

REEMPLAZAR CONDICIONALES POR POLIMORFISMO

Problema: El tener un condicional que tenga varias acciones dependiendo del tipo de objeto.

Solución: Crear subclases que coincidan con las ramas del condicional, y en ellas crear un método compartido.

Este refactor sirve cuando en el código se están haciendo varias tareas dependiendo de:

- Clase del objeto o la interfaz que la implementa.
- Un valor de algún atributo.
- El resultado de algún método del objeto.

En el caso de usar condicionales, cada vez que se agregue alguna propiedad al objeto se terminaría buscando estos condicionales para tener que agregarles esta condición, en cambio, si se utiliza polimorfismo esto no pasaría.

Beneficios:

- Se adhiere al principio de **Tell don't ask**, en vez de preguntar al objeto acerca de su estado, y de ahí decidir que acción hacer, es mucho mejor simplemente decirle al objeto que es lo que tiene que hacer, y que el mismo decida el cómo hacer.
- Elimina código duplicado.
- Si se quiere agregar una nueva variante lo que se debería hacer es agregar una nueva subclase, sin tener que tocar el código existente.

Notar que está completamente relacionado con el principio de diseño **Open - Close**, el cual dice que una clase debería estar cerrada a la modificación, pero abierta para la extensión, es decir, que se debe poder agregar comportamiento sin tener que modificar el código ya existente, como por ejemplo, el tener que agregar una condición en el if por haber agregado una nueva variante.