

# Práctica 4. Primera Parte: Implementación de los ejemplos

Noelia Escalera Mejías

30 de abril de 2020

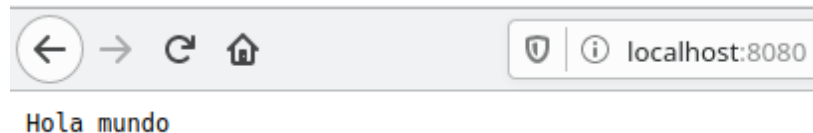
## 1. Ejemplo 1

Se trata de un hello world.

- Lanzamos el servicio

```
noelia@noelia-HP-ENVY-17-Notebook-PC:~/Escritorio/Universidad/3º/DSD/Prácticas/P4/Parte 1/Ejemplo 1$ nodejs helloworld.js
Servicio HTTP iniciado
```

- Vamos a localhost:8080



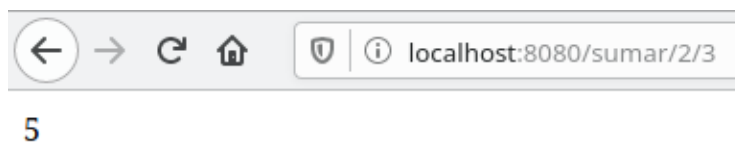
## 2. Ejemplo 2

Se trata de una calculadora.

- Lanzamos el servicio

```
noelia@noelia-HP-ENVY-17-Notebook-PC:~/Escritorio/Universidad/3º/DSD/Prácticas/P4/Parte 1/Ejemplo 2$ nodejs calculadora.js
Servicio HTTP iniciado
```

- Vamos a localhost:8080/sumar/2/3



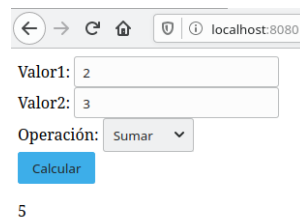
### 3. Ejemplo 3

Tenemos una calculadora similar a la del ejemplo anterior pero con una interfaz.

- Lanzamos el servicio

```
noelia@noelia-HP-ENVY-17-Notebook-PC:~/Escritorio/Universidad/3º/DSD/Prácticas/P4/Parte 1/Ejemplo 3$ nodejs calculadora-web.js
Servicio HTTP iniciado
```

- Vamos a localhost:8080 e intentamos hacer una operación



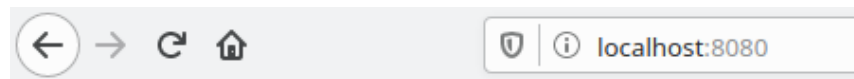
### 4. Ejemplo 4

Se trata de un programa que muestra las conexiones activas actualmente.

- Lanzamos el servicio

```
noelia@noelia-HP-ENVY-17-Notebook-PC:~/Escritorio/Universidad/3º/DSD/Prácticas/P4/Parte 1/Ejemplo 4$ nodejs connections.js
Servicio Socket.io iniciado
```

- Vamos a localhost:8080



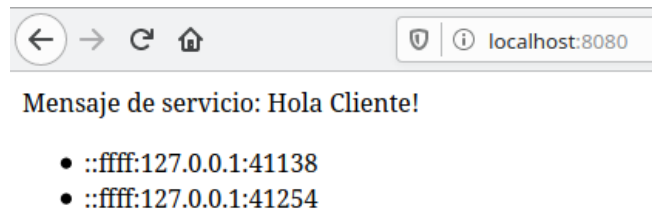
Mensaje de servicio: Hola Cliente!

