

# TEMA 1 INTRODUCCIÓN A NODE.JS

---

Parte II - Node.js como servidor web  
2.3- Enrutamiento

# Enrutamiento

---

- ❑ Tal como tenemos hecho el servidor, estas tres url funcionarían sin problemas:
    - *http://localhost:8080*
    - *http://localhost:8080/loquesea*
    - *http://localhost:8080/una\_pagina.html*
  - ❑ Como vemos hay páginas ni que existen, pero ahora mismo nuestro servidor funciona cualquier cosa que venga del 8080
  - ❑ Y esto no es real, necesitamos un mecanismo que identifique la ruta y actuar en consecuencia, o dar un error o dar el servicio que indica la url seleccionada.
  - ❑ Es lo que se denomina **enrutamiento** (*routing*), analizar la petición HTTP y generar una u otra respuesta en función de la misma.
-

# Enrutamiento II

---

- Veamos un mecanismo rudimentario, usando la propiedad url del objeto request.

```
var atenderPetición = (request, response) => {  
  response.writeHead(200, {"Content-Type": "text/plain"});  
  
  if (request.url === '/') {  
    response.write("Página principal");  
  } else if (request.url === '/bienvenida') {  
    response.write("Bienvenido/a");  
  } else if (request.url === '/despedida') {  
    response.write("Hasta pronto");  
  }  
  
  response.end();  
}
```

---

# Enrutamiento III

---

- ❑ Obviamente esta forma de enrutar también es insuficiente.
- ❑ En una web real tenemos partes de la url dinámicas.
- ❑ O debemos distinguir si venimos a una página sin id o con id, como podría ser (para ello los framework nos van ayudar, como ya veremos):

<http://mipaginadelibros.com/fichaLibro/1234>

[http://mipaginadelibros.com/fichaLibro?id\\_libro=1234](http://mipaginadelibros.com/fichaLibro?id_libro=1234)

---

# EJERCICIO 1

- ❑ Crea una carpeta llamada "Tema1\_EnrutamientoBasico" en tu carpeta de ejercicios. Dentro, deberás definir un servidor que identifique y dé una respuesta diferente para cada una de estas rutas:
- ❑ `/usuario` : mostrará el login del usuario que entró al sistema operativo (usando el módulo "os"), como texto plano
- ❑ `/carpeta` : mostrará un listado de los archivos y carpetas contenido en la carpeta raíz de la aplicación (usando el módulo "fs"), como texto plano, con un archivo o carpeta por línea
- ❑ `/` : mostrará una página HTML predefinida llamada "index.html", con este contenido:

```
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>Página de bienvenida</h1>
    <ul>
      <li><a href="/usuario">Ver usuario del sistema</a></li>
      <li><a href="/carpeta">Ver carpeta de la aplicación</a></li>
    </ul>
  </body>
</html>
```
- ❑ Notar que el contenido de la página contiene dos enlaces para acceder a las dos rutas anteriores.
- ❑ Cualquier otra ruta deberá mostrar una página de error como la siguiente (busca alguna imagen en Internet que te sirva para construir una página similar, y decide cómo construirla y mostrarla desde el servidor).



# Procesar partes dinámicas en la ruta

---

- ❑ Ante rutas con partes dinámicas como:
    - *`http://mipaginadelibros.com/fichaLibro/1234`*
  - ❑ Un método rudimentario (veremos mejores), es obtener la url y trocearla con `split`.

```
if (request.url.startsWith('/fichaLibro'))
{
    var partes = request.url.split("/");
    response.write("El código del libro es " + partes[2]);
}
```
  - ❑ Con `startsWith` arrancamos desde el patrón que nos interesa, y luego accedemos con el array que forma `split`.
-

# EJERCICIO 4

---

- ❑ Crea una carpeta llamada "Tema1\_EnrutamientoEdad" que deberá procesar una URL con esta estructura:
  - ❑ *http://servidor/saludo/nombre/edad*
  - ❑ Todo lo que llegue a *"/saludo"* deberá procesarse, y a continuación extraer el nombre del usuario y la edad. El servidor deberá saludar al usuario por su nombre, e indicarle si es o no mayor de edad. Por ejemplo:
  - ❑ *http://localhost:8080/saludo/nacho/39*
  - ❑ => Hola nacho, eres mayor de edad
  - ❑ *http://localhost:8080/saludo/pepe/15*
  - ❑ => Hola pepe, no eres mayor de edad
-

# Procesar la query string

---

- ❑ Trataremos de procesar una query string:  
<http://miservidor.com?param1=valor1&param2=valor2&param3=valor3>
- ❑ La idea es consultar la propiedad **query** del método **parse** módulo “**URL**”. Con el podemos acceder a cada dato por separado:

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

```
...
```

```
var parametros = url.parse(request.url, true).query;
```

```
var id = parametros.id; //Suponiendo que el campo se llame id.
```

---



# EJERCICIO 5

---

- ❑ Modifica el ejercicio anterior añadiendo un servidor alternativo que envíe el nombre y la edad del usuario como datos del formulario, de esta forma:
  - ❑ <http://servidor/saludo?nombre=xxx&edad=yyy>
  - ❑ Para procesar esta URL, emplea el módulo url como se ha explicado en el ejemplo anterior, y genera la misma respuesta que en el ejercicio previo.
-