Trabajo Autónomo: Refactoring

Tabla de contenido

[CODE SMELLS 2](#_Toc29651527)

[*Feature envy* 2](#_Toc29651528)

[ Nombre español: 2](#_Toc29651529)

[ Descripcion: 2](#_Toc29651530)

[ Lista de técnicas de refactorización: 2](#_Toc29651531)

[*Primitive Obsession* 2](#_Toc29651532)

[ Nombre español: 2](#_Toc29651533)

[ Descripcion: 2](#_Toc29651534)

[ Lista de técnicas de refactorización: 2](#_Toc29651535)

[*Refused Bequest* 2](#_Toc29651536)

[ Nombre español: 2](#_Toc29651537)

[ Descripcion: 2](#_Toc29651538)

[ Lista de técnicas de refactorización: 2](#_Toc29651539)

[*Switch statements* 2](#_Toc29651540)

[ Nombre español: 2](#_Toc29651541)

[ Descripcion: 2](#_Toc29651542)

[ Lista de técnicas de refactorización: 2](#_Toc29651543)

[*Shotgun surgery* 2](#_Toc29651544)

[ Nombre español: 2](#_Toc29651545)

[ Descripcion: 2](#_Toc29651546)

[ Lista de técnicas de refactorización: 3](#_Toc29651547)

[REFERENCIAS 3](#_Toc29651548)

# CODE SMELLS

## *Feature envy*

### Nombre español:

Envidia

### Descripcion:

Se da cuando un método accede a otros métodos o atributos de un objeto. Es un code smell porque no se debe permitir que una clase acceda a los atributos de otra, sin un getter o setter, así mismo con métodos.

### Lista de técnicas de refactorización:

## *Primitive Obsession*

### Nombre español:

Obsesión de primitivos

### Descripcion:

Uso de primitivos en lugar de clases nuevas y específicas para atributos como teléfonos, moneda, etc. que manejen su propio comportamiento.

### Lista de técnicas de refactorización:

## *Refused Bequest*

### Nombre español:

Legado Rechazado

### Descripcion:

Cuando una clase hija hereda un comportamiento que no le pertenece o no usa.

### Lista de técnicas de refactorización:

## *Switch statements*

### Nombre español:

Condicionales

### Descripcion:

Cuando tienes unas sentencias switch complicadas o de secuencias de if difíciles de entender.

### Lista de técnicas de refactorización:

## *Shotgun surgery*

### Nombre español:

Cirugia de escopeta

### Descripcion:

Se da cuando un solo cambio se hace en varios lugares que puede provocar errores en el codigo.

### Lista de técnicas de refactorización:

# REFERENCIAS

* Descripción corta: donde indique porque eso es un problema en el código.
* Lista de técnicas de refactorización: con las consideraciones necesarias para aplicar cada técnica y corregir el mal olor.
* Enlace a GitHub: Ejemplo (dentro de su propio repositorio) para demostrar el mal olor y el código corregido aplicando una de las técnicas de refactorización. No puede ser el mismo que el dado en las páginas de referencia.