

Desarrollo Aplicaciones Web Avanzadas

Mg. Jorge Eduardo Castañeda Alban

jcastaneda@tecsup.edu.pe

JavaScript con Node.js

Semana 03

01

Conocer el uso de los métodos HTTP en Express

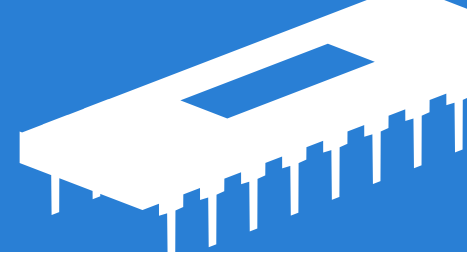
02

Conocer el uso de Request y Response en Express

03

Conocer el uso de Request Params en Express

Indice



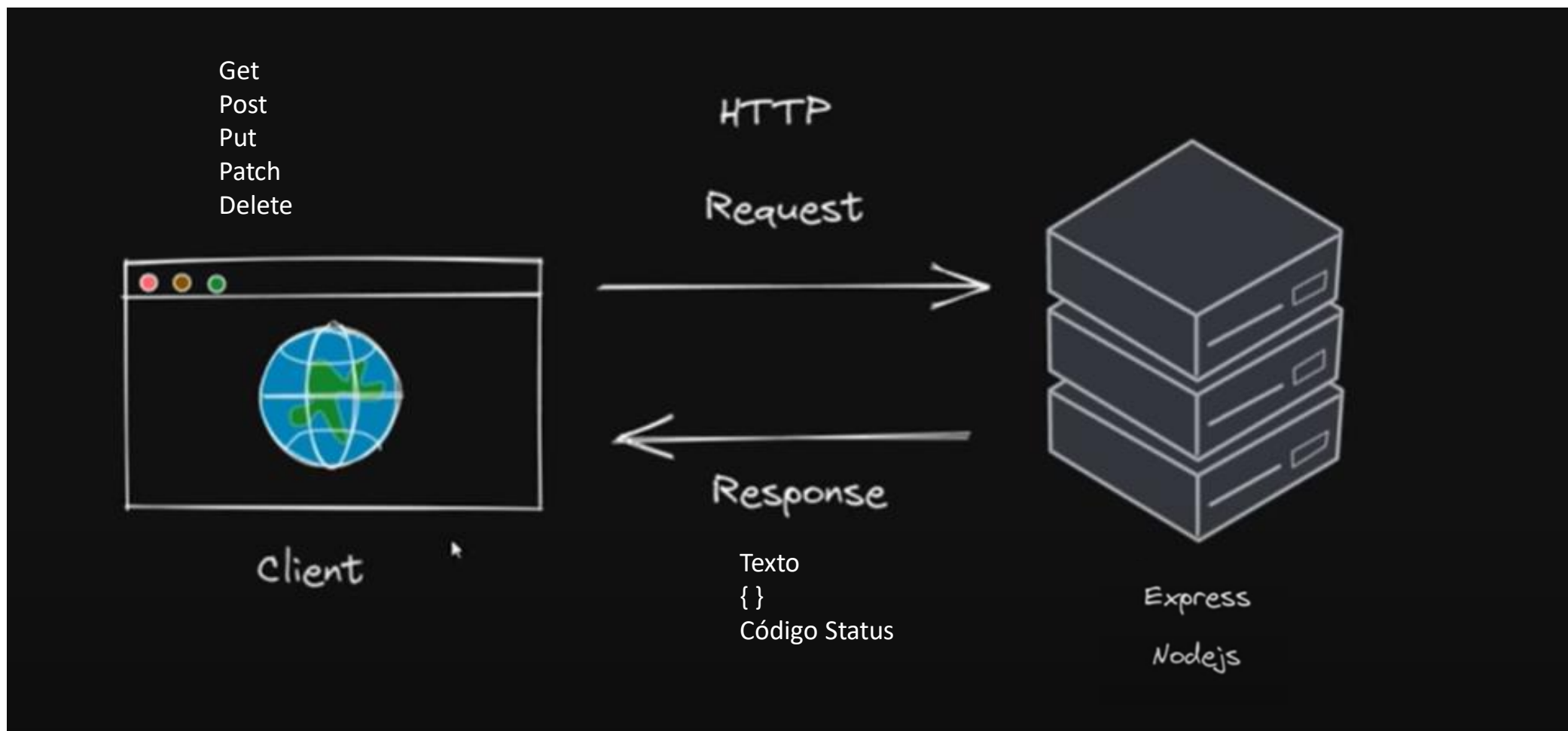
- Son formas mediante las cuales el cliente solicita alguna acción al servidor.
- Entenderlos es fundamental para comprender la forma en que funciona la metodología REST (Representational State Transfer).
- HTTP define una gran cantidad de métodos que son utilizados para diferentes circunstancias.

