Crear un servidor

Código inicial

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

// Creamos un servidor HTTP

const server = http.createServer((req, res) => {

  // Establecemos el código de estado de la respuesta (200 = OK)

  res.statusCode = 200;

  // Definimos el tipo de contenido de la respuesta como texto plano

  res.setHeader('Content-Type', 'text/plain');

  // Terminamos la respuesta enviando un mensaje al cliente

  res.end('Estamos online!\n');

});
```

Solicitud y respuesta

```
http.createServer((req, res) => {

  // ...

});
```

req y res son abreviaciones comunes para:

- req → Request (solicitud): representa la petición que el cliente (como tu navegador) hace al servidor. Contiene información como:

    - la URL que se solicitó (req.url)
    - el método HTTP usado (GET, POST, etc.) (req.method)
    - encabezados enviados (req.headers)
    - cuerpo de la petición (en caso de POST o PUT) (req.body, con ayuda de librerías)
- res → Response (respuesta): representa la respuesta que el servidor le va a devolver al cliente. Con res, vos podés:
    - establecer el código de estado (como 200, 404, etc.) → res.statusCode
    - definir encabezados → res.setHeader(...)
    - enviar contenido de respuesta → res.end(...)

Entonces, en conjunto:

- req es lo que te llega del usuario,
- res es cómo respondés a esa solicitud.

Métodos

- **statusCode**: Esto le dice al navegador o cliente que la solicitud fue procesada con éxito.
- **setHeader**: Define el tipo de contenido que el servidor va a devolver. Esto es clave para que el cliente interprete correctamente la respuesta. En el ejemplo, estas diciendo "esto es texto plano". Si enviaras HTML, cambiarías a 'text/html'.
- **end**: Finaliza la respuesta. Si no lo usás, el servidor no sabe cuándo terminó la respuesta y se puede colgar la conexión.

¿Siempre se deben usar estos métodos?

No siempre, pero casi siempre son recomendables. A medida que tu servidor crece o se vuelve más complejo (por ejemplo, devolviendo errores, JSON, archivos, etc.), controlar explícitamente el estado, los encabezados y el cierre de la respuesta te permite mantener un comportamiento coherente y depurable.