

Trabajo práctico

Crear un servidor HTTP de archivos estáticos con Node.js. Suponiendo que el servidor esté escuchando peticiones en **localhost** en el puerto **8000**, si se realizara una petición GET a la URL <http://localhost:8000/prueba.txt>, el servidor deberá verificar que exista el archivo llamado prueba.txt en el mismo directorio en el cual el proceso fué iniciado.

En el caso de que el archivo prueba.txt no exista, el servidor deberá responder la petición con el contenido de **notFound.html** y con un status code 404.

En el caso de que el archivo prueba.txt exista, deberá responder el contenido del mismo.

Adicionalmente, utilizar el [módulo URL](#) para verificar que cada petición al servidor contenga una extensión de archivo y en el caso de que no contenga extensión (por ejemplo, si se recibiera una petición a <http://localhost:8000/prueba> o <http://localhost:8000/algo/prueba>), responder la petición con el contenido de **invalid.html** y con un *status code* 400.

.

En caso de que el servidor reciba peticiones con verbos HTTP distintos a GET, responder también con el contenido de **invalid.html** y con el *status code* 400.

Subir el servidor a una solución [Platform as a Service](#) de su elección.

notFound.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Error 404</title>
</head>
<body>
  <h2> Recurso no encontrado</h2>
</body>
</html>
```

invalid.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Error 400</title>
</head>
<body>
  <h2> Realizaste una petición inválida.</h2>
</body>
</html>
```

Recursos

[Introducción a HTTP \(inglés\)](#)

[Módulo HTTP](#)

[Módulo URL](#)

[Material Platform as a Service \(español\)](#)

[Ejemplo de server](#)