

Resumen de Bad Smells

Bad Smell	Significado	Refactoring Sugerido
Large Class (Clase Larga)	Clase con muchas responsabilidades.	Extract Class, Extract Subclass
Lazy Class (Clase Vaga)	Clase con poca lógica.	Inline Class
Data Class (Clase de Datos)	Solo tiene atributos públicos o getters/setters sin ninguna lógica y expone directamente	Encapsulate Field, Add Behavior
Duplicated Code (Codigo duplicado)	Código repetido en varios lugares.	Extract Method, Pull Up Method
Feature Envy (Envidia de atributos)	Método usa datos de otra clase.	Move Method
Long Method (Metodo Largo)	Método largo y difícil de entender.	Extract Method
Temporary Field (Campo temporal)	Campo usado solo en ciertos casos.	Introduce Null Object
Temporary Variable (Variable temporal)	Se refiere cuando una variable local se usa innecesariamente para almacenar un valor que podría ser directamente calculado o retornado en una expresion mas simple	Replace Temp with Query
Divergent Change	Clase cambia por varios motivos.	Extract Class
Switch Statements (Switchs o exceso de if)	Uso excesivo de condicionales.	Replace Conditional with Polymorphism
Loop Abusers (Abuso de loops)	ocurre cuando hay bucles explicitos for y while para realizar operacioens que podrian expresarse de manera mas declarativa con Streams	Replace Loop with Pipeline
Bad Naming (Mal nombre)	Nombre no refleja comportamiento.	Rename Method/Variable
Primitive Obsesion	Uso excesivo de primitivos.	Replace Primitive with Object
Speculative Generality	Código agregado por si acaso.	Remove Dead Code
Dead Code	Código que nunca se usa.	Remove Dead Code
Long Parameter List	Método con muchos parámetros.	Introduce Parameter Object
Shotgun Surgery	Un cambio afecta muchas clases.	Move Method, Move Field
Parallel Inheritance	Jerarquías paralelas innecesarias.	Collapse Hierarchies
Inappropriate Intimacy	Clase conoce demasiado a otra.	Move Method, Move Field
Message Chains	Llamadas encadenadas a objetos.	Hide Delegate
Middle Man	Clase solo delega.	Remove Middle Man
Data Clumps	Variables que siempre aparecen juntas	Introduce Parameter Object
Refused Bequest	Subclase hereda comportamiento no usado	Replace Inheritance with Delegation

Resumen de Refactoring

Refactoring Explicación	Explicación
Encapsulate Field (Encapsular Campo)	Hacer campos privados, con getters si es necesario
Encapsulate Class (Encapsular Clase)	Encapsular variables relacionadas.
Extract Class (Extraer Clase)	Mover responsabilidades a una nueva clase
Extract Method (Extraer Metodo)	Dividir método largo en métodos pequeños.
Move Method (Mover Metodo)	Mover método a clase donde más se usa.
Move Field (Mover Campo)	Mover atributo a clase correcta.
Pull Up Method (Subir metodo a superclase)	Mover método común a superclase.
Pull Up Field (Subir campo a superclase)	Mover campo por ejemplo uando se detectan variables de instancia iguales entre clases (similares) como EmpleadoPlanta y EmpleadoContratado
Rename Method/Variable (Renombrar Metodo/Variable)	Renombrar para mayor claridad.
Remove Dead Code	Eliminar código no usado.
Introduce Null Object	Usar objeto nulo en vez de campo temporal.
Replace Loop with Pipeline	Convertir un for / while en una expresion con Stream
Extract Subclass	Crear subclase para responsabilidades específicas.
Inline Class	Eliminar clase innecesaria y mover su contenido.
Add Behavior	Agregar lógica a clases de solo datos.
Replace Inheritance with Delegation	Reemplazar herencia por composición.
Introduce Parameter Object	Agrupar parámetros relacionados.
Preserve Whole Object	Pasar un objeto en lugar de múltiples parámetros.
Hide Delegate	Ocultar delegaciones innecesarias.
Remove Middle Man	Eliminar clases que solo delegan.
Collapse Hierarchies	Simplificar jerarquías innecesarias.
Replace Primitive with Object	Sustituir primitivos por objetos.
Introduce Null Object	Usar objeto nulo en vez de campo temporal.