

**Universidad Nacional Mayor de San Marcos**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**Refactorización Patrones veranito Chesss**

**Curso:** Gestión de Mantenimiento de Software

**Profesor:** Inga Lopez, Wilder Inga

**Ciclo:** 9no

**Escuela:** Ingeniería de Software

**Integrante:**

* Suarez Hernandez, Kenny Joel



**Lima - Perú**

**Agosto, 2022**

# 

[**Definición del Requerimiento 4**](#_aaqumuaci57g) **3**

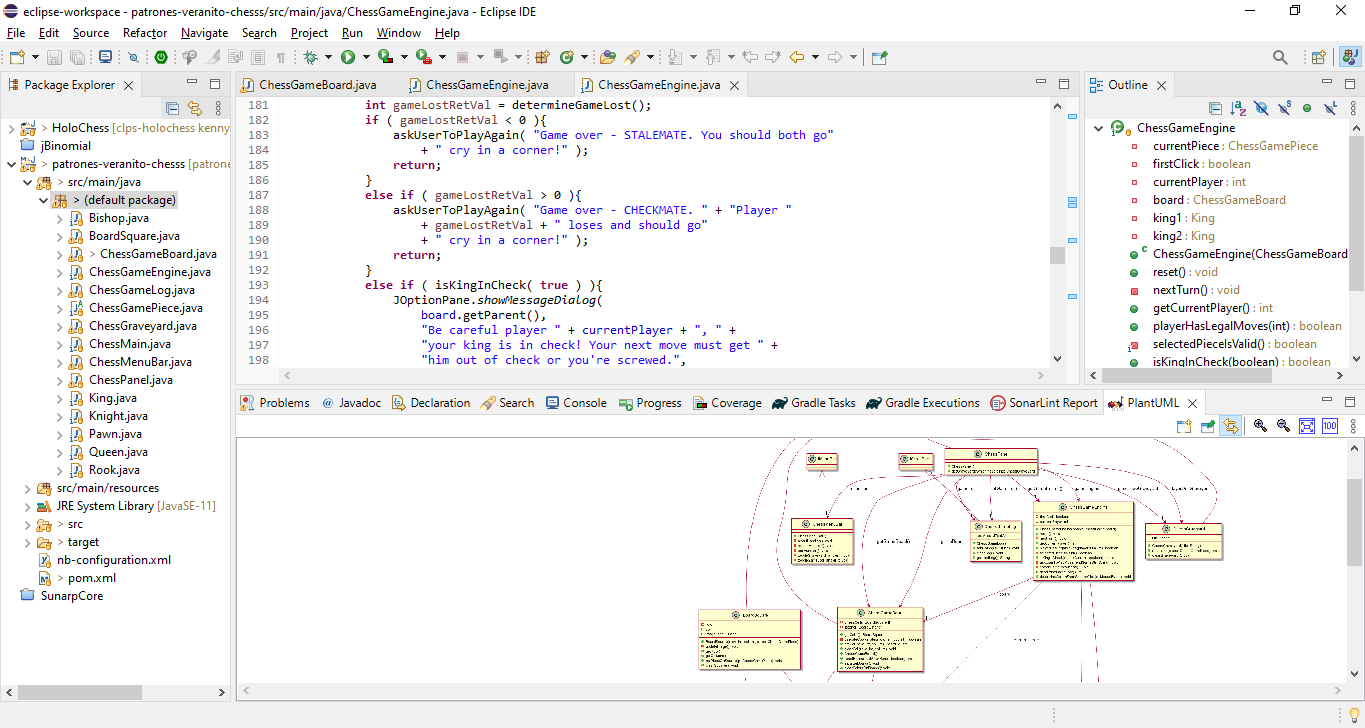
[**Definición del Requerimiento 5**](#_e3qyh5y48rtr) **5**

