

**Escuela Colombiana de ingeniería Julio Garavito**

**Seguridad y privacidad de TI**

**Laboratorio 11**

**Integrantes**

**Juan Camargo**

**Diego Castellanos**

**Profesor**

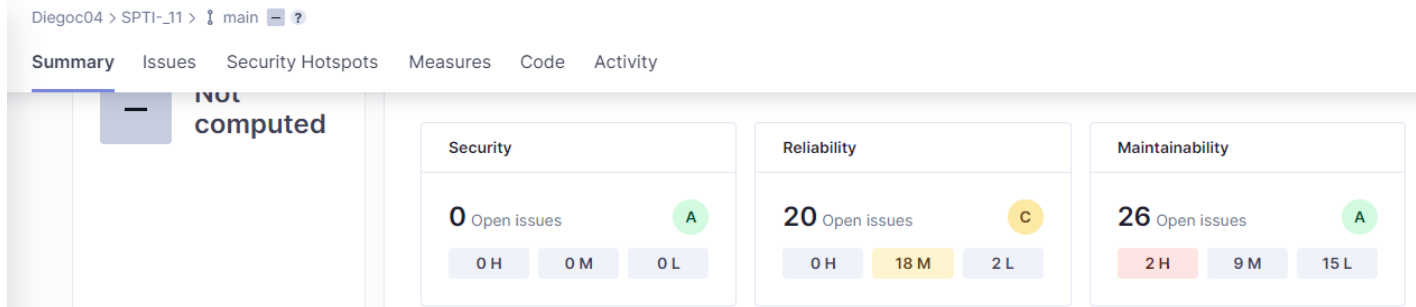
**Daniel Esteban Vela Lopez**

**Bogotá 2024**



<https://github.com/SebSanUwU/cvds-lab7.git>

## Sonar cloud



1)

src/.../java/co/edu/escuelaing/cvds/lab7/Lab7Application.java [See all issues in this file](#)

```
1 jorge... package co.edu.escuelaing.cvds.lab7;
2
3 jorge... import co.edu.escuelaing.cvds.lab7.model.Configuration;
4 juanca... import co.edu.escuelaing.cvds.lab7.model.Employee;
```

Remove this unused import 'co.edu.escuelaing.cvds.lab7.model.Employee'.

```
import co.edu.escuelaing.cvds.lab7.model.Configuration;
import co.edu.escuelaing.cvds.lab7.model.Employee;

package co.edu.escuelaing.cvds.lab7;

import co.edu.escuelaing.cvds.lab7.model.Configuration;
import co.edu.escuelaing.cvds.lab7.service.ConfigurationService;
```

Se nos explica que hay una importación innecesaria en el código y debería ser eliminada.

2)

```
@SpringBootApplication
public class Lab7Application {
    @Autowired
```

Remove this field injection and use constructor injection instead.

```
ConfigurationService configurationService;
@Autowired
EmployeeService employeeService;
```

```
@SpringBootApplication  Jorge Ulises Useche Cuellar +1 *
public class Lab7Application {

    ConfigurationService configurationService; 2 usages
    EmployeeService employeeService; no usages

    @Autowired  Jorge Ulises Useche Cuellar
    public static void main(String[] args) { SpringApplication.run(Lab7Application.class, args); }
```

No recomienda remover el campo de inyección y utilizar un constructor de inyección.

3)

```
@Bean
public CommandLineRunner run() throws Exception {
```

Define and throw a dedicated exception instead of using a generic one.

```
@Bean no usages  Jorge Ulises Useche Cuellar *
public CommandLineRunner run(){
```

Explica que es mejor usar una excepción dedicada en vez de una genérica, pero como en el método no se tiran excepciones, podemos eliminarlo.

4)

```
@Bean no usages Jorge Ulises Useche Cuellar +1 *
public CommandLineRunner run() {
    return (args) -> {

        logger.info(msg: "Adding Configurations....");
        configurationService.addConfiguration(new Configuration(

        logger.info(msg: "\nGetting all configurations....");
```

```
@Bean
public CommandLineRunner run(){
    return (args) -> {

        System.out.println("Adding Configurations....");

        configurationService.addConfiguration(new Configuration("price",
"800000"));

        System.out.println("\nGetting all configurations....");
        configurationService.getAllConfigurations().forEach(configuration
System.out.println(configuration));
```

Replace this use of System.out by a logger.

