Tarea 6 Refactorización

1. Motivos para refactorizar

Mejorar el diseño del software

Hacer que el código sea más fácil de entender

Hacer que sea más fácil encontrar fallos

Para permitir programar más rápidamente

1. ¿Cuando refactorizar?

Cuando se modifique un código

Cuando hay código sospechoso.

1. Busca y explica brevemente la metáfora de los dos sombreros

Se dice que el programador tiene 2 sombreros para refactorizar. Uno para modificar código y otro para añadir nuevas funcionalidades. Dependiendo de la situación escoge uno u otro.

1. ¿Qué es el código espagueti?

El código espagueti es un término “despectivo” que se utiliza para los

programas de computación que **tienen una estructura de control de**

**flujo compleja**y cuando vemos este código se parece mucho a un

plato de espagueti, como si fuera un montón de hilos enredados.

1. Ventajas y desventajas de la refactorización.

Ventajas:

Una mejor comprensión que facilita el mantenimiento y la ampliación del software

La reestructuración del código fuente puede realizarse sin cambiar la funcionalidad

La mejora en la legibilidad del código facilita que otros programadores lo comprendan.

Desventajas:

Una refactorización imprecisa podría generar nuevos bugs y errores en el código

Cuando la refactorización es realizada por equipos grandes, llegar a acuerdos podría suponer mas trabajo del esperado

1. ¿Cuando no es recomendable refactorizar?

Cuando el código original es tan malo que merece mas la pena reescribirlo desde el principio

Cuando se están a punto de cumplir los plazos

1. ¿Qué son los métodos o patrones de refactorización de Eclipse? Indica 3 ejemplos.

Son patrones de refactorización que nos permiten plantear casos y previsualizar las posibles soluciones que se nos ofrecen.

Rename

Move

Extract Constant

1. ¿Qué son las pruebas unitarias y funcionales?

Las pruebas unitarias son una forma de comprobar que un fragmento de código funciona correctamente. Es un procedimiento más de los que se llevan a cabo dentro de una metodología ágil de trabajo.

Las pruebas funcionales se centran en comprobar que los sistemas desarrollados funcionan acorde a las especificaciones funcionales y requisitos del cliente. Este servicio ayuda a su organización a detectar los posibles defectos derivados de errores en la fase de programación.