

## 4.A. Refactorización

### 1. Introducción.

¿Podemos mejorar la estructura del código y que sea de mayor calidad, sin que cambie su comportamiento? ¿Cómo hacerlo? ¿Qué patrones hay que seguir?.

La **refactorización** es una técnica, que consiste en realizar pequeñas transformaciones en el código de un programa, para mejorar la estructura sin que cambie el comportamiento ni funcionalidad del mismo. Su objetivo es mejorar la estructura, la legibilidad o la eficiencia del código.

Con la refactorización se mejora el diseño del software, hace que el software sea más fácil de entender, hace que el mantenimiento del software sea más sencillo, la refactorización ayuda a que el programa sea más rápido.

La idea de refactorización de código, se basa en el concepto matemático de factorización de polinomios. Así, resulta que  $(x + 1)(x - 1)$  se puede expresar como  $x^2 - 1$  sin que se altere su sentido.

Algunas pistas que nos pueden indicar la necesidad de refactorizar un programa son:

- Código duplicado.
- Métodos demasiado largos.
- Clases muy grandes o con demasiados métodos.
- Métodos más interesados en los datos de otra clase que en los de la propia.
- Grupos de datos que suelen aparecer juntos y parecen más una clase que datos sueltos.
- Clases con pocas llamadas o que se usan muy poco.
- Exceso de comentarios explicando el código.

Hay que resaltar que la refactorización no cambia el comportamiento observable del software. El software sigue cumpliendo la misma función que hacía antes. Ningún usuario, ya sea usuario final u otro programador, podrá determinar qué cosas han cambiado.