

BAD SMELLS

Temporary field (atributo temporal):

Atributos que se usan solo en algunos métodos y no son esenciales para la clase.

Data class (clase de datos): Clases que solo contienen atributos y métodos de acceso, sin lógica significativa.

Comments (comentarios excesivos): Comentarios innecesarios o en exceso, indicando que el código no es lo suficientemente claro.

Speculative generality (generalidad especulativa):

Funcionalidades añadidas "por si acaso" que no son necesarias actualmente.

Parallel inheritance hierarchies

(jerarquías paralelas): Jerarquías de clases que crecen en paralelo y requieren cambios coordinados.

Primitive obsession (obsesión por tipos primitivos):

Uso excesivo de tipos primitivos para representar entidades más complejas.

Message chains (cadena de mensajes):

Llamadas en cadena a métodos a través de varios objetos, generando complejidad.

Dead code (código muerto):

Elementos que ya no se usan en el código.

Duplicated code (código duplicado):

Código repetido o que realiza la misma tarea en varias partes del sistema.

Data clumps (grupo de datos):

Conjuntos de datos que siempre aparecen juntos en múltiples métodos o clases.

Refused bequest (legado rechazado):

Subclases que heredan métodos o atributos innecesarios de una clase base.

Alternative Classes with Different Interfaces (clases alternativas con interfaces diferentes):

Clases que hacen lo mismo pero con interfaces diferentes, generando inconsistencias.

Divergent change (cambio divergente):

Una clase que cambia por diferentes razones, violando el principio de responsabilidad única.

Large class (clase grande): Clases con demasiados atributos o métodos, manejando demasiadas responsabilidades.

Feature envy (envidia de características): Métodos que acceden a los atributos de otra clase más que a los propios.

Middleman (intermediario):

Clases que delegan la mayor parte de sus responsabilidades a otras clases sin aportar valor propio.

Long parameter list (lista de parámetros larga):

Métodos o constructores que reciben demasiados parámetros.

Incomplete Library Class (clase de biblioteca incompleta):

Uso de bibliotecas que no ofrecen toda la funcionalidad necesaria.

Shotgun surgery (cambio en cadena):

Un pequeño cambio en una clase requiere cambios en varias otras clases.

Lazy class (clase perezosa): Clases que no tienen suficiente funcionalidad para justificar su existencia.

Switch statement: Uso excesivo de sentencias switch o if-else anidados.

Middleman (intermediario):

Clases que delegan la mayor parte de sus responsabilidades a otras clases sin aportar valor propio.

Long method (método largo):

Métodos con demasiadas líneas de código.

REFACTORIZACIONES

Encapsulate Collection:
En lugar de exponer una colección (List, Set, Map), se la encapsula y se proporcionan métodos controlados para modificarla.

Extract Method: Se extrae un fragmento de código complejo o largo a un método separado con un nombre descriptivo.

Introduce Parameter Object:
Se agrupan varios parámetros relacionados en un solo objeto.

Pull Up Method/Field: Se mueve un método o atributo común a la clase padre.

Decompose Conditional: Se descompone una condición compleja en métodos separados para mayor claridad.

Encapsulate Field:
Los atributos de una clase se hacen privados y se accede a ellos mediante getters y setters.

Inline Method:
Cuando un método es pequeño y simple, se reemplaza la llamada por su propio código.

Move Method/Move Field: Se mueve un método o atributo a la clase más adecuada.

Push Down Method/Field: Se baja un método o atributo de la clase padre a las subclasses cuando solo es relevante para algunas de ella.

Remove Middleman:
Se elimina una clase intermedia que delega la mayor parte de su trabajo a otra sin aportar valor.

Extract Class: Se extrae parte de la funcionalidad de una clase con demasiadas responsabilidades a una nueva clase.

Inline Temp: Se elimina una variable temporal que se usa solo una vez, reemplazándola con la expresión original.

Rename Method/Field: Se cambia el nombre de un método o atributo para reflejar mejor su propósito.

Introduce Explaining Variable: Se extrae una expresión compleja a una variable con un nombre descriptivo.

Extract Superclass: Se mueven atributos o métodos comunes a una superclase para eliminar duplicación y mejorar la reutilización.

Introduce Null Object:
Se crea un objeto que represente un "nulo" con comportamiento definido para evitar comprobaciones constantes de null.

Replace Primitive with Object: Se reemplazan tipos primitivos con clases que encapsulan datos y comportamiento.

Replace Conditional with Polymorphism: Se usa polimorfismo para reemplazar condicionales (if, switch), permitiendo que las subclasses manejen comportamientos específicos.