Se nos recomienda usar un logger, este permite tener niveles de registro.

5)

1 / 18 issues

src/.../ConfigurationController.java

Remove this unused import  
'co.edu.escuelaing.cvds.lab7.model.ToDoItem'

Remove this unused import  
'org.springframework.web.bind.annotation'

Remove this unused import  
'org.springframework.web.bind.annotation'

Remove this unused import  
'org.springframework.web.bind.annotation'

Remove this unused import  
'org.springframework.web.client.RestTemplate'

Remove this field injection and use  
constructor injection instead

Intentionality | Not clear

Remove this unused import 'co.edu.escuelaing.cvds.lab7.model.ToDoItem'.  
Unnecessary imports should be removed [java:S1128](#)

Software qualities impacted: Maintainability

☐ Open ☐ Not assigned ☐ Code Smell ☒ Minor

Where is the issue? Why is this an issue? How can I fix it? Activity More info

src/.../co/edu/escuelaing/cvds/lab7/controller/ConfigurationController.java

1 jorge... package co.edu.escuelaing.cvds.lab7.controller;

2

3 import co.edu.escuelaing.cvds.lab7.model.ToDoItem;

4

5 import co.edu.escuelaing.cvds.lab7.service.ConfigurationService;

import org.springframework.beans.factory.annotation.Autowired;

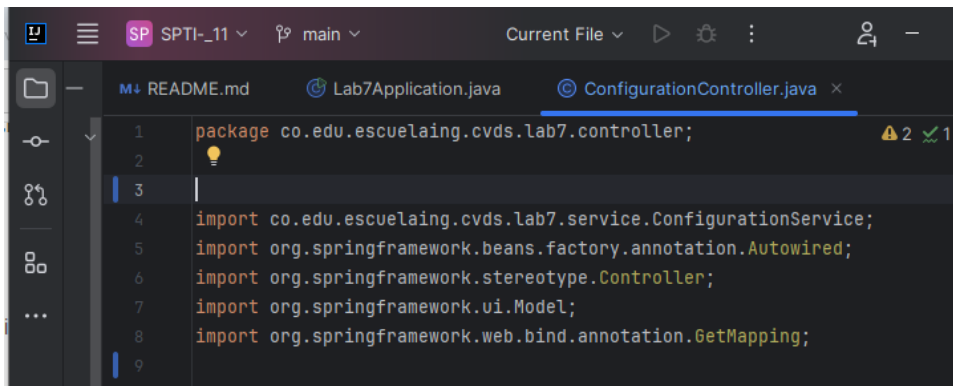
Remove this unused import 'co.edu.escuelaing.cvds.lab7.model.ToDoItem'.

Tags  
unused +

Line affected  
L3

Effort  
1 min

Introduced  
6 months ago



Se nos recomienda eliminar importaciones que no son utilizadas.

6)

```
import co.edu.escuelaing.cvds.lab7.model.Employee;
import co.edu.escuelaing.cvds.lab7.service.EmployeeService;
import jakarta.persistence.Access;
```

Remove this unused import 'jakarta.persistence.Access'.

```
import co.edu.escuelaing.cvds.lab7.model.Employee;
import co.edu.escuelaing.cvds.lab7.service.EmployeeService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
```

Eliminar importaciones no utilizadas.

7)

ne lures

```
}  
// Procesar el formulario y agregar el empleado  
@PostMapping("/agregar")  
public String agregarEmpleado(Employee empleado) {  
    employeeService.addEmployee(empleado);  
    return "redirect:/employees/list"; // Redirigir a la lista de empleados  
}
```