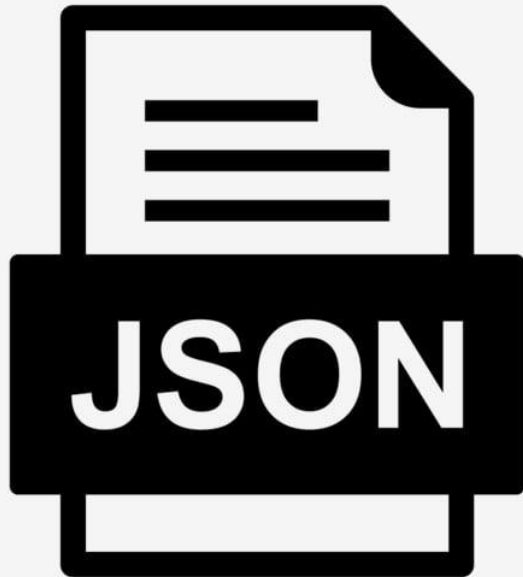
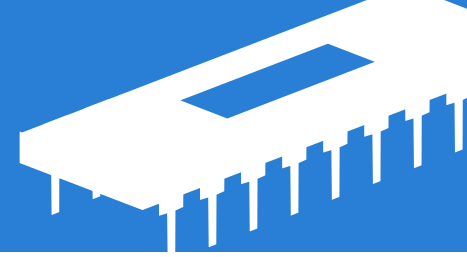
¿Cuáles son los métodos mas empleados?

- **GET:** Es utilizado para **consultar** información al servidor, son muy parecidos a los SELECT de BD.
- **POST:** Es utilizado para solicitar la **creación de un nuevo registro**, es decir, algo que no existía previamente. Equivale al INSERT en BD.
- **PUT:** Se utiliza para **actualizar por completo un registro existente** es decir, realizar un UPDATE en BD.
- **PATCH:** Este método es similar al método PUT, pues permite actualizar un registro, sin embargo, este se utiliza cuando se requiere actualizar solo **parte de registro**.
- **DELETE:** Este método es usado para **Eliminar un registro**.



¿Cómo trabajan los métodos HTTP?





- Sus iniciales provienen de JavaScript Object Nation.
- Es un formato ligero de intercambio de datos.
- Se basa en indeterminados subconjunto de datos.
- Es una alternativa mas sencilla de manejar que el XML.

Ejemplos de Métodos HTTP - 1

GET

```
app.get('/products', (req, res) => {  
  . . . .  
  . . . //validate data  
  . . . //query bd  
  . . . //process data  
  . . . //other actions on data  
  
  . . . res.send('lista de productos')  
})
```


Ejemplos de Métodos HTTP - 2

POST

```
app.post('/products', (req, res) => {  
  ...res.send('creando productos')  
})
```

PUT

```
app.put('/products', (req, res) => {  
  ...res.send('actualizando productos')  
})
```

