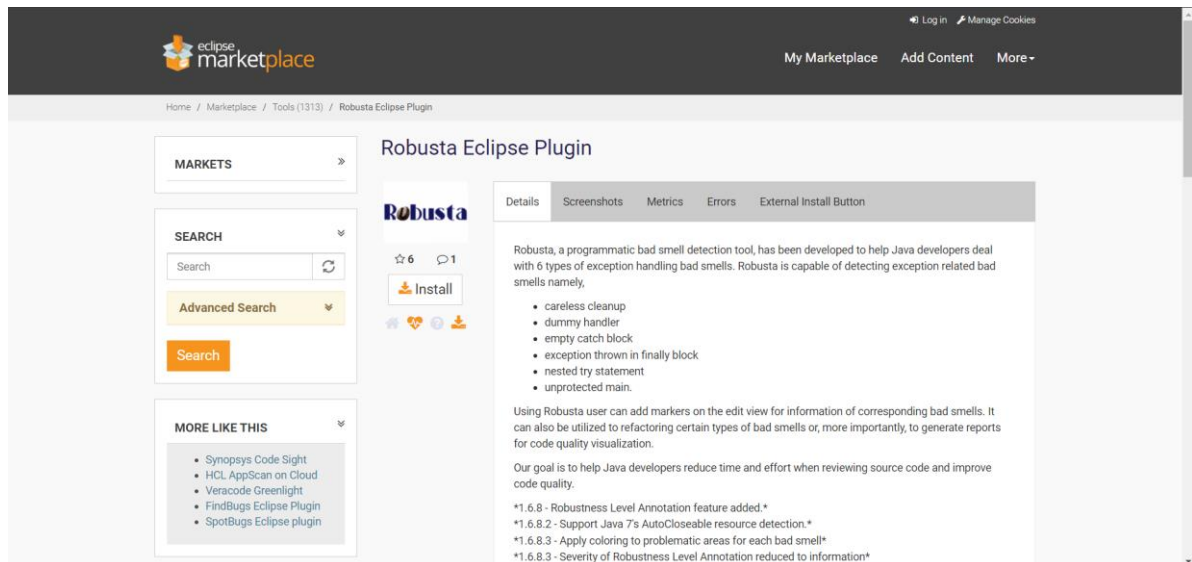


GUÍA DE INSTALACIÓN PLUGIN ROBUSTA EN ECLIPSE

Grupo 3

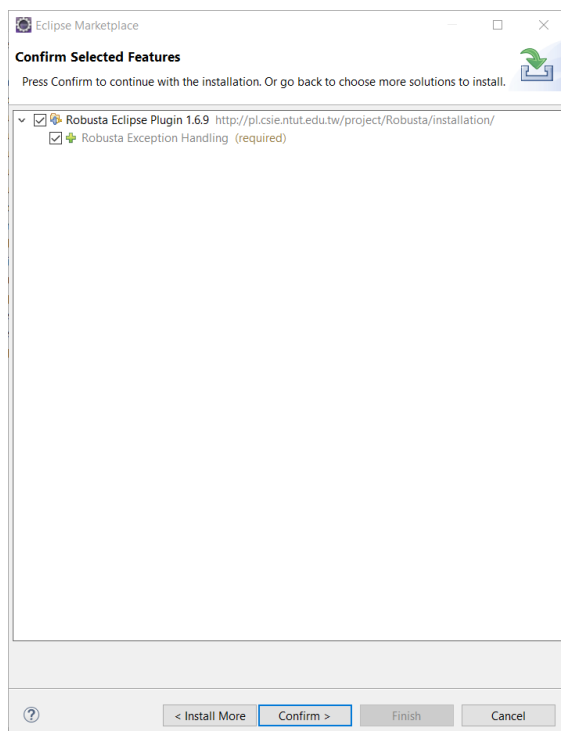
Paso 1:

Entrar en la siguiente página: <https://marketplace.eclipse.org/content/robusta-eclipse-plugin>



Paso 2:

Arrastrar el botón central en el que pone **Install** hasta Eclipse. Aparecerá esta ventana:



Paso 3:

Pulsar *confirm*, luego *finish*, y después a *install anyway*. A continuación te pedirá reiniciar eclipse, así que reiniciamos eclipse.