- ::ffff:127.0.0.1:41138

- En la terminal nos aparece el siguiente mensaje:

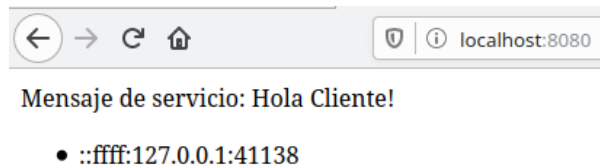
```
noelia@noelia-HP-ENVY-17-Notebook-PC:~/Escritorio/Universidad/3º/DSD/Prácticas/P4/Parte 1/Ejemplo 4$ nodejs connections.js
Servicio Socket.io iniciado
New connection from ::ffff:127.0.0.1:41138
```

- Si abrimos otra pestaña de navegación



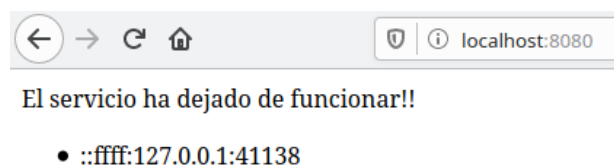
```
noelia@noelia-HP-ENVY-17-Notebook-PC:~/Escritorio/Universidad/3º/DSD/Prácticas/P4/Parte 1/Ejemplo 4$ nodejs connections.js
Servicio Socket.io iniciado
New connection from ::ffff:127.0.0.1:41138
New connection from ::ffff:127.0.0.1:41254
```

- Si ahora cerramos una ventana

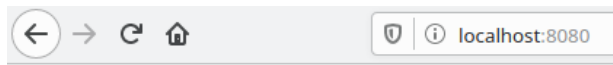


```
noelia@noelia-HP-ENVY-17-Notebook-PC:~/Escritorio/Universidad/3º/DSD/Prácticas/P4/Parte 1/Ejemplo 4$ nodejs connections.js
Servicio Socket.io iniciado
New connection from ::ffff:127.0.0.1:41138
New connection from ::ffff:127.0.0.1:41254
El cliente ::ffff:127.0.0.1 se va a desconectar
[ { address: '::ffff:127.0.0.1', port: 41138 },
  { address: '::ffff:127.0.0.1', port: 41254 } ]
El usuario ::ffff:127.0.0.1 se ha desconectado
```

- Y si paramos el servicio



- Si lo volvemos a poner a funcionar se reconecta automáticamente.



Mensaje de servicio: Hola Cliente!

- ::ffff:127.0.0.1:41532

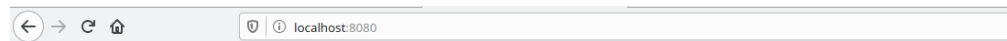
## 5. Ejemplo 5

Se trata de un programa que guarda un histórico de conexiones.

- Lanzamos el servicio

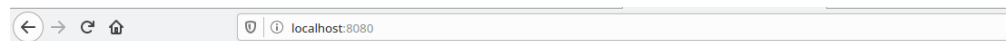
```
noel@moeta-HP-ENVY-17-Notebook-PC:~/Escritorio/Universidad/3º/DSO/Prácticas/P4/Parte 1/Ejemplo 5$ nodejs mongo-test.js
Servicio MongoDB iniciado
(node:5070) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new
Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
```

- Vamos a localhost:8080



- {"\_id":"5ea9cc955af53b303290da8f","host":"::ffff:127.0.0.1","port":{"host":"::ffff:127.0.0.1","port":42180},"time":"2020-04-29T18:51:01.444Z"}
- {"\_id":"5ea9cca05af53b303290da90","host":"::ffff:127.0.0.1","port":{"host":"::ffff:127.0.0.1","port":42196},"time":"2020-04-29T18:51:12.832Z"}

- Si abrimos otra pestaña se añadie otra conexión a la lista.



- {"\_id":"5ea9cc955af53b303290da8f","host":"::ffff:127.0.0.1","port":{"host":"::ffff:127.0.0.1","port":42180},"time":"2020-04-29T18:51:01.444Z"}
- {"\_id":"5ea9cca05af53b303290da90","host":"::ffff:127.0.0.1","port":{"host":"::ffff:127.0.0.1","port":42196},"time":"2020-04-29T18:51:12.832Z"}
- {"\_id":"5eaae0d120e12613cee6e02c","host":"::ffff:127.0.0.1","port":{"host":"::ffff:127.0.0.1","port":53820},"time":"2020-04-30T14:29:37.390Z"}
- {"\_id":"5eaae0f620e12613cee6e02d","host":"::ffff:127.0.0.1","port":{"host":"::ffff:127.0.0.1","port":53844},"time":"2020-04-30T14:30:14.255Z"}

Aunque detengamos el servicio y lo volvamos a iniciar se guardan las conexiones.