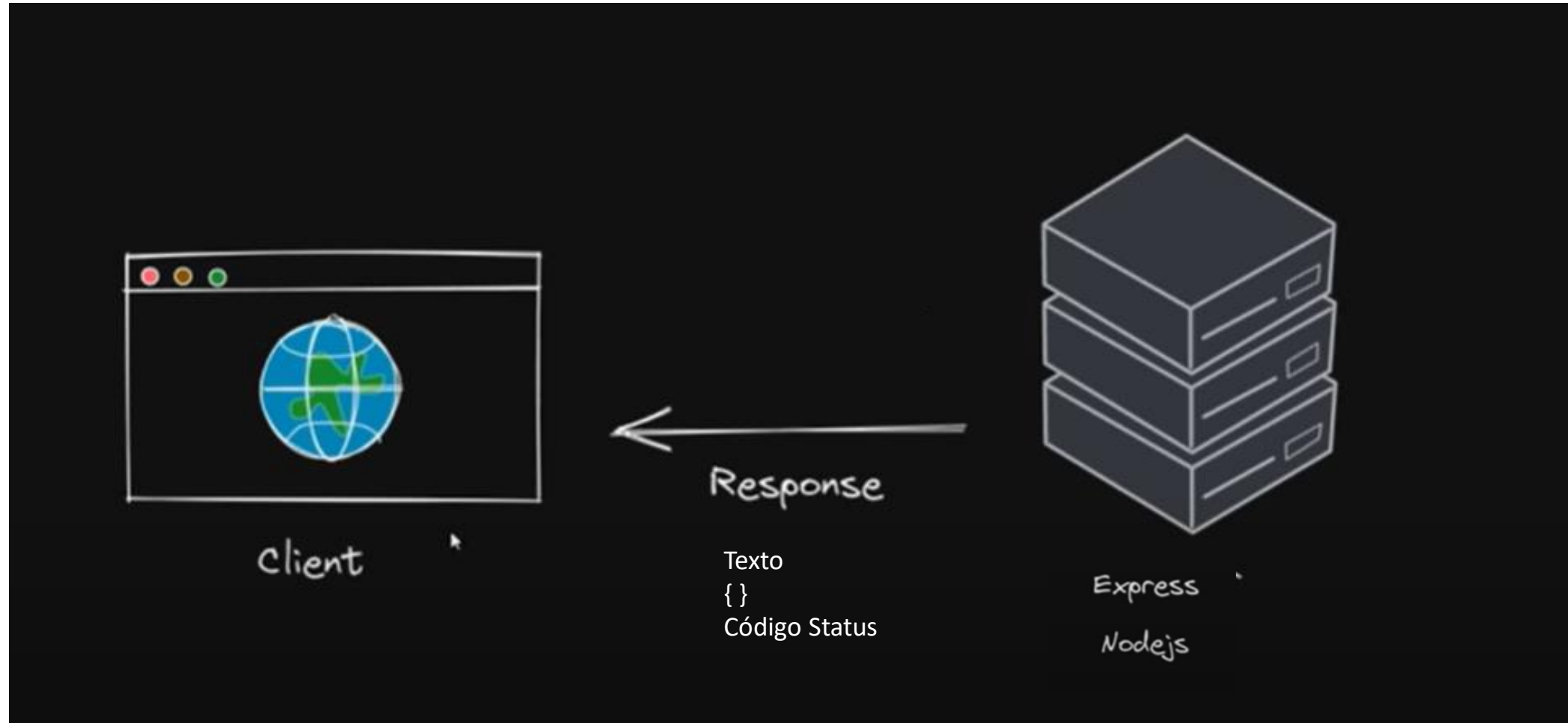
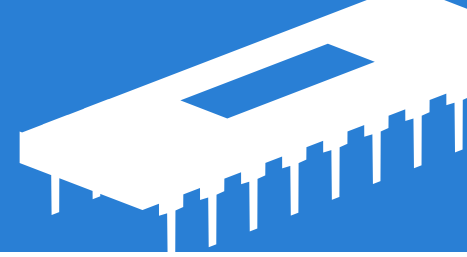
Ejemplos de Métodos HTTP - 3

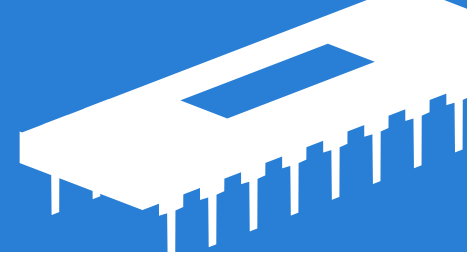
DELETE

```
app.delete('/products', (req, res) => {  
  ...res.send('eliminando productos')  
})
```

PATCH

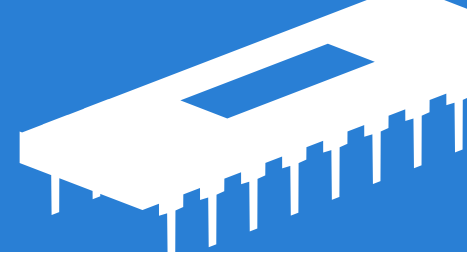
```
app.patch('/products', (req, res) => {  
  ...res.send('actualizando una parte del producto')  
})
```





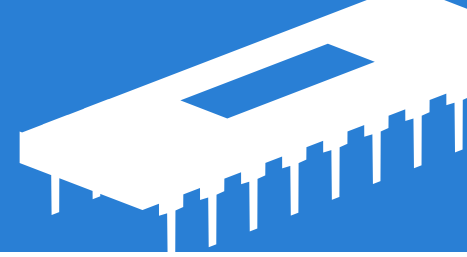
Response con JSON

```
app.get('/user', (req, res) => {  
  res.json({ nombre: "Jorge",  
             apellido: "Castañeda",  
             edad: 42,  
             points: [1, 2, 3],  
             adress: {  
               ciudad: "Lima - Peru",  
               distrito: "San Miguel",  
               calle: "Av. Brigida Silva"  
             })  
})
```



Response con Archivo

```
app.get('/myfile', (req, res) => {  
  res.sendFile('./arquitectura.png', {  
    root: __dirname  
  })  
})
```



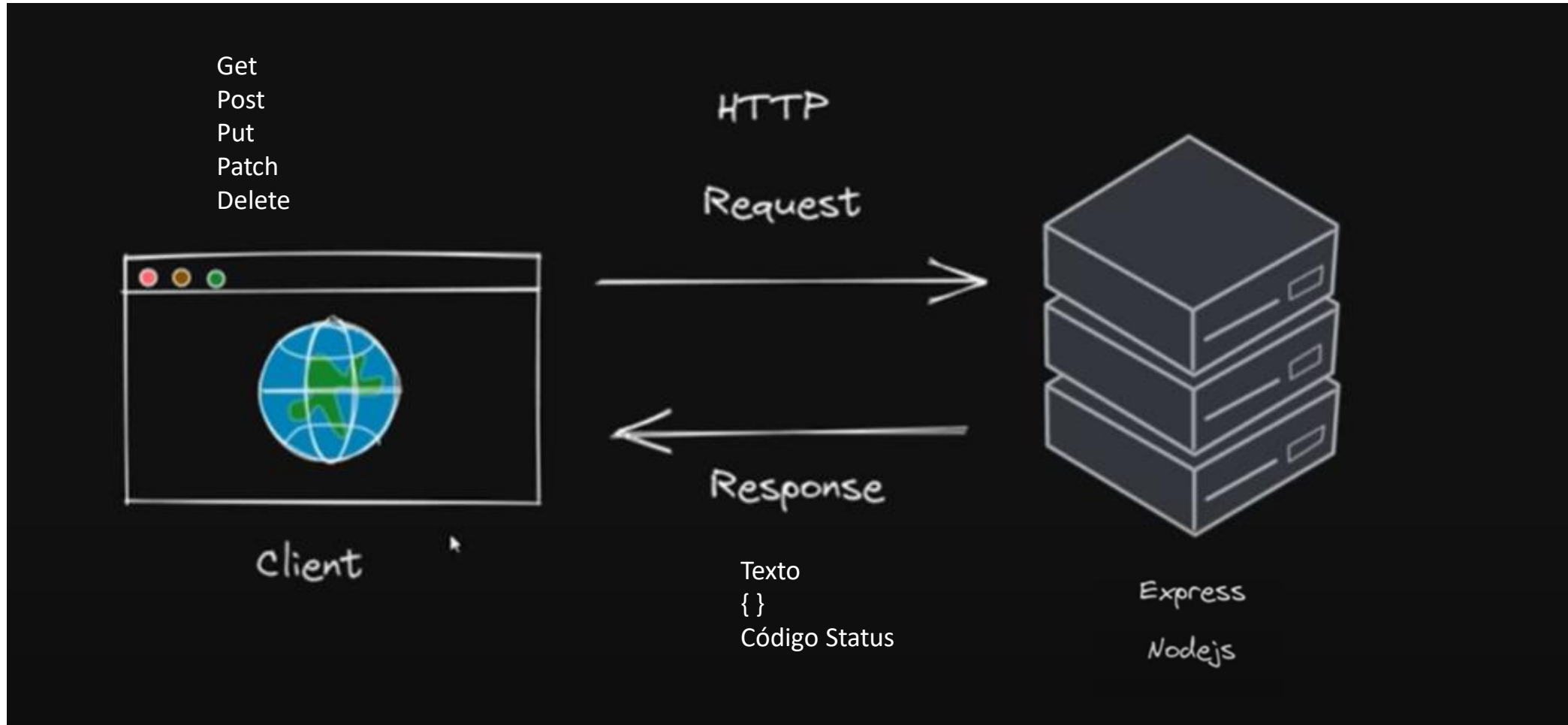
Response con Archivo

```
app.get('/myfile', (req, res) => {  
  res.sendFile('./arquitectura.png', {  
    root: __dirname  
  })  
})
```

