

Code Smell	Refactoring
<i>Clases alternativas con diferentes interfaces</i>	<ul style="list-style-type: none"> <li>- Renombrar de metodo</li> <li>- Mover metodo</li> <li>- Extraer superclase</li> </ul>
Comentarios	<ul style="list-style-type: none"> <li>- Renombrar metodo</li> <li>- Extraer metodo</li> <li>- Introducir asercion</li> </ul>
Clase de datos	<ul style="list-style-type: none"> <li>- Mover metodo</li> <li>- Encapsular campo</li> <li>- Encapsular coleccion</li> <li>- Eliminar metodo de configuracion</li> <li>- Extraer metodo</li> <li>- Ocultar metodo</li> </ul>
Grupos de datos	<ul style="list-style-type: none"> <li>- Extraer clase</li> <li>- Conservar todo el objeto</li> <li>- Recibir objeto por parametro</li> </ul>
Codigo duplicado:	<ul style="list-style-type: none"> <li>- Extraer metodo</li> <li>- Subir atributo a clase padre</li> <li>- Subir metodo a clase padre</li> <li>- Extraer clase</li> </ul>
Envidia de atributos	<ul style="list-style-type: none"> <li>- Extraer metodo</li> <li>- Mover metodo</li> <li>- Mover campo</li> </ul>
Clase perezosa: Clase que no justifica su existencia porque tiene muy poca funcionalidad	<ul style="list-style-type: none"> <li>- Eliminar jerarquia</li> <li>- Incorporar clase</li> </ul>
Intimidad inapropiada: Ocurre cuando una clase accese demasiado a los detalles internos de otra clase. Violando el encapsulamiento.	<ul style="list-style-type: none"> <li>- Mover metodo</li> <li>- Mover campo</li> <li>- Extraer clase</li> </ul>
Clase de biblioteca incompleta: Ocurre cuando una clase de una biblioteca o framework no tiene toda la funcionalidad que necesita.	<ul style="list-style-type: none"> <li>- Extraer metodo ajeno</li> <li>-</li> </ul>
Clase larga	<ul style="list-style-type: none"> <li>- Extraer clase</li> <li>- Extraer subclase</li> <li>- Extraer interfaz</li> <li>-</li> </ul>
Metodo largo	<ul style="list-style-type: none"> <li>- Extraer metodo</li> <li>- Recibir un objeto por parametro</li> </ul>

Lista larga de parametros	<ul style="list-style-type: none"> <li>- Reemplazar parametro con metodo</li> <li>- Recibir objeto por parametro</li> </ul>
Switch Statement: Podemos detectar este code smell, por que vemos distintos case anidados y falta de poliformismo.	<ul style="list-style-type: none"> <li>- Extraer metodo</li> <li>- Mover metodo</li> <li>- Reemplazar switch con subclases</li> <li>- Reemplazar condicional por polimorfismo</li> </ul>
Nombres pocos autoexplicativos	<ul style="list-style-type: none"> <li>- Renombrar metodo</li> <li>- Renombrar variable</li> </ul>
Codigo muerto	<ul style="list-style-type: none"> <li>- Eliminar codigo muerto</li> </ul>
Clase con demasiadas responsabilidades	<ul style="list-style-type: none"> <li>- Extraer clase</li> <li>- Extraer subclase</li> <li>- Aplicar principio de responsabilidad unica</li> </ul>