

TEMA 1 INTRODUCCIÓN A NODE.JS

Parte V – Introducción a los Servicios REST
5.1 Introducción y Comandos URIs en
NODE.JS

INTRODUCCIÓN A SERVICIOS REST

- ❑ REST: REpresentational State Transfer. Un estilo de arquitectura de aplicaciones distribuidas.
 - ❑ Cada recurso es una URI (identificador uniforme de recurso).
 - ❑ Delimitado número de comandos:
 - GET: Obtener datos. SELECT
 - POST: Añadir elementos. INSERT
 - PUT: Modificar elementos. UPDATE
 - DELETE: Eliminar elementos. DELETE
-

Funcionamiento API REST

- ☐ Se identifica el recurso a solicitar y el comando a aplicarle.
 - ☐ El servidor que ofrece la API REST devuelve una respuesta a esa petición.
 - ☐ La respuesta puede ser en JSON o XML.
 - ☐ Para simular estas peticiones usaremos el programa POSTMAN.
-

Obtener el comando de la petición

- ❑ Tenemos que acceder a la propiedad `method` de la petición (`Request`).
 - ❑ Crear un fichero llamado `"prueba_rest.js"`:

```
let atenderPetición = (request, response) => {  
  response.writeHead(200, {"Content-Type": "text/plain"});  
  response.write("El comando recibido es: " +  
    request.method);  
  response.end();  
};
```
 - ❑ Si hacemos una llamada con el navegador, nos dirá que es la operación GET. Que es la operación por defecto de un navegador.
-

Obtener el comando de la petición

II

- ❑ Para la operación POST, necesitaríamos utilizar un formulario HTML.

```
<html>
<head>
  <meta charset="utf8">
</head>
<body>
  <form action="http://localhost:8080"
    method="post">
    <input type="submit" value="Enviar formulario" />
  </form>
</body>
</html>
```
 - ❑ Para conseguir PUT o DELETE ya deberíamos usar algún código JAVASCRIPT en el cliente.
-

Enrutar comandos y URIs

- ❑ Es similar a lo que ya vimos en enrutamiento de URL.
 - ❑ Combinaremos el `request.url` y `request.method` para distinguir qué URI se ha solicitado y con qué método y así poder dar la respuesta que queramos a cada una. Lo veremos en un ejemplo posterior.
-