Informe de proyecto: Refactorización y Modelado UML

Entornos De Desarrollo

Miembros del equipo

Marina, Migue y Pablo.

Proyecto original

El proyecto original era una aplicación de software con anomalías y deficiencias de diseño en su código, desarrollado antes como parte de un proyecto de trabajo. La refactorización de dicho proyecto ha sido el objetivo principal de este trabajo junto con la generación del Javadoc.

Organización del trabajo

Hemos realizado una reunión inicial en la que vimos en qué consistía el proyecto y cómo organizarnos. Aprovechamos esa reunión para hacer un code smell del anterior proyecto y ver las redundancias de código y cómo refactorizarlo. Una de las técnicas usadas para esta tarea ha sido la programación por pares, en este caso los tres participantes del equipo, mediante una llamada de Discord se ha compartido pantalla y hemos sugerido ideas y cambios en el momento. Otra herramienta usada ha sido el sistema de control de versiones GitHub, en el que hemos ido guardando todos los cambios del proyecto.

Al final de este documento se encuentra el link al repositorio.

Proceso de refactorización

La refactorización es un proceso de mejora de la calidad del código sin alterar la funcionalidad esencial de la aplicación del software. Limpia y organiza el código, haciéndolo más legible y manejable.

El equipo trabajó en conjunto para identificar las áreas problemáticas del código original y hacer las modificaciones necesarias para mejorar su calidad. Este proceso implicó la revisión exhaustiva de cada línea de código, la identificación de redundancias y la implementación de soluciones más optimizadas.

Diagramas UML

Uno de los hitos del proyecto fue la creación de diagramas UML para ilustrar la estructura y el funcionamiento del proyecto refactorizado. Para la realización de estos hemos usado el programa Umbrello.

Diagrama de clases

El diagrama de clases muestra cómo las clases del software están relacionadas entre sí y cómo interactúan. Ayuda a comprender la estructura y la arquitectura del software de forma visual. Marina creó un diagrama detallado que reflejaba las nuevas clases y la relación entre ellas después de la refactorización.

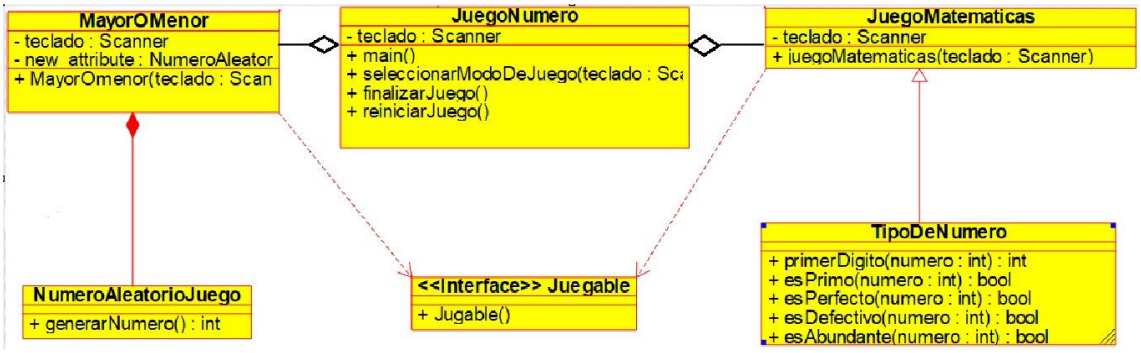


Diagrama de casos de uso

El diagrama de casos de uso ilustra las diferentes formas en que un usuario puede interactuar con el software. Esto ayuda a comprender el flujo y la funcionalidad del software desde la perspectiva del usuario. Marina también trabajó en el desarrollo del diagrama de casos de uso, mostrando todas las posibilidades de interacción con el software.

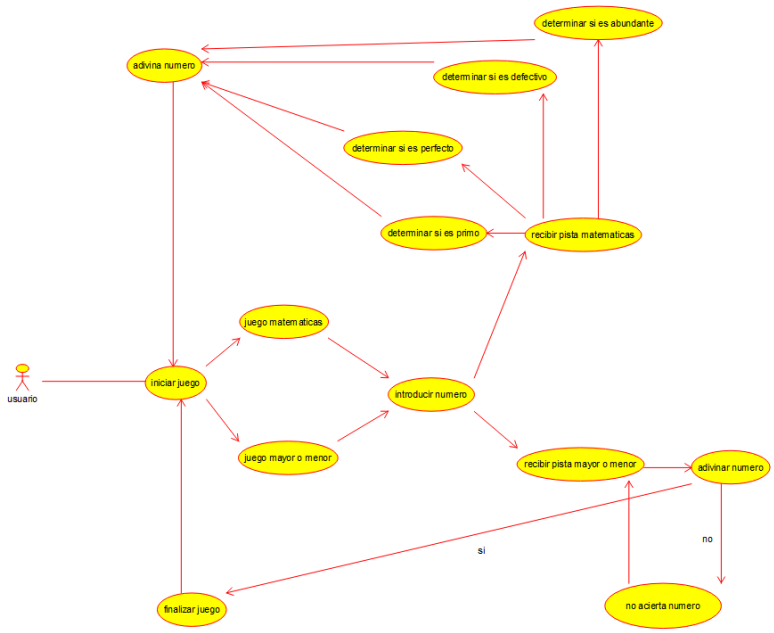
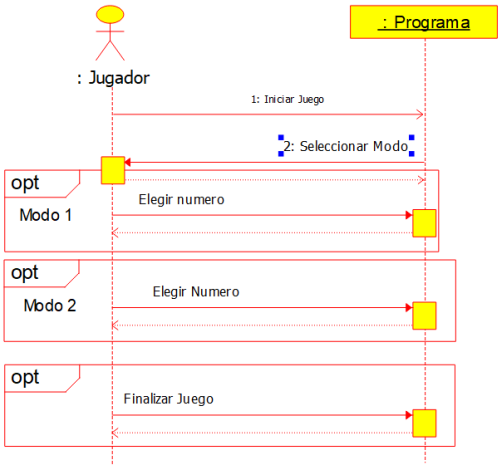


Diagrama de secuencia

El diagrama de secuencia muestra las interacciones entre las clases en un orden cronológico. Revela cómo fluye la lógica dentro de la aplicación de software.



Resultados

La refactorización resultante produjo un software eficiente y mejor estructurado, que era más fácil de leer y mantener. Además, incluimos la encapsulación de excepciones.

Los diagramas UML, a su vez, sirvieron como una valiosa documentación visual para la comprensión de la estructura y el funcionamiento de la aplicación.

Conclusión

Este proyecto ha sido una experiencia de aprendizaje para el equipo en términos de refactorización y diseño de software. Se mejoró significativamente la calidad del proyecto original y se obtuvo una comprensión más profunda de las técnicas UML.

Enlace GitHub

https://github.com/Pablofh97/Proyecto-Juego