Response con Texto

```
app.get('/', (req, res) => {  
  res.send('Hello world')  
})
```


Request Params en Express



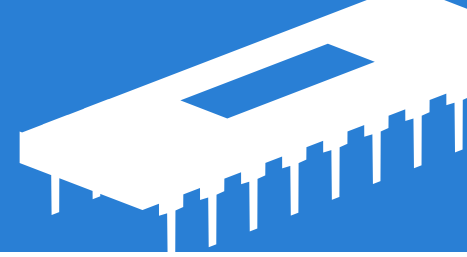
Ejemplo de Request Params en Express

Parámetro Texto

```
app.get('/hello/:username', (req, res) => {  
  //console.log(req.params)  
  console.log(typeof req.params.username)  
  res.send(`Hello ${req.params.username.toUpperCase()}`)  
})
```

Parámetro Numérico

```
app.get('/add/:x/:y', (req, res) => {  
  const {x, y} = req.params  
  res.send(`Result: ${parseInt(x) + parseInt(y)}`)  
})
```



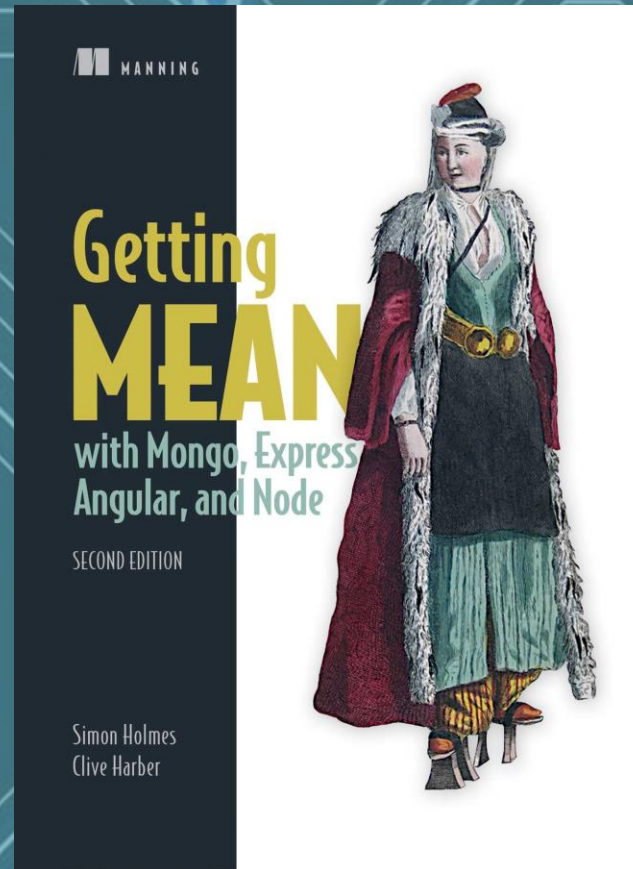
BASE DE DATOS

SERVIDOR con
una API REST

FRONTEND



Bibliografía





Muchas Gracias

Para la Próxima Clase: Revisar la Plataforma