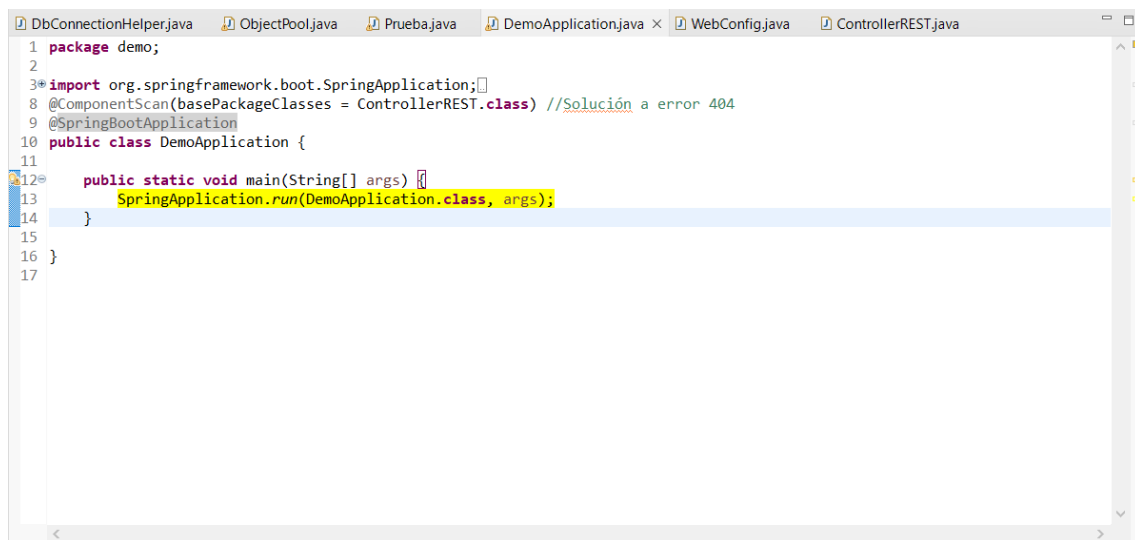
Paso 4:

Para hacer funcionar Robusta en un proyecto, deberemos hacer click derecho en el proyecto que queremos, buscar Robusta, y pulsar en Detect Smells.

Click derecho en proyecto -> Robusta -> Detect Smells

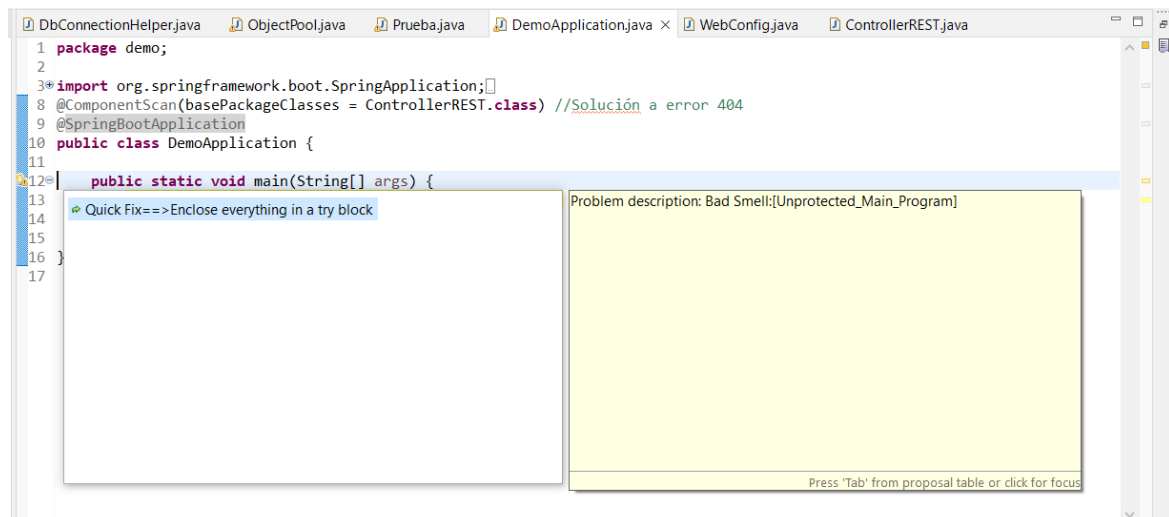
Paso 5:

A continuación se nos resaltara en amarillo las línea de código en las que hay code smells.



```
1 package demo;
2
3 import org.springframework.boot.SpringApplication;
4 @ComponentScan(basePackageClasses = ControllerREST.class) //Solución a error 404
5 @SpringBootApplication
6 public class DemoApplication {
7
8     public static void main(String[] args) {
9         SpringApplication.run(DemoApplication.class, args);
10    }
11 }
12
13
14
15
16
17
```

A continuación, pulsaremos sobre la bombilla y se nos indicará como resolver el code smell.



