

## **CLEAN CODE**

### **Presentado por:**

Nicolas Quezada Mora - [nquezada@unal.edu.co](mailto:nquezada@unal.edu.co)

Sharick Yelixa Torres Monroy - [shtorres@unal.edu.co](mailto:shtorres@unal.edu.co)

Laura Sofia Vargas Rodriguez - [lavargasro@unal.edu.co](mailto:lavargasro@unal.edu.co)

Jeronimo Bermudez Hernandez- [jebermudez@unal.edu.co](mailto:jebermudez@unal.edu.co)

### **Profesor:**

Oscar Eduardo Alvarez Rodriguez - [oalvarezr@unal.edu.co](mailto:oalvarezr@unal.edu.co)

Viernes 31 de Enero



**Universidad Nacional de Colombia**

**Facultad de Ingeniería**

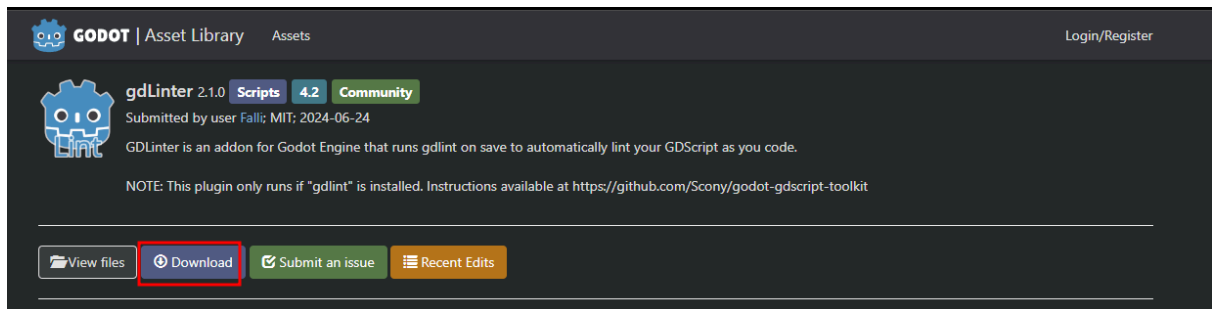
**Departamento de Industrial y Sistemas**

**2025**

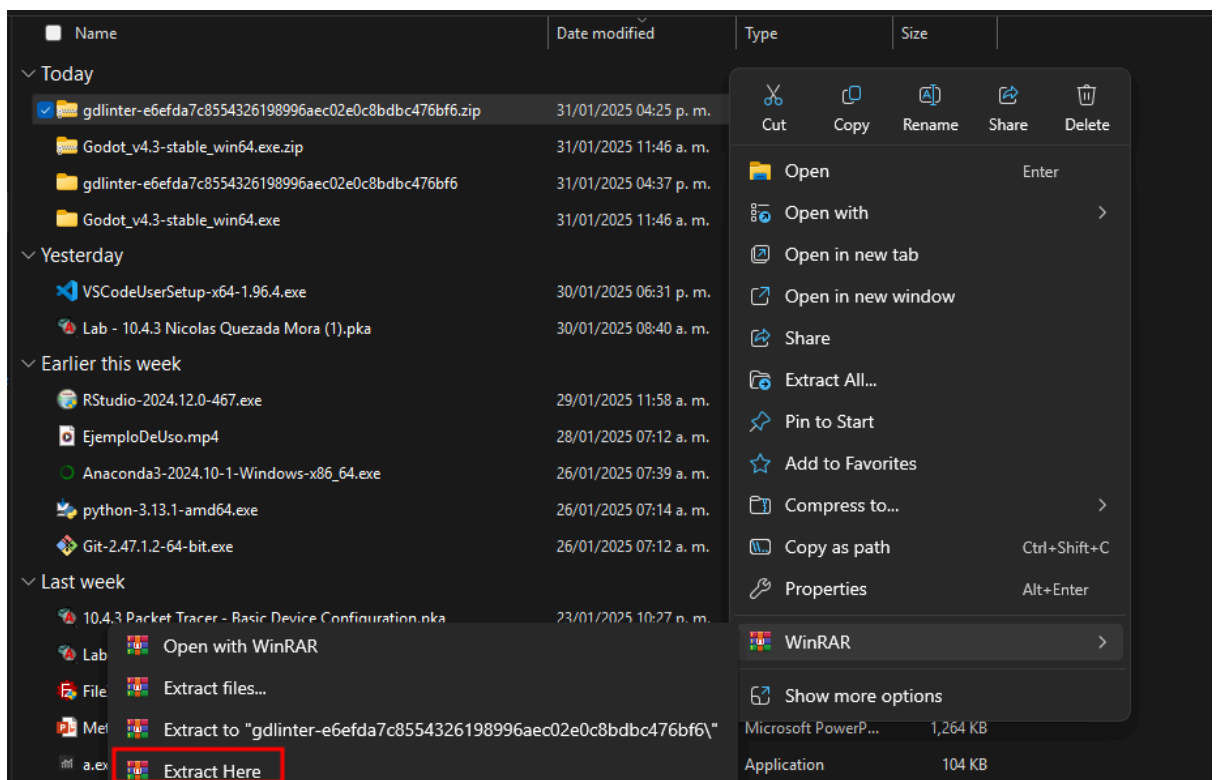
## 1. Clean code:

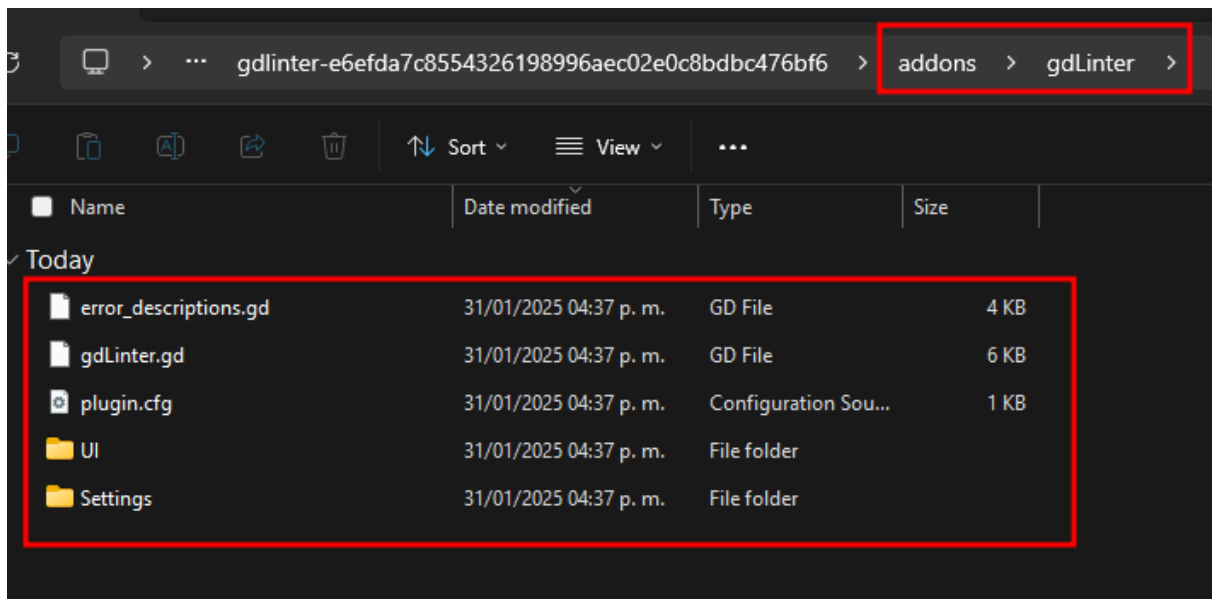
Para el linter de nuestro programa, seleccionamos el plugin para Godot denominado gdLinter, el cual se encarga de analizar y depurar el código. Este plugin revisa el código en busca de inconsistencias, errores de sintaxis y áreas de mejora, lo que facilita la optimización del rendimiento y la adherencia a las buenas prácticas de programación.

