

Informe código limpio - CLEAN CODE

Ingeniería de Software I



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Facultad de Ingeniería
Departamento de Ingeniería de Sistemas e Industrial

Presentado por:

Nicolas Quezada Mora - nquezada@unal.edu.co

Sharick Yelixa Torres Monroy - shtorres@unal.edu.co

Laura Sofia Vargas Rodriguez - lavargasro@unal.edu.co

Jeronimo Bermudez Hernandez - jebermudez@unal.edu.co

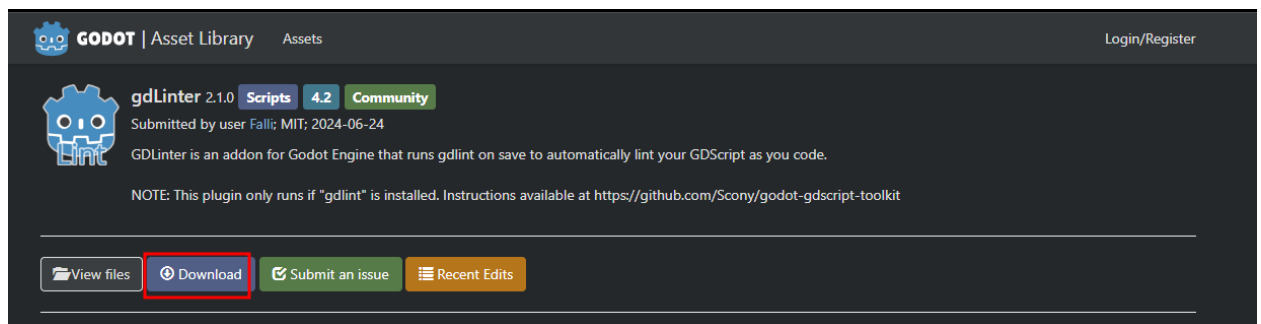
Docente:

Oscar Eduardo Alvarez Rodriguez - oalvarezr@unal.edu.co

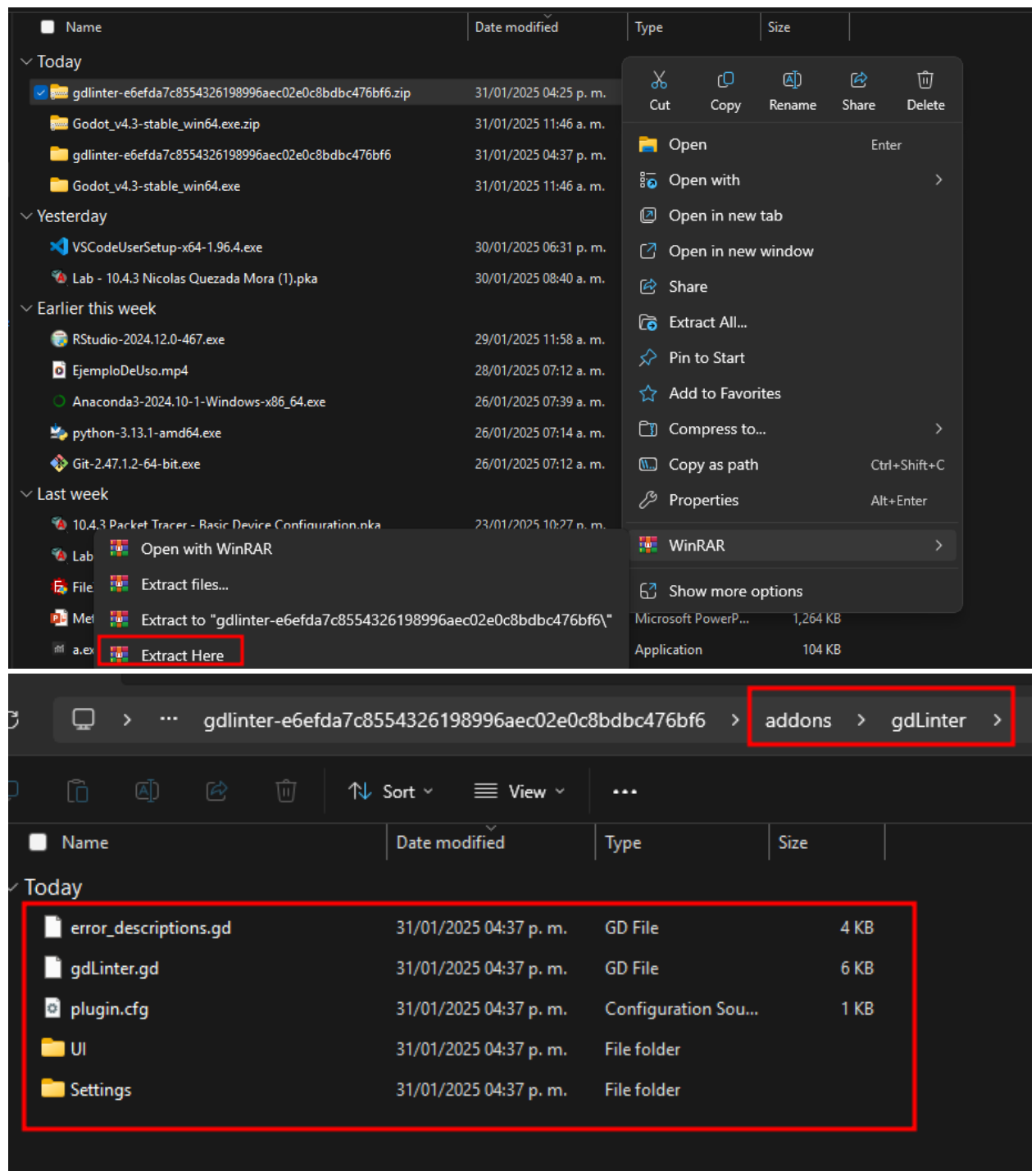
1. Clean code:

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.

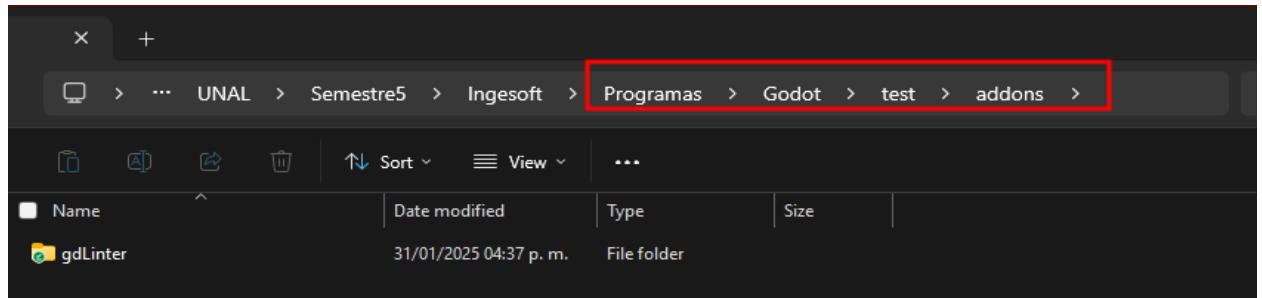
En el primer paso, lo descargamos:



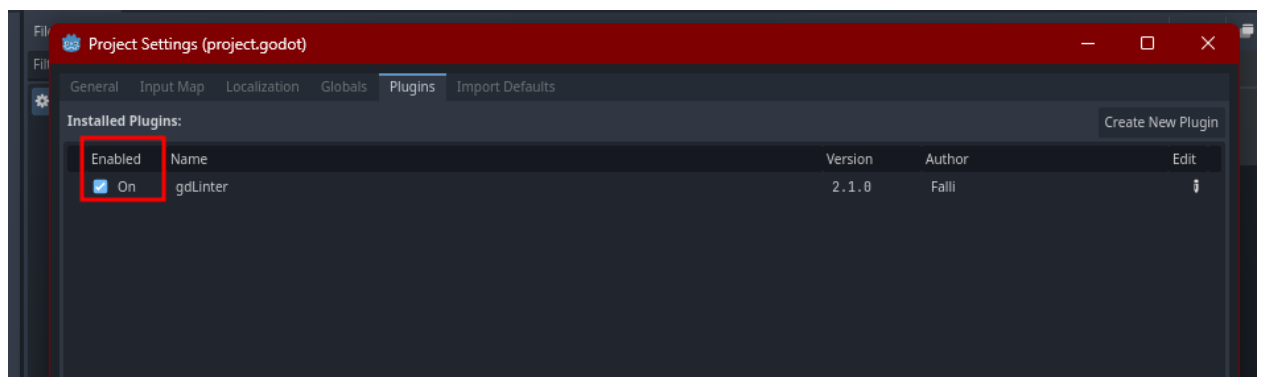
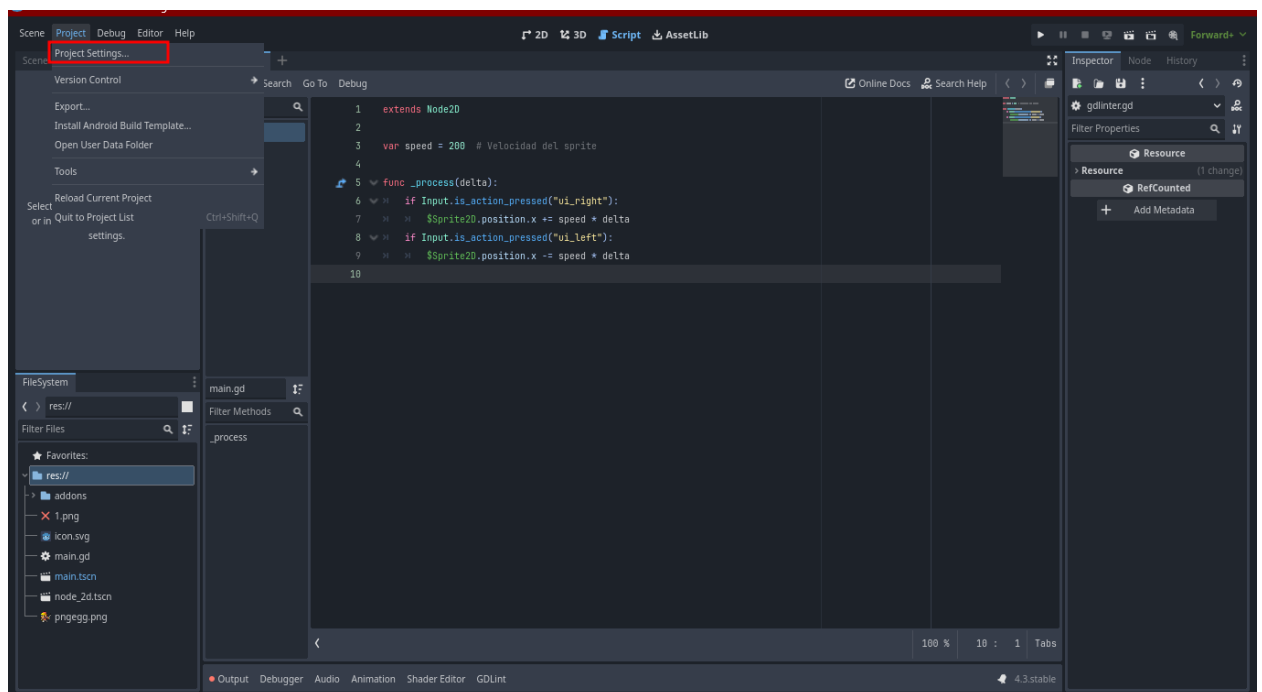
Luego se extrae:



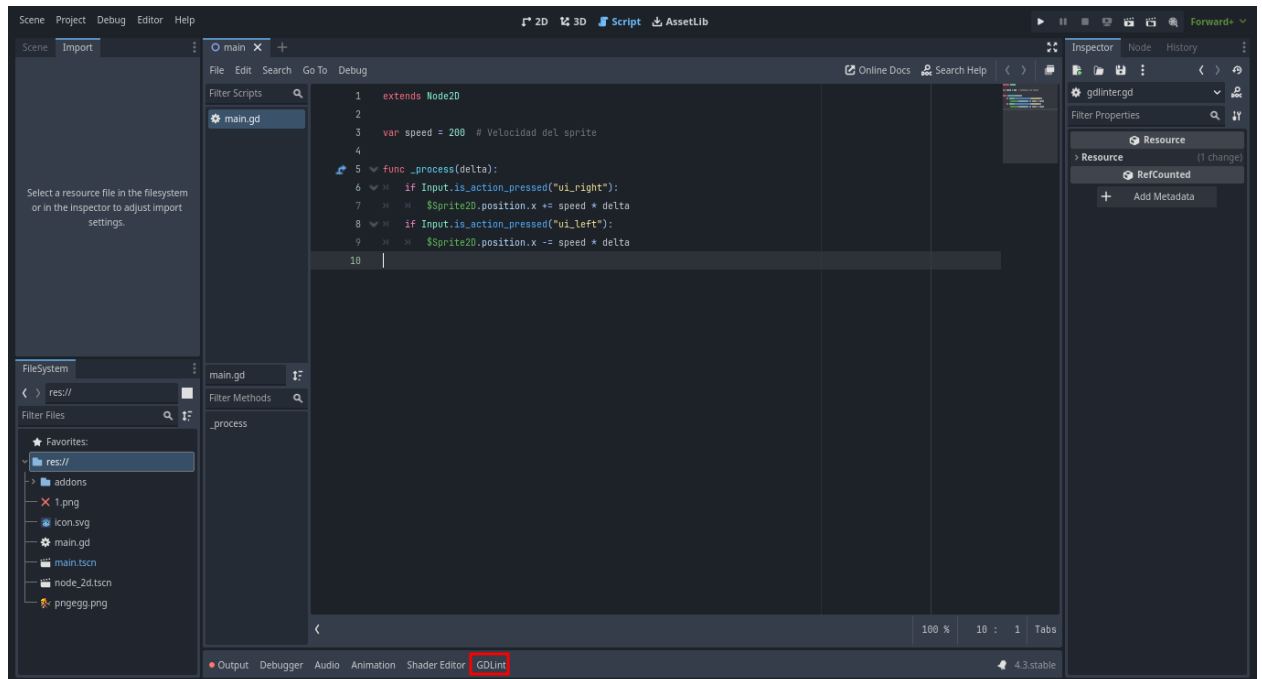
Después se incorpora en la carpeta de addons del proyecto en el que estamos trabajando:



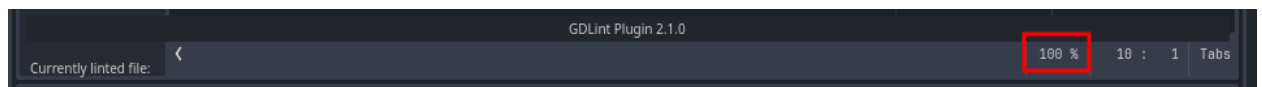
Una vez realizado este procedimiento, lo activamos desde la pestaña de plugins:



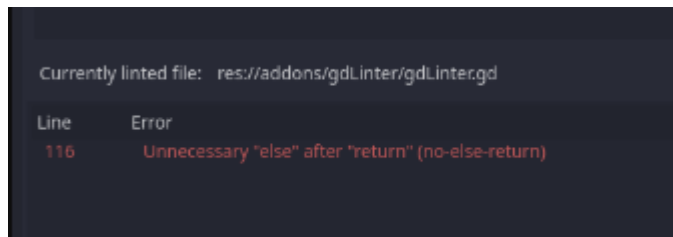
Se selecciona en la parte de abajo del entorno:



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:



Comentarios acerca del código en equipo:

- **Sharick Torres:**

Considero que el código general refleja un esfuerzo coordinado, se nota que cada uno puso de su parte siendo coherentes con los aportes de los demás, aunque aún falta parte del código, hasta el momento hemos tratado de mantener un estándar común que facilite la comprensión del resto y futuras modificaciones. Sin embargo, creo que podríamos hacer mejor uso de los comentarios.

- **Jeronimo Bermudez:**

Desde mi perspectiva, el código demuestra una buena organización y es fruto de un excelente trabajo en equipo, cada uno tenía claro su rol y eso permitió que cada parte se desarrollara de forma autónoma sin confusiones, la división clara de funciones hizo que el avance fuera ordenado y acorde a lo que queríamos lograr

- **Nicolas Quezada:**

En general, considero que la estructura lograda es uniforme y funcional, y se ajusta a los estándares que acordamos, me gustó mucho lo claro y legible que resultó el código, cada segmento cumple una función específica, estas características hicieron que fuera sencillo entender qué hace cada parte.

- **Sofia Vargas:**

A mi parecer, el código que hemos desarrollado resulta muy claro y sigue buenas prácticas de programación, lo que se nota en la organización y calidad de cada sección, además, he aprendido mucho al ver el trabajo de mis compañeros, sus enfoques y técnicas me han ayudado a mejorar mi propia forma de programar, la experiencia compartida ha sido muy enriquecedora para mí.