

**Análisis estático**

Presentado por:

Santiago Andrés Arteaga

Santiago Hurtado

Asignatura:

Programación orientada a objetos

Grupo:

POOB-4

Presentado a:

Juan Sebastián Frasica Galeano

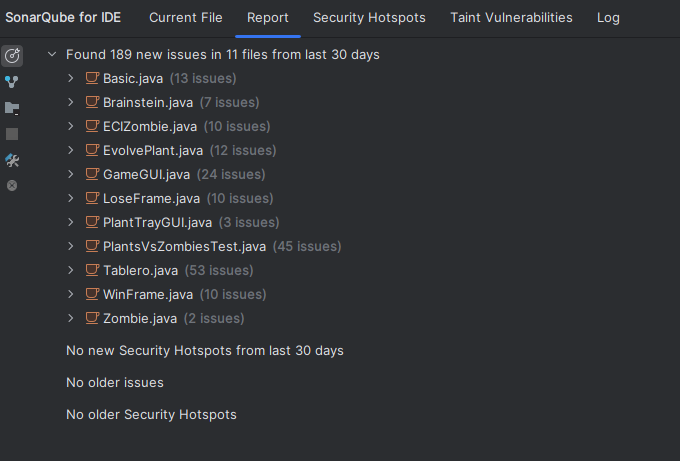
María Irma Diaz Rozo

Utilizando la extensión SonarQube de IntelliJ se generó el siguiente reporte estático sobre el código de Plantas vs Zombies correspondiente al proyecto final de programación orientada a objetos 2024-2.

El análisis se realizó sobre las clases del proyecto en cuestión, tanto en la capa de dominio como de presentación  
  
Se buscaron las siguientes violaciones de código:

* BestPractices
* CodeStyle
* Design
* Documentation
* ErrorProne
* Multithreading
* Performance

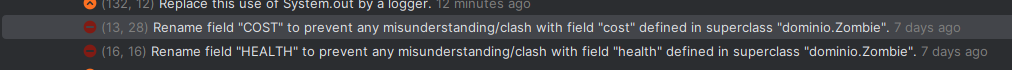
Se obtuvieron los siguientes resultados:

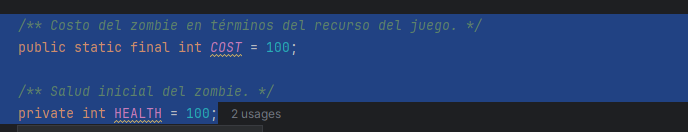


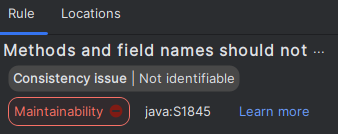
En total se encontraron 189 violaciones entre las cuales se resaltan las más críticas:

Para la capa de dominio:

Basic:



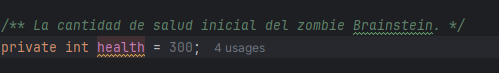




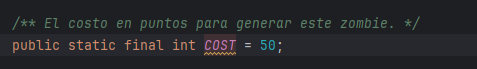
Se pide renombrar para no confundir HEALTH y COST con health y cost de la clase padre Zombie  
  
El mismo problema se encontró en los siguientes zombies:

Brainstein:





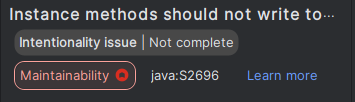




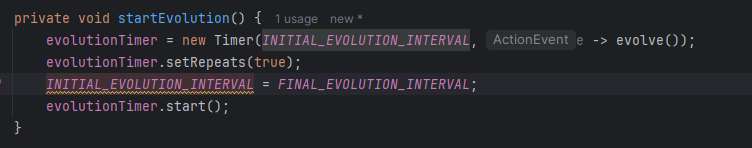
ECIZombie:

Mismos errores anteriores sobre el costo y la salud

EvolvePlant:





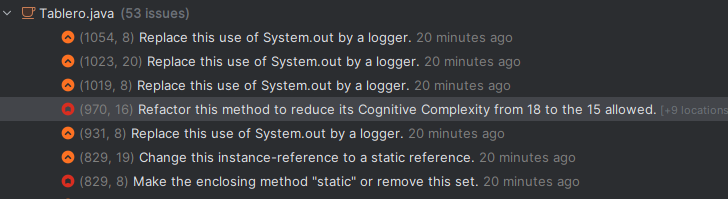


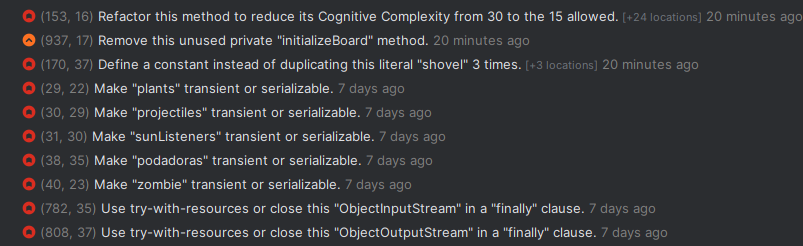
Se debe hacer que la evolución sea un método estático porque la evolución es un método propio de la clase EvolvePlant.

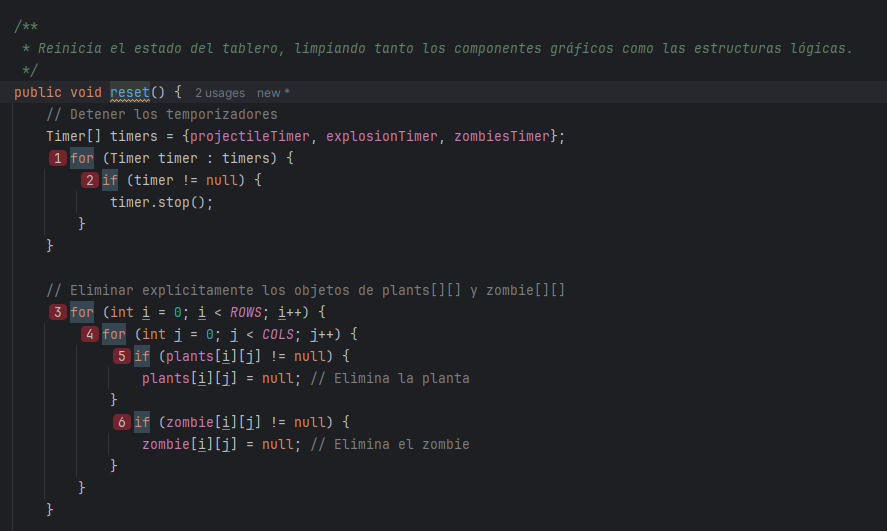


Se están imprimiendo muchos mensajes que pueden refactorizarse como constantes y ser llamadas.

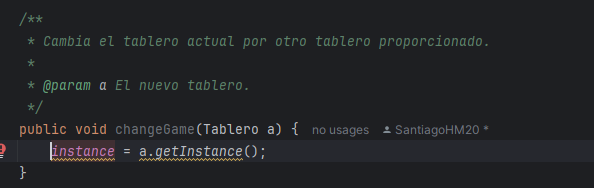
Tablero:



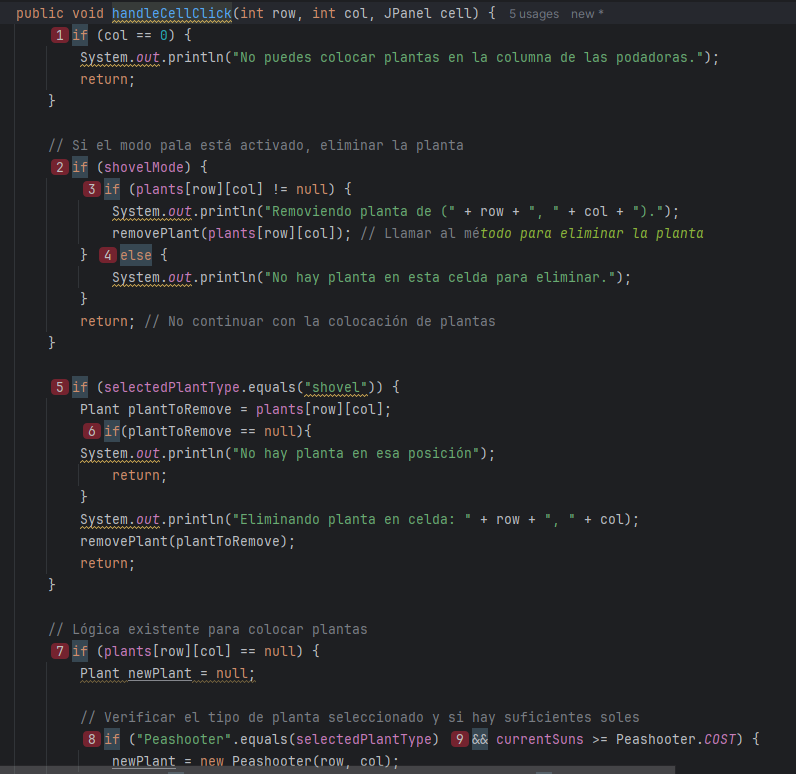




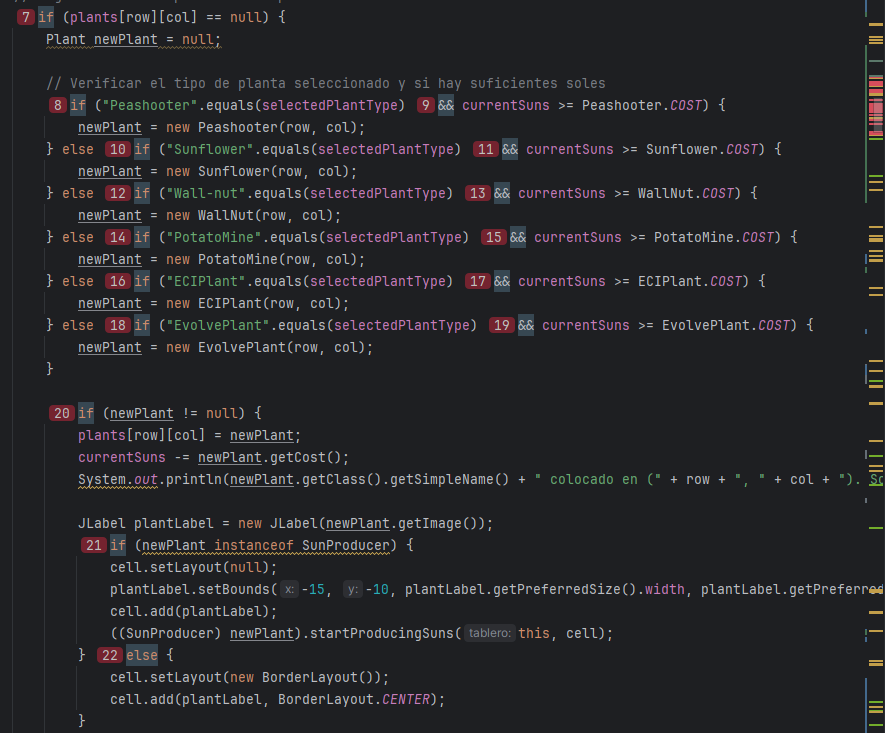
El código de reset debe ser refactorizado para manejar buenas prácticas



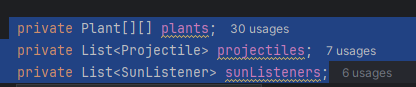
La instancia de tablero debe ser estática



