**Tarea 5: Unidad 3**

1. Si un código fuente ha sido verificado (funciona correctamente) y validado (hace lo especificado en los requisitos) habiendo superado todos las pruebas, ¿crees que es mejorable? ¿en qué sentido?

Si, siempre puede ser simplificado y los códigos pueden evolucionar metiendo cosas nuevas en el para una mayor optimización.

1. ¿Qué es ‘Code Smell’?.

Es un código con una serie de síntomas que vienen a indicar que tal vez no se están haciendo las cosas de una forma del todo correcta, lo que puede llevar a que haya algún problema a futuro y un problema de trasfondo

1. Un desarrollador puede intuir que su sistema contiene code smells cuando se encuentra con:
   1. Código duplicado
   2. Métodos que necesitan muchos parámetros
   3. Métodos muy largos
   4. Clases muy grandes

Explica brevemente cada uno de los problemas anteriores, e indica cómo podría solucionarse cada uno.

a) Nos lleva a que cada vez que haya requisitos nuevos haya que tocar ese código en más de un sitio.

b) Aparecen cuando un método es muy largo o cuando se intenta independizar a las clases. Pero al delegar menos entre clases, necesitamos utilizar demasiados parámetros para que el método tenga lo que precisa para actuar.

c) cualquiera puede notar que mientras más largo sea un procedimiento, más difícil va a ser entenderlo.

d) Una clase intenta tomar demasiadas responsabilidades, por lo que debería seguir el principio de single responsibility, o sea, cumplir solamente con una de las funcionalidades del software y estar bien encapsulada.

1. ¿Qué es la refactorización?

Es una técnica de la ingeniería de software para reestructurar un código fuente, alterando su estructura interna sin cambiar su comportamiento externo.

1. ¿Qué técnicas se utilizan a menudo a la hora de refactorizar?

Desarrollo rojo-verde, branching by abstraction, combinar métodos, mover propiedades entre clases, organización de los datos, simplificación de fórmulas condicionadas, simplificación de las llamadas a método.