Define a constant instead of duplicating this literal "redirect:/employees/list" 3 times.

```
EmployeeService employeeService; 5 usages  
private static final String ACTION_1 = "redirect:/employees/list";
```

```
}  
// Procesar el formulario y agregar el empleado  
@PostMapping("/agregar") no usages 1 SebSanUwU +1  
public String agregarEmpleado(Employee empleado) {  
    employeeService.addEmployee(empleado);  
    return ACTION_1; // Redirigir a la lista de empleados  
}  
  
@GetMapping("/modificar/{employeeId}") no usages 1 SebSanUwU  
public String mostrarFormularioModificarEmpleado(@PathVariable Long employeeId, @ModelAttribute Employee model) {  
    model.addAttribute("attributeName", "empleado", employeeService.getEmpleadoById(employeeId));  
    return "updateEmployee";  
}  
  
@PostMapping("/modificar/{employeeId}") no usages 1 SebSanUwU +1  
public String modificarEmpleado(@PathVariable Long employeeId, @ModelAttribute Employee model) {  
    System.out.println(updatedEmployee);  
    employeeService.updateEmployee(updatedEmployee);  
    return ACTION_1;  
}  
  
@PostMapping("/eliminar/{employeeId}") no usages 1 SebSanUwU +1  
public String eliminarEmpleado(@PathVariable Long employeeId) {  
    employeeService.deleteUser(employeeId);  
    return ACTION_1;  
}
```

Definir una constante para no duplicar texto.

8)

```
@PostMapping("/modificar/{employeeId}")
public String ModificarEmpleado(@PathVariable Long employeeId, @ModelAttribute
employee updatedEmployee) {
```

---

Rename this method name to match the regular expression `^[a-z][a-zA-Z0-9]*$`.

---

```
    @PostMapping("/modificar/{employeeId}")
    public String modificarEmpleado(@PathV
```

Colocar en minúsculas la M.

9)

```
    if (configurationRepository.findByPropiedad(configuration
        .getPropiedad()).size() == 0) {
```

Use isEmpty() to check whether the collection is empty or not.

```
public Configuration updateConfiguration(Configuration configuration) {
    if (configurationRepository.findByPropiedad(configuration.getPropiedad()).isEmpty()) {
        return configurationRepository.save(configuration);
    }

    return null;
}
```

Utilizamos isEmpty() en lugar de == 0.

10)

```
<main>
```

```
<h1 th:text="'Gran premio: ' + ${premio} + '!'" />
```

Headings must have content and the content must be accessible by a screen reader.

```
<h1> th:text="'Gran premio: ' + ${premio} + '!'"</h1>
```

Ponemos dentro de la etiqueta el texto.

11)

```
@Test
```

```
void contextLoads() {
```

Add at least one assertion to this test case.

```
✚ Jorge Ulises Useche Cuellar *
```

```
@Test
```

```
void contextLoads() {
```

```
    assertTrue( condition: true, message: "Spring context loaded successfully");  
}
```

Agregamos una aserción al método para que no quede vacío.

12)



1	jorge...	<!DOCTYPE HTML>
2		<html xmlns:th="http://www.thymeleaf.org">

Add a <title> tag to this page.

```

<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title>Titulo</title>
</head>
<th:block th:replace="fragments/general.html::headerfiles('Example Page')"></th:block>
<body>

```

Agregamos un título a la pagina.

13)

```

public void deleteUser(Long userId) {
    employeeRepository.deleteById(Long.valueOf(userId));
}

```

Remove the boxing to "Long"; The argument is already of the same type.

```

1 usage  SebSanUwU *
public void deleteUser(Long userId) {
    employeeRepository.deleteById(userId);
}

```

Eliminamos la asignación long ya que el argumento ya es de tipo long.

14)

```
<!DOCTYPE HTML>
```

```
<html xmlns:th="http://www.thymeleaf.org">
```

Add "lang" and/or "xml:lang" attributes to this "<html>" element

```
<!DOCTYPE HTML>
```

```
<html xmlns:th="http://www.thymeleaf.org" lang="en" xml:lang="en">
```

Agregamos lang a la etiqueta html.

## Security Hotspot

**Make sure allowing safe and unsafe HTTP methods is safe here.** [🔗](#)

Allowing both safe and unsafe HTTP methods is security-sensitive [java:S3752](#)

**Review priority:**

🔴 High

**Category:**

Cross-Site Request Forgery (CSRF)

```
@RequestMapping(value = "/to-do-item/{id}", method = RequestMethod.GET)
public String getUser(@PathVariable Integer id, Model model) {
    String uri = "https://jsonplaceholder.typicode.com/todos/" + Integer.toString(id);
    RestTemplate restTemplate = new RestTemplate();

    ToDoItem toDoItem = restTemplate.getForObject(uri, ToDoItem.class);

    model.addAttribute(attributeName: "toDoItem", toDoItem);
    return "to-do-item";
}
```