Escuela Superior Politécnica del Litoral

Diseño de Software

Nombre: Luis Enrique Anchundia Solórzano

Trabajo autónomo: Refactoring

Olores del código

Feature envy – envidia de características

Descripción: Este mal olor de programación se da cuando los métodos de una clase están mas interesados en otra clase que en si misma. Es un problema porque se accede a los datos de otra clase y no a los datos de su clase.

Técnicas de refactorización:

- Mover los métodos a la clase en la que están interesados.
- Remover los métodos por completo si no son relevantes y crear nuevos métodos en la clase utilizada.
- Aplicar el método de extracción para reestructurar el método en varias partes para luego ubicar dichas partes en las clases correctas.

Primitive Obsession – Obsesion primitiva

Descripción: Existe obsesión primitiva cuando se abusa en el uso de datos primitivos (String, int, constantes) como por ejemplo el uso de un string para almacenar una dirección url de un sitio web, en este caso una dirección url posee ciertas características y no es correcto almacenarla solo en una cadena de caracteres. La lógica de una clase es controlada por un dato primitivo lo cual podría causar problemas de seguridad.

Técnicas de refactorización:

- Crear clases pequeñas que manejen datos con pocos atributos como una url o un número de teléfono.
- Utilizar objetos en lugar de datos primitivos

Data clumps – grupos de datos

Descripción: Son grupos de datos que se encuentran juntos y se repiten en algunas partes del código, por lo general suelen ser datos primitivos.

Técnicas de refactorización:

- Convertir estos grupos de datos en objetos.
- Crear una clase e insertar estos datos y luego pasar instancias del mismo.

Refused bequest – Legado rechazado

Descripción: Este malo olor se da cuando una subclase hereda métodos y propiedades de una superclase, pero solo utiliza unos pocos métodos de todos los métodos que le ha proporcionado la superclase, esto se da cuando las clases no tienen relación alguna y son diferentes entre sí.

Técnicas de refactorización:

- Eliminar la herencia cuando esta no tenga sentido.
- Si la herencia tiene sentido, separar métodos en subclases y luego modificar la superclase para hacerla coincidir con ambas subclases.

Shotgun surgery – Cirugía de escopeta

Descripción: Este mal olor de programación se puede evidenciar cuando es necesario realizar el mismo cambio en muchas clases.

Técnicas de refactorización:

- Ubicar los comportamientos de clases existentes en una sola clase.
- Crear una clase nueva que reúna todos los comportamientos.

Bibliografía

Data clumps, primitive obsession and hidden tuples. (s. f.). Recuperado 11 de enero de 2020, de https://codesai.com/2018/08/hidden-tuples

Obsesión primitiva: Olor a código que más duele a las personas. (s. f.). Recuperado 10 de enero de 2020, de https://medium.com/@arpitjain.iec/primitive-obsession-code-smell-that-hurt-people-the-most-5cbdd70496e9

Principios sólidos—¿Qué es un legado rechazado? - Desbordamiento de pila. (s. f.). Recuperado 11 de enero de 2020, de https://stackoverflow.com/questions/28271150/what-is-a-refused-bequest

Refused Bequest—A Code Smell. (s. f.). Recuperado 11 de enero de 2020, de https://www.c-sharpcorner.com/article/refused-bequest-a-code-smell/

The Shotgun Surgery Code Smell—DZone Java. (s. f.). Recuperado 11 de enero de 2020, de https://dzone.com/articles/code-smell-shot-surgery

Un repaso por los code smells más comunes—Intive Argentina Blog. (s. f.). Recuperado 10 de enero de 2020, de https://blog.intive-fdv.com.ar/repaso-los-code-smells-mas-comunes/