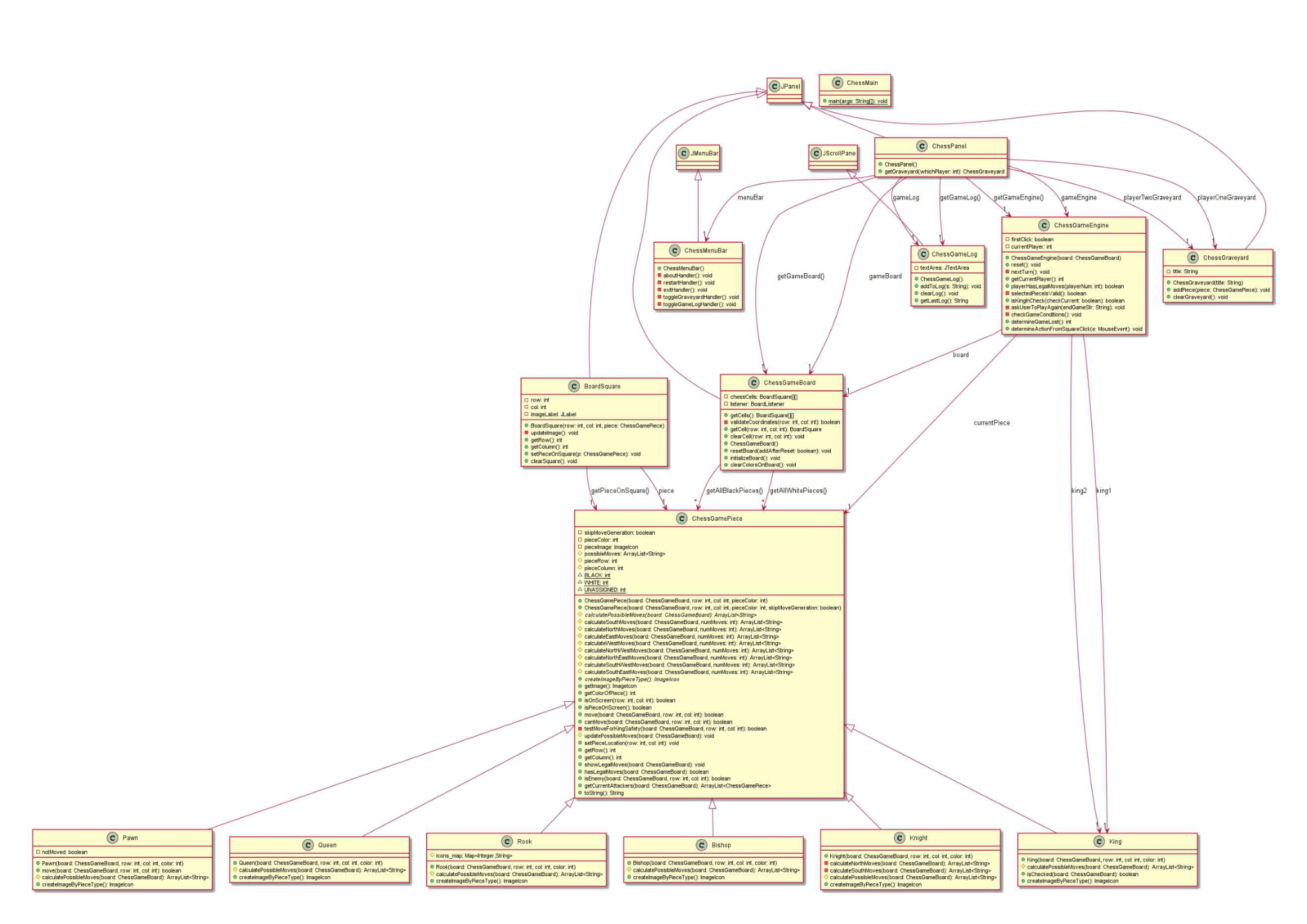
[Repartición de los errores encontrados](#_kfk68t5ffk80) 6

[Resultado final](#_4jf8k8x9k1kt) 7

# 

Los pasos utilizados para realizar la refactorización fueron los siguientes: En primer lugar, se utilizó la función para la creación automática de diagrama de clases mediante código fuente del software PlantUML que cuenta con un plugin que permite la integración con Eclipse IDE. Posteriormente, se hizo uso de un software de evaluación de código fuente para detectar errores relevantes para la refactorización. Después, se realizó una corrección de estos y se procedió a generar el diagrama de clases del producto resultante.



