Reporte Refactoring

Los 5 Code Smells que deseo documentar son los siguientes:

Feature Envy (envidia de características).

Este mal olor de programación se refiere a una clase que utilizan métodos de otras clases más veces de lo que la propia clase lo hace, lo cual genera muchas llamadas a otra clase por el uso del método. (Shvets, 2014)

- Move Method

- Extract Method

Primitive Obsession (obsesión de los datos primitivos)

Se refiere a que las clases de un programa solo utilicen exclusivamente datos primitivos, y no se utilicen estructuras de datos o clases como tales. Por qué la programación OO, ayuda a que los atributos (clases) sean mas robustos y más seguros. (Shvets, 2014)

- Replace Data Value with Object

Data Clumps (grupo de datos)

Hace referencia a un conjunto de datos que funcionan solo si están completos, es decir si tengo 3 atributos que conforma un data clump, si yo tuviera un dato de los 3, eso no significa nada, pero si tengo los 3, tendrían un significado completo y entendible. Es un mal olor por que este puede contribuir a otros malos olores como long parameter list. Y además no es eficiente. (Shvets, 2014)

- Extract Class

- Introduce Parameter Object

- Preserve Whole Object

Long Class (clases largas)

Este mal olor quiere decir que una clase que tiene muchos métodos, y esto genera que la clase tenga muchas responsabilidades, lo cual no es recomendado según los principios SOLID. (Shvets, 2014)

- Extract Class

- Extract Subclass

- Extract Interface

Shotgun Surgery (cirugía por shotgun o múltiple)

Este mal olor se da cuando una clase está asociada a varias clases y se realiza un cambio en esta clase, esto genera que las otras clases cambien para acomodarse al cambio que ocurrieron en la clase. (Shvets, 2014)

- Move Method

- Move Field

- Inline Class

# Bibliografía

Shvets, A. (2014). *Refactoring Guru*. Obtenido de Refactoring Guru: https://refactoring.guru/