



# Sistemas Distribuidos e Internet

## Desarrollo Web con JEE y Spring Boot

### Ejercicios Complementarios

### Curso 2023/2024

#### Contenido

1	INTRODUCCIÓN.....	1
2	SESIÓN 3 - ARQUITECTURA MVC CON SPRING BOOT .....	2

## 1 Introducción

*Una de las condiciones* que el alumno *tiene que cumplir* para no ir a prueba de autoría es realizar al menos el **50% de los ejercicios complementarios** propuestos. La realización del porcentaje mínimo exigido se realizará en el aula de prácticas de laboratorios en horario de clase, pudiendo realizar el resto de los ejercicios en casa. En este caso los ejercicios que deberán realizarse de **forma obligatoria** durante la **sesión de laboratorio** serán los **ejercicios 1 y 2**.

Es **recomendable** realizar también el **ejercicio 3** para desarrollar todas las capas del patrón MVC.

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-IDGIT-sesiónX-Ejercicio-complementarioY.”**

Donde:

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

**Ejemplo:**

**Commit Message -> “SDI-IDGIT-sesión3eEjercicio-complementario1.”**



## 2 Sesión 3 - Arquitectura MVC con Spring Boot

1. **Ejercicio 1** - Haciendo uso del patrón MVC añadir a la aplicación web actual la funcionalidad de permitir gestionar profesores (professor). La aplicación tiene que ser capaz de responder a las peticiones de ***añadir, editar, ver detalle, listar y borrar*** un profesor. **(35%)**

### Cosas a tener en cuenta:

- a. El **controlador (ProfessorsController)** responderá a las distintas peticiones con un string y el texto correspondiente (no redirigirán a ninguna plantilla).
- b. Los datos del profesor serán: ***DNI, Nombre (name), Apellidos(surname) y Categoría (category, por ejemplo, Profesor Titular, Catedrático, Profesor Ayudante, Profesor Asociado, etc.).***
- c. Habrá un **servicio (ProfessorsService)** que gestionará la lógica del Negocio.
- d. Hay que crear una lista de profesores por defecto.
- e. Aquí no hace falta probar las peticiones que vayan POST (Se hará en el siguiente ejercicio).
- f. ***Todos los datos se guardan en memoria (No hay que guardar nada en Base de Datos)***

**Nota: Subir el código a GitHub en este punto. Commit Message -> "SDI-IDGIT-sesión3-ejercicio-complementario1."**

2. **Ejercicio 2:** Añadir las **Vistas HTML** que permitan **Añadir, Listar y Borrar** profesores de forma dinámica. **(15%)**

### Cosas para tener en cuenta:

- a. Ahora los métodos del controlador tendrán que redirigir a una vista HTML.
- b. Hay que utilizar el motor de plantilla Thymeleaf.
- c. Añadir al menú las opciones para acceder a estas funcionalidades.
- d. Las vistas se guardarán en una subcarpeta ***"/professor"***.
- e. Cuando se añada o se borre un profesor se redigirá a la lista de profesores para visualizar la lista actualizada.

**Nota: Subir el código a GitHub en este punto. Commit Message -> "SDI-IDGIT-sesión3-ejercicio-complementario2."**



3. **Ejercicio 3:** Al Ejercicio anterior añadir la capa **Modelo** que permita guardar, modificar, borrar y consultar la información en una base de datos. **(10%)**

**Cosas a tener en cuenta:**

- Utilizar la Interfaz **CrudRepository (ProfessorsRepository)** para la implementación de acceso a datos.
- La entidad profesor tiene que mapearse automáticamente al repositorio CRUD definido en el paso anterior.
- Define un campo id como código autonómico en la entidad Professor, si no lo incluiste antes.

La lista por defecto del ejercicio anterior ya no será necesaria. Esta se podrá crear de forma dinámica.

**Nota: Subir el código a GitHub en este punto. Commit Message -> "SDI- IDGIT-sesión3-ejercicio-complementario3."**

4. **Ejercicio 4 -** Añadir la funcionalidad que permita editar el registro de un profesor. **(25%)**

**Cosas para tener en cuenta:**

- Al editar el registro de un profesor NO enviará el identificador del registro en la URL, sino en el cuerpo del mensaje.
- A modificar un registro se mostrarán los datos del registro con los cambios actuales (ver detalle).
- Añadir en la lista de profesores una opción para acceder a esta funcionalidad.

**Nota: Subir el código a GitHub en este punto. Commit Message -> "SDI-IDGIT-sesión3-ejercicio-complementario4."**



5. **Ejercicio 5** - Refactorizar todas las vistas actuales para que utilicen los fragmentos head, nav y footer y eliminar los warnings de HTML. (15%) Ejemplo, **list.html**:

```
!DOCTYPE html>
<html lang="en">
<head th:replace="fragments/head"/>
<body>
!-- Barra de Navegación superior -->
<nav th:replace="fragments/nav"/>


Nota: Subir el código a GitHub en este punto. Commit Message -> "SDI-IDGIT-sesión3-ejercicio-complementario5."



Página 4 | 4



Sistemas Distribuidos e Internet



Este documento es copyright (C) 2024 Edward Rolando Núñez Valdez, Jordán Pascual y Enrique De la Cal  
Esta prohibida su distribución sin el consentimiento expreso y por escrito de los autores.


```