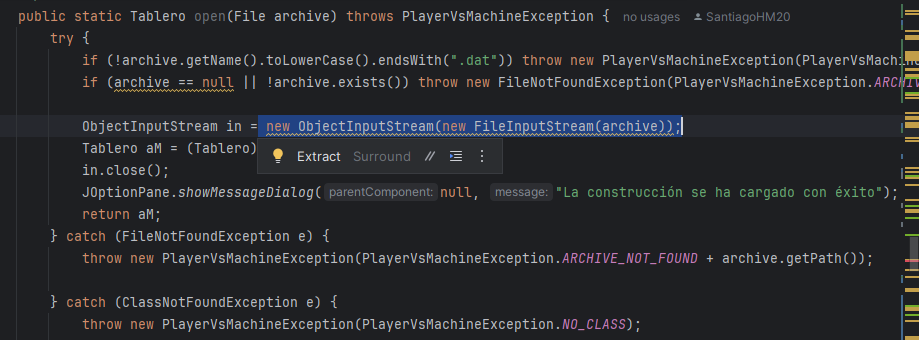
Este método también debe ser refactorizado para disminuir complejidad y manejar buenas prácticas de código.

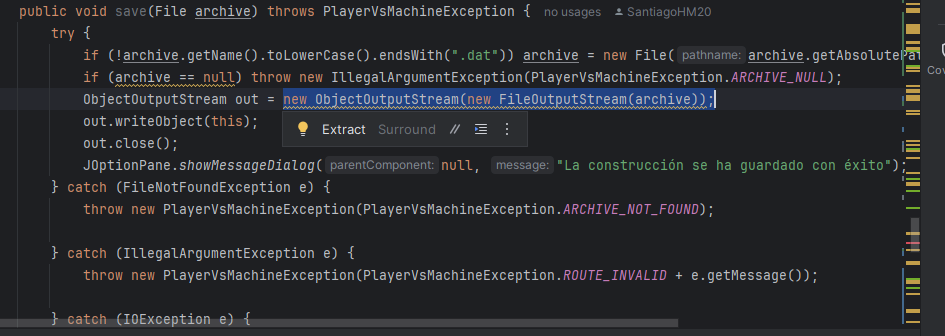


Mismo caso que el anterior.



Es bueno manejar estos atributos como serializables

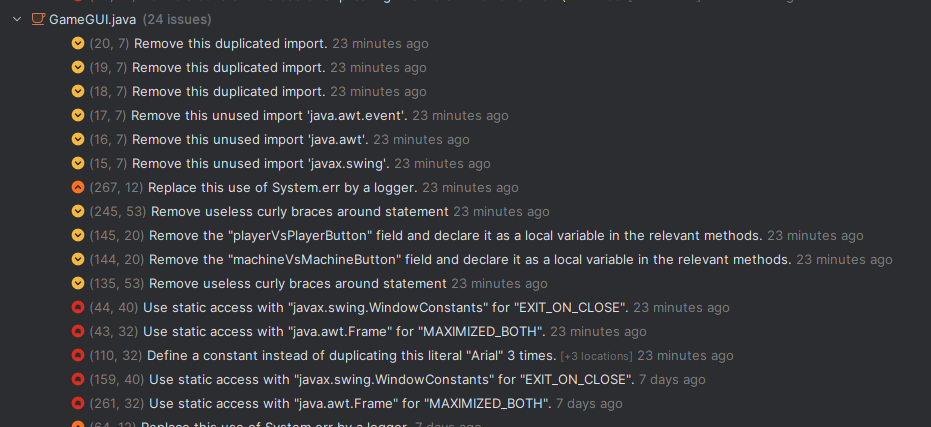


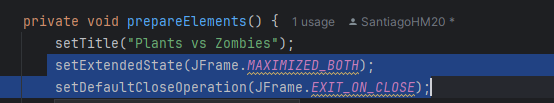


Para crear una nueva instancia es mejor hacerlo en un bloque finally.