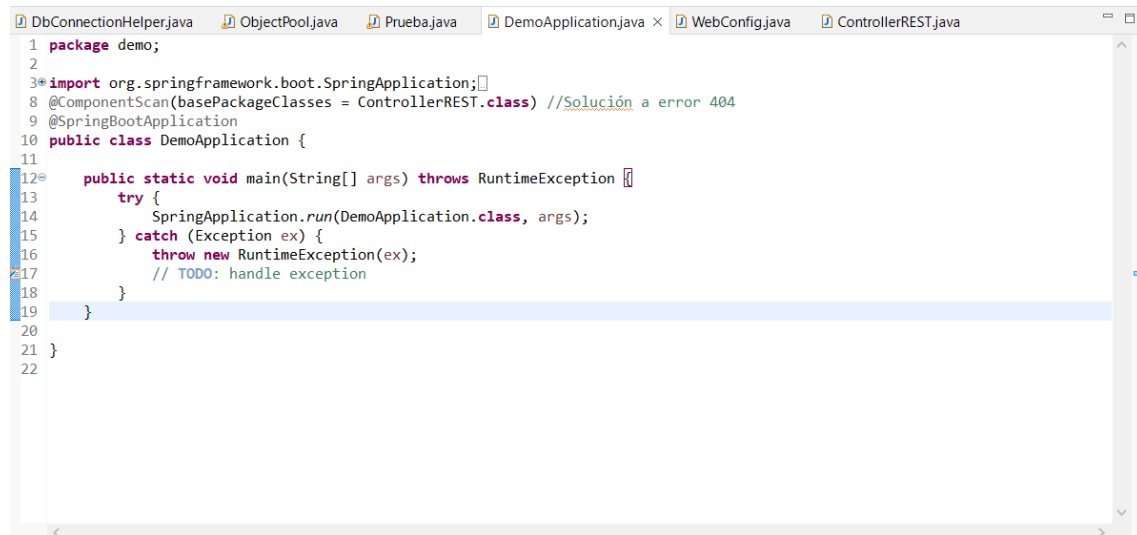
```
1 package demo;
2
3 import org.springframework.boot.SpringApplication;
4 @ComponentScan(basePackageClasses = ControllerREST.class) //Solución a error 404
5 @SpringBootApplication
6 public class DemoApplication {
7
8     public static void main(String[] args) {
9         SpringApplication.run(DemoApplication.class, args);
10    }
11 }
12
13
14
15
16
17
```

Quick Fix==>Enclose everything in a try block

Problem description: Bad Smell:[Unprotected_Main_Program]

Press "Tab" from proposal table or click for focus

Resolvemos el code smell y el resaltado amarillo desaparece.



```
1 package demo;
2
3 import org.springframework.boot.SpringApplication;
4
5 @ComponentScan(basePackageClasses = ControllerREST.class) //Solución a error 404
6 @SpringBootApplication
7 public class DemoApplication {
8
9     public static void main(String[] args) throws RuntimeException {
10         try {
11             SpringApplication.run(DemoApplication.class, args);
12         } catch (Exception ex) {
13             throw new RuntimeException(ex);
14             // TODO: handle exception
15         }
16     }
17 }
18
19
20
21
22
```

The screenshot shows an IDE window with several tabs: DbConnectionHelper.java, ObjectPool.java, Pruebajava, DemoApplication.java (active), WebConfig.java, and ControllerREST.java. The active file, DemoApplication.java, contains the following code. A yellow highlight is visible on line 12, indicating a code smell that has been resolved. The code is a Spring Boot application with a single package 'demo'. It uses Spring Boot's auto-configuration and component scanning. The main method is a static void method that throws a RuntimeException. It contains a try-catch block that calls SpringApplication.run() and catches any Exception, throwing a new RuntimeException. A TODO comment is present in the catch block.