EJERCICIOS DE SPRING

6.2. Crea un proyecto implementando el CRUD básico de la entidad *Empleado* (id, nombre, email, salario, enActivo y genero), añade elementos BootStrap, Lombok, validaciones sobre los tres primeros campos de Empleado y crea un enumerado para el género (MASCULINO, FEMENINO, OTROS). Además, hay que crear una interfaz EmpleadoService donde me permita realizar en memoria las operaciones CRUD y un servicio EmpleadoServiceImplExcep, que implemente sus métodos. Cuando entremos en la aplicación nos mostrará los empleados de la imagen (introducidos en el Main en un Bean que implemente CommandLineRunner).

public interface EmpleadoService {

Empleado añadir(Empleado empleado);

List<Empleado> obtenerTodos();

Empleado obtenerPorId(long id) throws RuntimeException;

Empleado editar(Empleado empleado) throws RuntimeException;

void borrar(Long id) throws RuntimeException;

List<Empleado> buscarPorNombre(String nombre);

List<Empleado> buscarPorGenero(Genero genero);

}

A screenshot of a web page

Description automatically generated

A screenshot of a computer

Description automatically generated

* Hacer que el servicio lance excepciones en caso de error, donde el controlador las capture y llegue al usuario como textos con formato *alert* de BootStrap. No es necesario crear tus propias excepciones, pudiendo ser de tipo *RuntimeException.*
* No se admiten salarios de menos de 18000 euros, ni en el alta ni en la modificación de empleados.
* Añade a la vista principal una caja de búsqueda en la que el usuario pueda introducir un texto y se muestren los empleados que tienen ese texto incluido en alguna parte de su nombre sin distinguir mayúsculas de minúsculas. Incluirá también un botón *‘Reestablecer’* para volver a la situación inicial en la que sí se muestran todos.

A screenshot of a computer

Description automatically generated

* Añade a la vista principal (listView.html) una lista desplegable (script de JS utilizando window.location.href) con los valores de la enumeración *Genero* y que los usuarios puedan filtrar la vista según el género.

A screenshot of a computer

Description automatically generated

* La vista de todos los empleados debe estar ordenada por nombre.

**Pista**: para ordenar un ArrayList por un campo Long (p.ej: id):

Collections.sort(arrayAlumnos, Comparator.comparingLong(Alumno::getId));

Por un campo String (p.ej: nombre):

Collections.sort(arrayAlumnos, Comparator.comparing(Alumno::getNombre));

Para orden descendente:

Collections.sort(arrayAlumnos, Comparator.comparing

(Alumno::getNombre, Comparator.reverseOrder()));