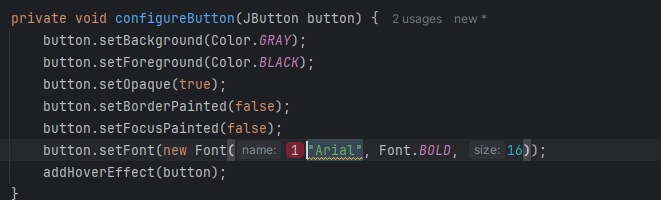
Para la capa de presentación:

GameGUI:

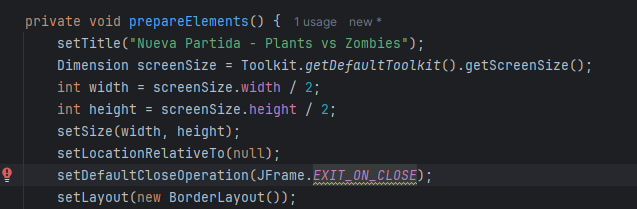




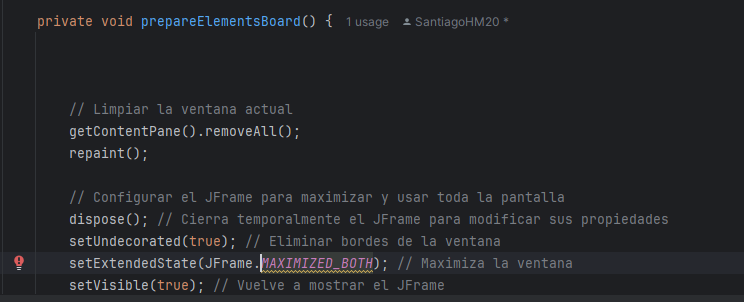
Usar un acceso estático a estos llamados



Definir constante

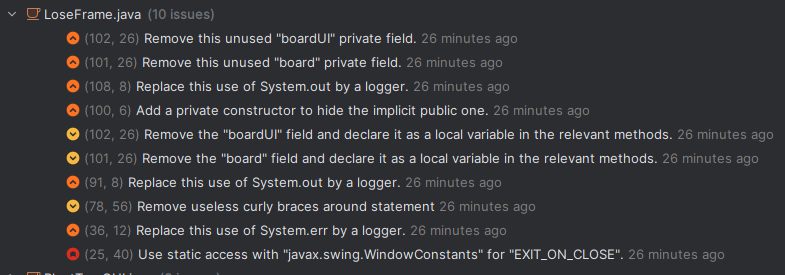


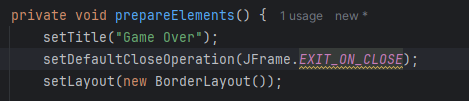
Acceso estatico



Acceso estatico

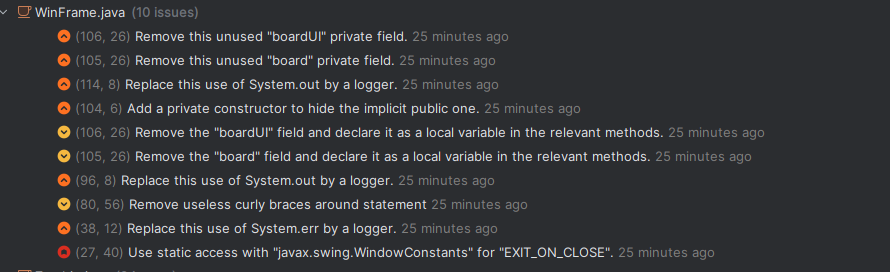
LoseFrame:

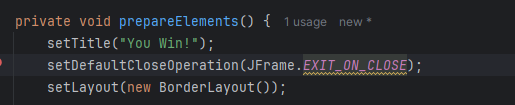




Acceso estatico

WinFrame:





Acceso estático

El análisis estático realizado utilizando SonarQube nos permitió identificar un montón de errores a corregir para pulir el proyecto.