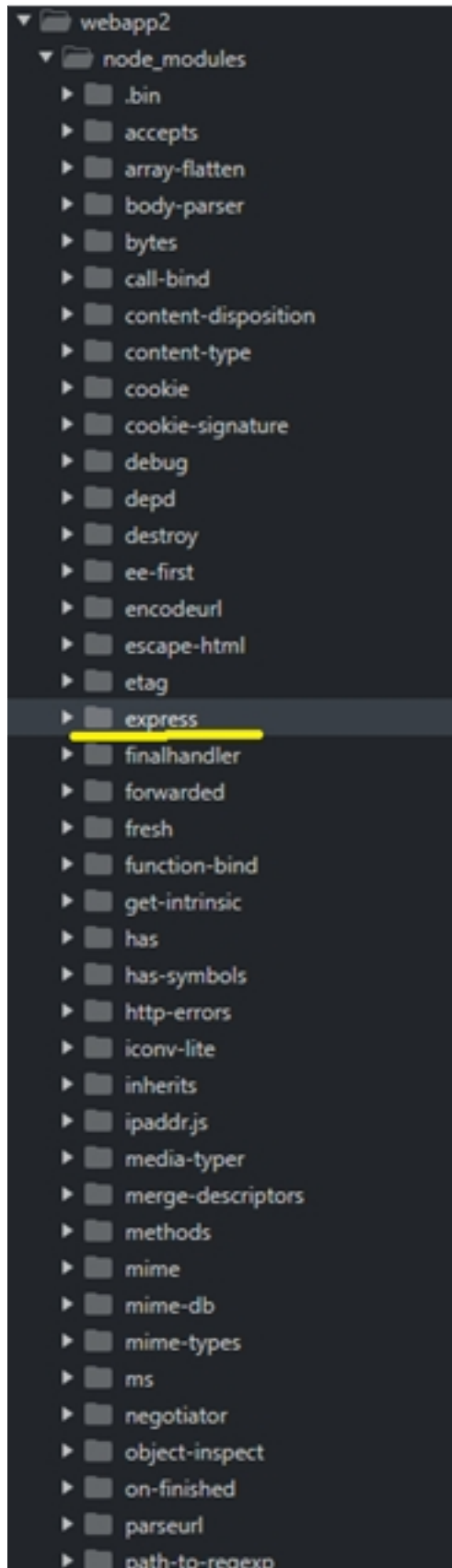
Una vez creado el espacio de trabajo del proyecto 'webapp2' desde la línea de comandos, procedemos a instalar el framework mediante el npm:

```
c:\workspace\webapp2>npm install express
```

```
Símbolo del sistema
D:\workspace\webapp2>npm install express
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.
added 57 packages, and audited 58 packages in 2s
7 packages are looking for funding
  run 'npm fund' for details
found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 8.11.0 -> 8.12.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.12.1
npm notice Run npm install -g npm@8.12.1 to update!
npm notice
D:\workspace\webapp2>
```

Instalación de Express

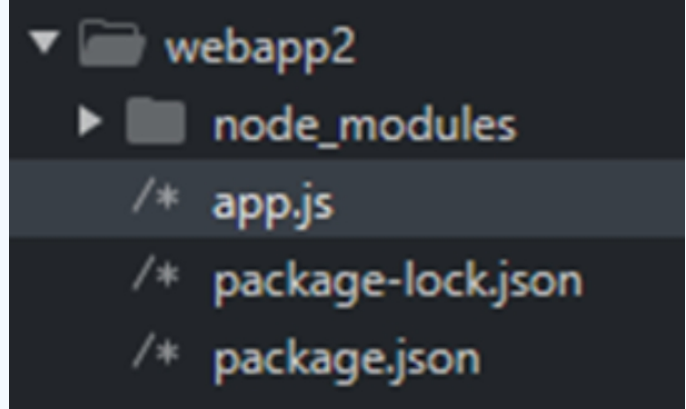
Cuando se instala el *framework*, como vemos, se crea la carpeta 'node_modules', y en esta, una subcarpeta 'express' y otras más que dependen del *framework* Express:



Validación de instalación de Express

Ya se encuentra el *Express* instalado, ahora procederemos a crear la primera app con el *framework*; para ello, se debe ingresar al espacio de trabajo de webapp2 y crear un archivo denominado app.js, que será el archivo principal del proyecto.