Se obtuvo el siguiente diagrama de clases:

# 

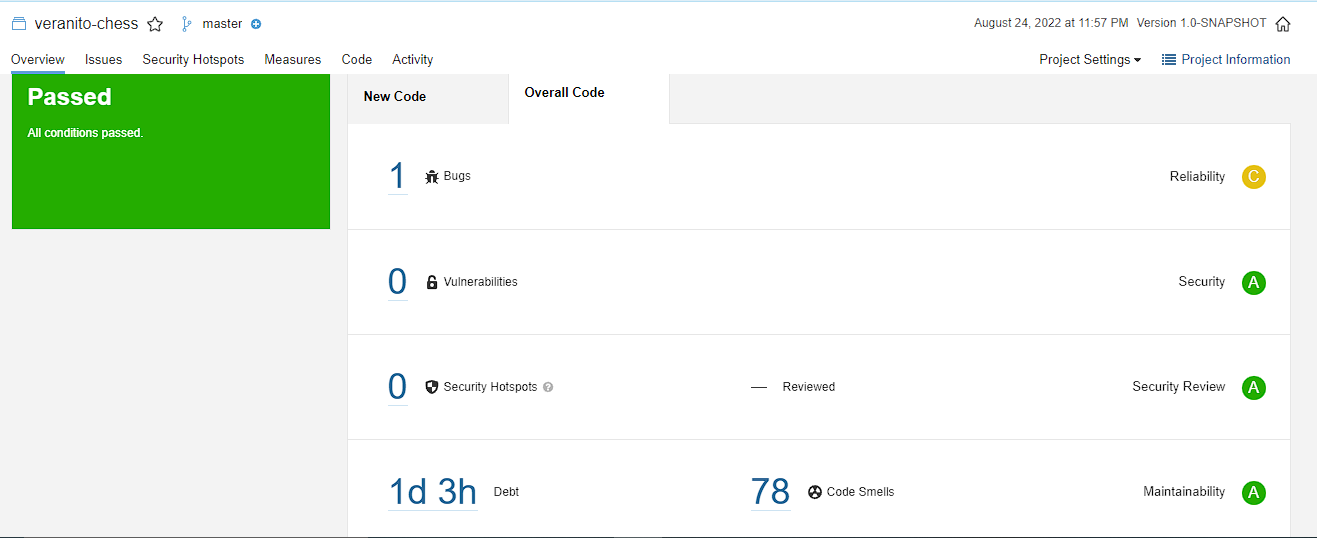
# 

# 

# Definición del Requerimiento 5

RQ5: ¿Qué tipos de datasets son usados por practicantes de software para realizar la validación de la refactorización?

Se usó la herramienta de evaluación de código fuente SonarQube para realizar la detección de errores dentro del proyecto a refactorizar. Se obtuvo el siguiente resultado después de realizar la primera pasada:

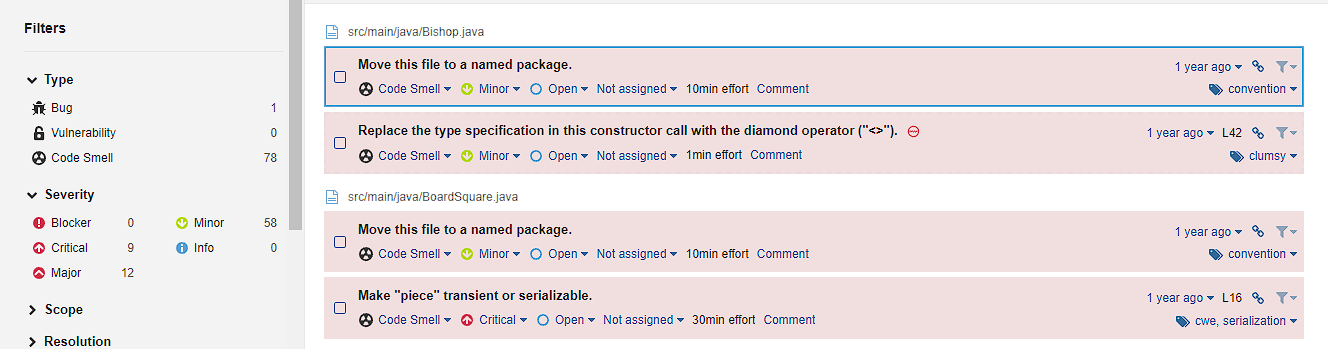


Podemos observar que dentro del sector de mantenibilidad existen 78 Code Smells que generan una deuda técnica de 1 día con 3 horas. Otro punto importante a observar es que en el apartado de confiabilidad se muestra que todavía existe un bug existente. Para finalizar, hay un total de 3.2% de líneas duplicadas dentro del proyecto analizado.

