EMMANUEL BUSTAMANTE VALBUENA ALEJANDRO BECERRA

Nombre del profesor: LEON DANIEL JARAMILLO RAMIREZ

Nombre de la asignatura: Técnicas y programación

4 de septiembre del 2022

Reto 8. Refactorización del código de programación

A lo que vamos con este informe es resumir todo el proceso de refactorización en los códigos del reto 7 y reto 6 respectivamente, con el objetivo de adquirir conocimiento acerca de las buenas prácticas, desarrollando las prácticas de código limpio y principios SOLID, mostrando cada uno de los cambios realizados con su respectivo análisis

Primer apartado, Código Limpio y buenas prácticas.

El término código limpio, se ha venido trabajando desde el inicio de la programación, puesto que anteriormente y por la "ignorancia" con respecto al tema, era muy usual cometer errores lógicos y de organización en el código, es así como a traves de tiempo, se ha venido desarrollando una serie de patrones o buenas prácticas, que se deben aplicar al código, para que este resulte mucho más fácil de entender y sea sostenible en el tiempo.

Ahora con respecto a estas prácticas vamos a comparar el código realizado en el reto (7) con respecto a las variables.

- En el nombrado de las variables se logra identificar unas buenas prácticas donde su mayoría cumple con los requisitos para ser catalogadas como buenas prácticas.

- En el nombrado de Clases se presenta un fallo puesto que las clases no son lo suficientemente autodescriptivas, se procede a cambiarse.
- Por el contrario en el nombrado de métodos se encontró que no estaban escritos correctamente haciendo uso de un verbo, por lo que se dispuso a cambiarse.
- En todo el código se evitó los casos de Code Smells.
- No se percibe ningún antipatrón
- Se agregaron par de comentarios en el código para ser más entendible

Comparativa con el reto (5)

- Se hacen ligeros cambios en algunas variables para dejarlas aún más claras. Por lo demás se cumple las buenas prácticas para el nombrado de variables.
- Las clases presentan buen nombramiento, ya que son lo suficientemente auto descriptivas ya que están haciendo el uso de un nombre propio e inmutable.
- Se encontró que algunos métodos no estaban escritos correctamente haciendo uso de un verbo, por lo que se dispuso a cambiarse.
- En todo el código se evitó los casos de Code Smells.
- Se presentó un antipatrón “Entrada chapuza” por lo que al interior del codigo se creo una clase exception, que es la encargada de evitar el ingreso de alguna entrada invalida