



Sistemas Distribuidos e Internet

Desarrollo Web con NodeJS

Ejercicios Complementarios

Curso 2022/ 2023

Contenido

1	INTRODUCCIÓN	1
2	SESIÓN 7 – INTRODUCCIÓN A NODE.JS	2

1 Introducción

Una de las condiciones que el alumno *tiene que cumplir* para no ir a la prueba de autoría es realizar al menos el **50% de los ejercicios complementarios** propuestos. La realización del porcentaje mínimo exigido se realizará en el horario de clase, pudiendo realizar el resto de los ejercicios en casa.

Para cada uno de los ejercicios realizado el alumno deberá realizar un commit en GitHub donde se indique la sesión y el número del ejercicio correspondiente. El mensaje del commit correspondiente se realizará siguiendo la siguiente nomenclatura:

Commit Message -> “SDI-sessionX-Ejercicio-complementarioY”

Donde:

- X – indica el número de la sesión
- Y – indica el número del ejercicio complementario

Ejemplo:

Commit Message -> “SDI-IDGIT-7.10x-Ejercicio complementario x”



2 Sesión 7 – Introducción a Node.js

1. **Ejercicio 1-** Crear un nuevo controlador **authors.js** con las siguientes rutas:
 - a. **authors/add:** Mostrar un formulario (vista: `authors/add.twig`) que permita añadir autores con los siguientes parámetros: `name`, `group` y `rol` (posibles valores: cantante, batería, guitarrista, bajista o pianista). En la carpeta `views` hay que crear una subcarpeta `authors` para guardar los ficheros `.twig` asociados a la entidad autores. **15%.**
 - b. **Versión POST del anterior.** Mostrará la información enviada por el formulario anterior. En el caso de que algún parámetro sea `undefined`, se mostrará el mensaje: “<Parámetro> no enviado en la petición.”. **15%.**
 - c. **authors/:** contendrá una variable de autores predefinidos, tal y como en la práctica lo tiene canciones. Enviará esta lista a la vista (`authors/authors.twig`) y mostrará por pantalla los datos (`name`, `group` y `rol`) de cada autor. **10%.**
 - d. Crear un método GET para **controlar, mediante comodines**, las rutas que incluyan parámetros a `authors/`. Deberán redirigir (**método: `res.redirect`**) a `authors`. Contrólese la posición de este método en el código, **los métodos anteriores deben seguir funcionando**. **10%.**

Nota: Subir el código a GitHub en este punto. Commit Message -> “SDI-IDGIT-7.101-Ejercicio complementario 1”

2. **Ejercicio 2-** Modificar el código necesario para **que los posibles roles del ejercicio anterior** (`authors/add.twig`), se reciban (array predefinido) **como parámetro en la vista**. Las **<option> del <select>** se rellenarán con la información de este parámetro. Si no se implementó la funcionalidad anterior, puede realizarse con el género de las canciones creado durante la práctica. **10%.**

Nota: Subir el código a GitHub en este punto. Commit Message -> “SDI-IDGIT-7.102-Ejercicio complementario 2”

3. **Ejercicio 3 –** Implementar en `authors.js` un método **authors/filter** que recibirá como **parámetro uno de posibles roles** (ejemplo: `authors/filter/cantante`) y **mostrará en pantalla solo los autores que tengan ese rol**. Obviamente, los autores estarán predefinidos en un array. Si no se implementó la funcionalidad anterior, puede realizarse con el género de las canciones creado durante la práctica. **40%.**

Nota: Subir el código a GitHub en este punto. Commit Message -> “SDI-IDGIT-7.103-Ejercicio complementario 3”