La revisión y atención de estos problemas será realizada por cada uno de los integrantes del grupo. Acontinuación, se presentará el aporte de cada uno de los integrantes con respecto a la atención de errores detectados mediante la herramienta SonarQube.

## Errores encontrados

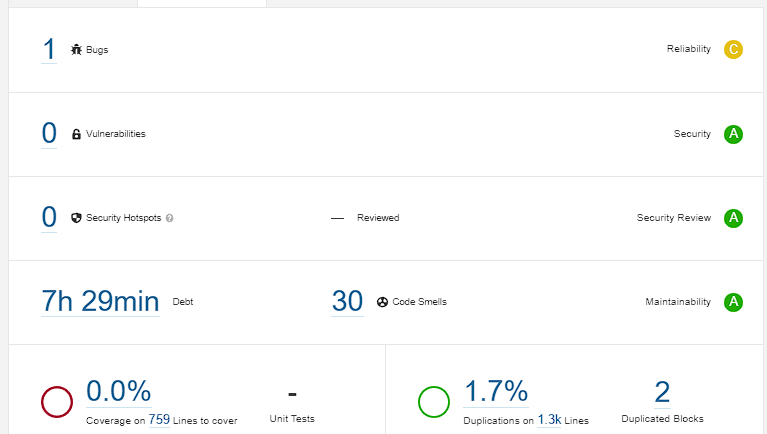
Se realizó la repartición de los errores encontrados por SonarQube. Se consideró el nivel de severidad y la dificultad de los errores para la repartición. Estos se muestran en la siguiente captura obtenida de SonarQube.



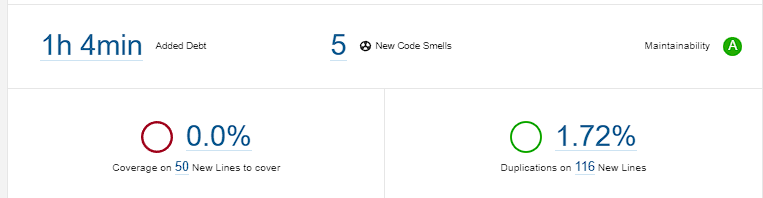
| **Integrante** | **Total errores asignados** |
| --- | --- |
| Suarez Hernandez, Kenny Joel | 20 |

## Resultado final

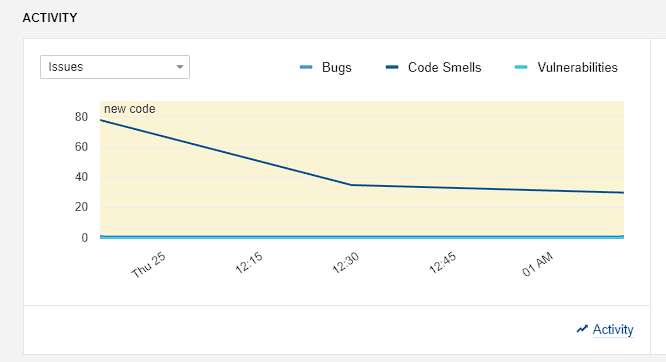
Después de haber realizado la atención de problemas, se realizó una segunda pasada usando la herramienta SonarQube. Los resultados obtenidos se presentan en la siguiente imagen:

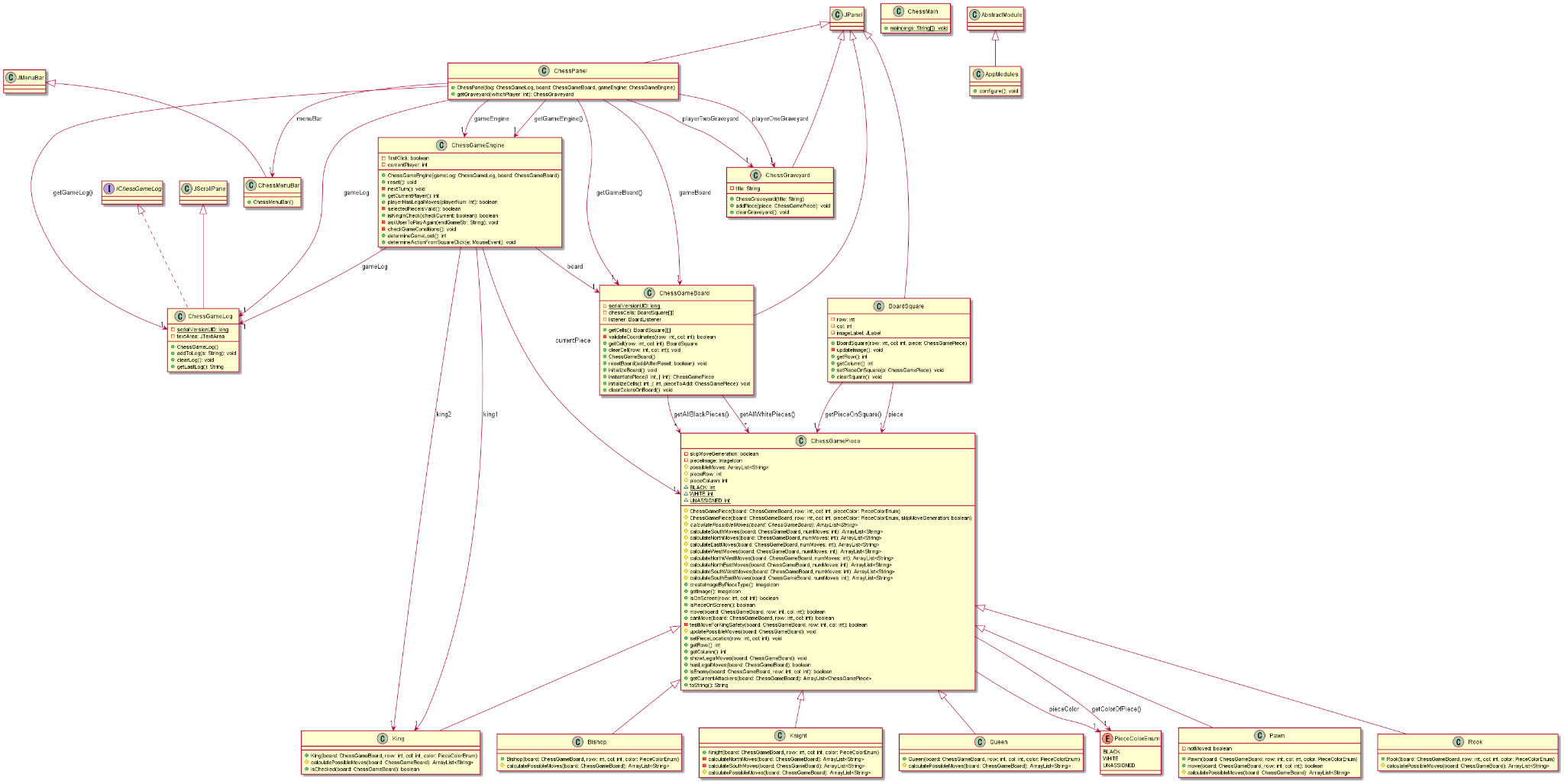


Podemos observar que ha sido solucionado un total de 52 errores de los 78 encontrados. Sin embargo, las líneas duplicadas se vieron afectadas durante la solución de los errores aumentando a un total de 1.72% y agregando 5 code smells

.

El diagrama de clases obtenido después de la refactorización posee 1 nueva clase llamada PieceColorEnum.java. Esta clase se creó para poder reducir la complejidad del código. Finalmente, se muestra el diagrama de clases resultante:





Enlace directo: <https://raw.githubusercontent.com/devvildaz/clps-holochess/rama-anderson/diagramaRefactor.svg>