La estructura del proyecto debe quedar de la siguiente manera:



Primera aplicación con Express

En el código del archivo principal, se deberá llamar al *framework* para iniciar a hacer uso de sus módulos y funcionalidades preestablecidas.

```
app.js
1  const express = require('express')
2  const app = express()
3
4  app.get('/', (req, res) => {
5    res.send(`<!doctype html><html><head></head><body><h1>
6      Mi primer pagina con Express</h1></body></html>`)
7  })
8
9  const server = app.listen(8888, () => {
10    console.log('Servidor web iniciado')
11  })
```

Código de app.js

Lo primero que se puede apreciar es que se incluye el módulo del *framework*:

```
const express = require('express')
```

Se invoca la función que retorna el objeto donde se almacena la constante app:

```
const app = express()
```

Por medio del método **get** del objeto **app**, se especifica la ruta '/' (raíz de nuestro sitio) asociada a una función anónima que le pasamos como segundo parámetro, la cual tiene la lógica que se requiere al invocar el *path*.

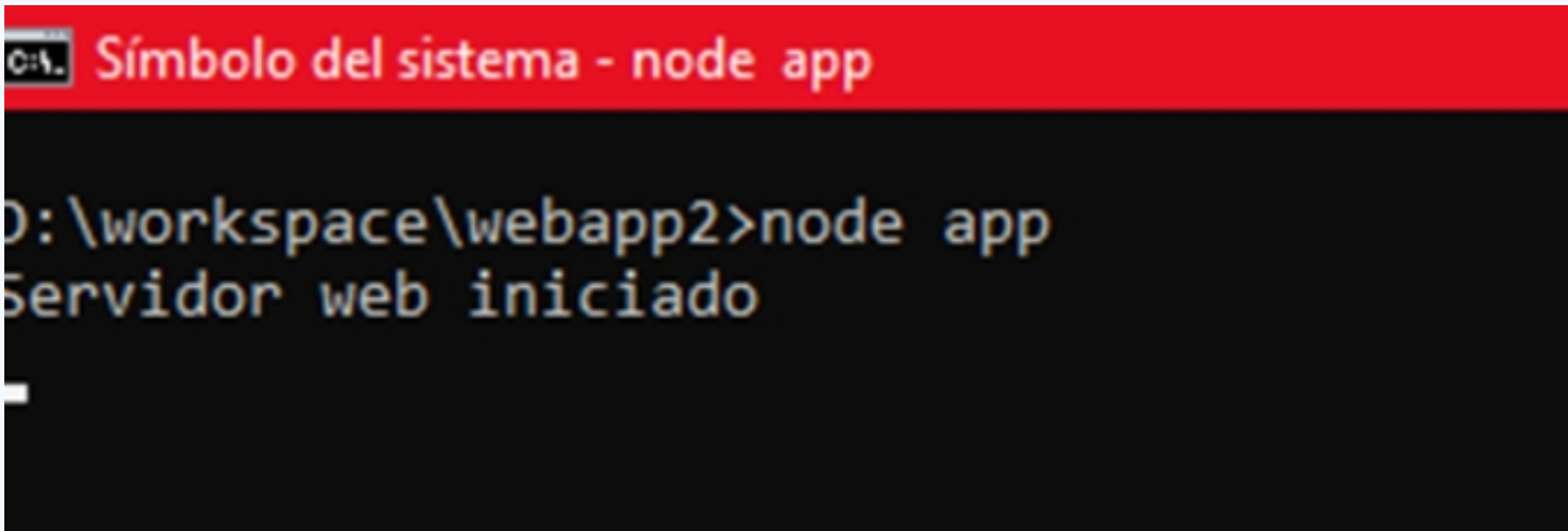
La función anónima recibe dos parámetros: 'req' y 'res'. Mediante el objeto 'req', recuperamos los datos enviados en la petición, y por medio del 'res', se responde al navegador que hizo la solicitud generando código HTML:

```
app.get('/', (req, res) => {
  res.send(`<!doctype html><html><head></head><body><h1>
    Mi primer pagina con Express</h1></body></html>`)
})
```

A la final, se inicia el servidor web invocando el método listen:

```
const server = app.listen(8888, () => {
  console.log('Servidor web iniciado')
})
```

Por último, se inicia la aplicación. Recordemos que se inició el servidor en el puerto 8888:



Inicio de aplicación web

Si accedemos a la raíz del dominio localhost por el puerto 8888, tenemos como resultado:



Resultado en navegador web