En el primer paso, lo descargamos:

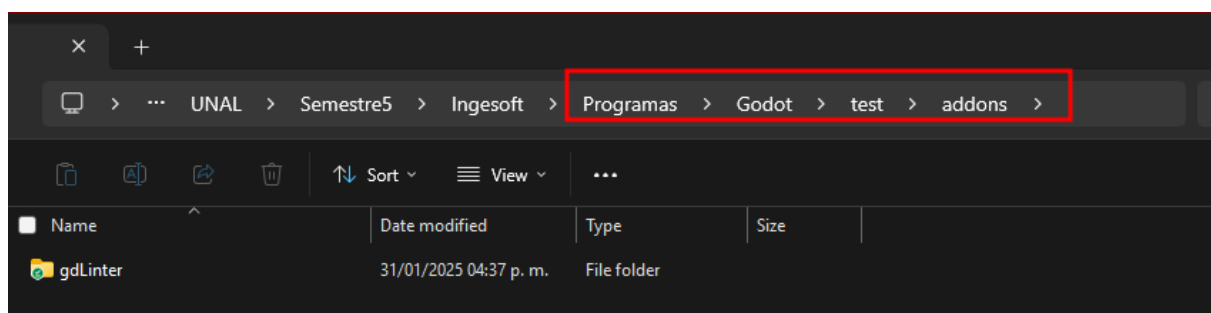


Luego se extrae:

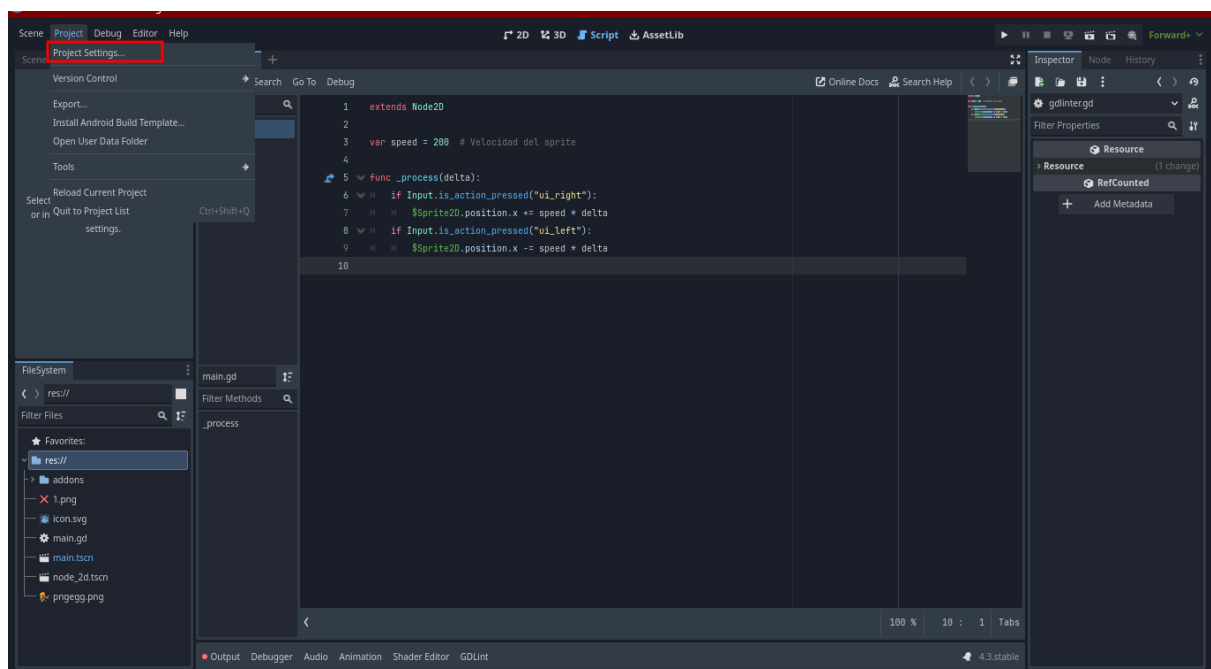


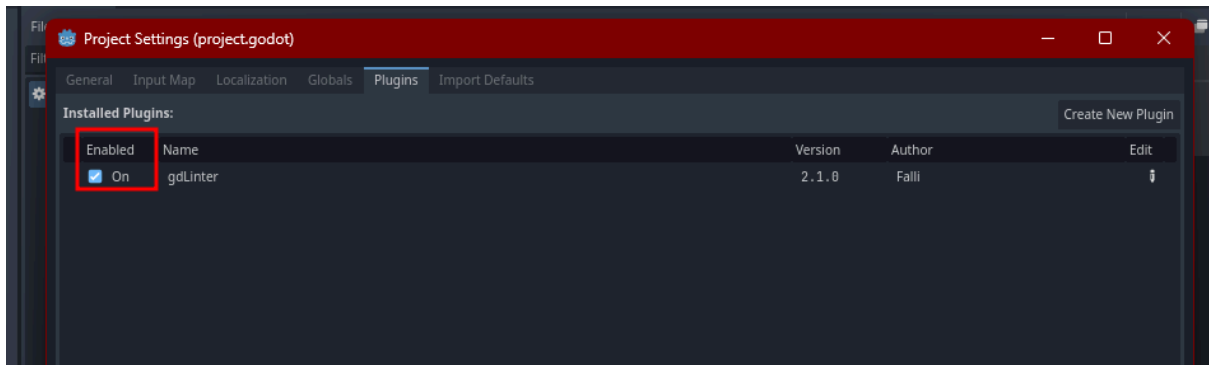


Después se incorpora en la carpeta de addons del proyecto en el que estamos trabajando:

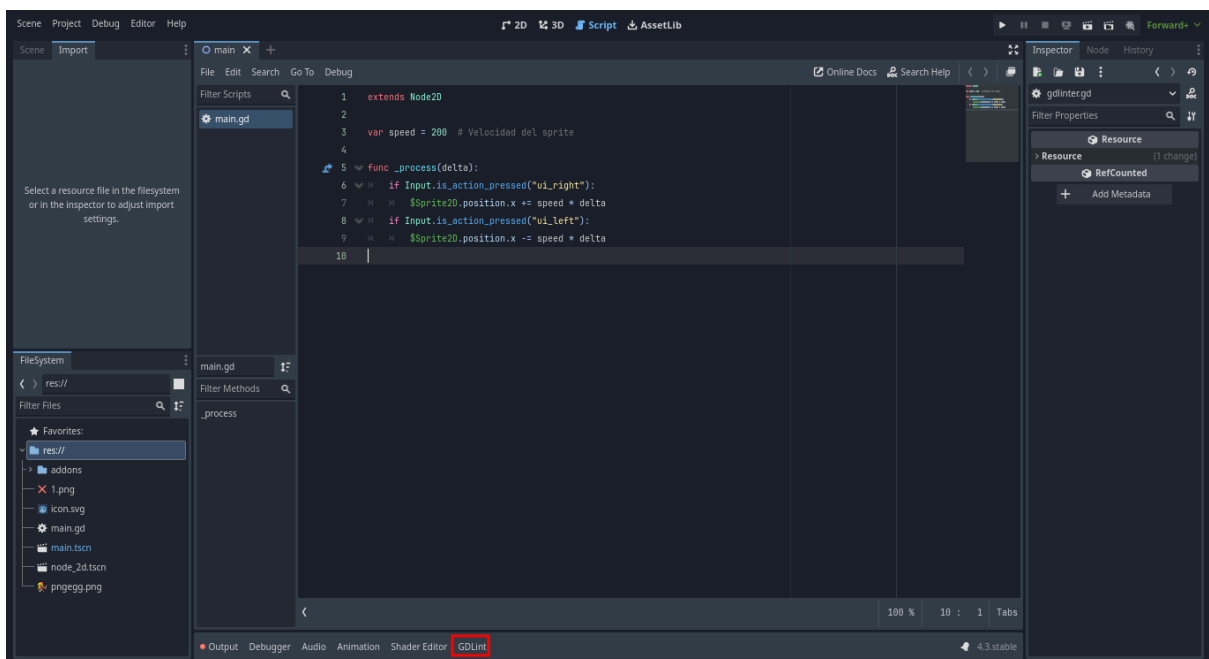


Una vez realizado este procedimiento, lo activamos desde la pestaña de plugins:

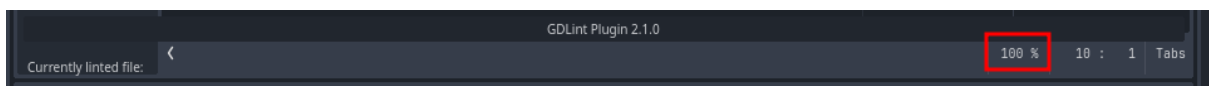




Se selecciona en la parte de abajo del entorno:



Como se puede observar, en el código inicial no se detectaron errores en el código inicial que tenemos:



A continuación, se muestra una imagen que ilustra cómo se vería el entorno en caso de que se presentara algún error:

