

A stylized illustration of a desk setup. In the center is an open laptop with a teal screen and a dark keyboard. To the left of the laptop is a stack of three books in teal, orange, and teal. Below the books is a potted plant with long, pointed leaves in teal and orange, sitting in an orange pot. To the right of the laptop is a teal pen holder with a pink base, containing three pens in orange, teal, and orange. Above the laptop is a teal folder or notebook with a white border and a pattern of small white crosses. In the bottom right corner, there is a stylized illustration of a computer monitor with a pink screen and teal accents.

SISTEMA DE TUTORIAS

Análisis de detección de Code Smells

Desarrollado con Django, SQLite

Integrantes

- **Casilla Percca Dante Vladimir**
- **Choque Bueno Fiorella Silvia**
- **Luycho Ancaifuro Mariela**
- **Huaman Hermoza Antony Isaac**
- **Tacusi Larota Jhon Edwin**
- **Ttito Quilca Cesar Rodrigo**
- **Ttito Saya Alexander**
- **Vargas Arqqe Jeremyk Rufino**

Grupo 4

Grupo Nro 4

Integrantes

- Casilla Percca Dante Vladimir
- Choque Bueno Fiorella Silvia
- Luycho Ancaifuro Mariela
- Huaman Hermoza Antony Isaac
- Tacusi Larota Jhon Edwin
- Ttito Quilca Cesar Rodrigo
- Ttito Saya Alexander
- Vargas Arqqe Jeremyk Rufin

Code Smells

Tiene el principal objetivo de indentificar problemas estructurales en un sistema que surgen por las malas prácticas de desarrollo y pueden generar problemas en un futuro.

Con esto se busca:

- Código que se fácil de comprender.
- Código que no sea duplicado
- Código que sea fácil de mantener
- Código que pueda ser utilizado o probado por otros programadores

Principales Code Smells

Tipo de Code Smell	Descripción
God Class	Son clases que tienden a centralizar la inteligencia del sistema. Una God Class realiza demasiado trabajo por cuenta propia, delega sólo detalles menores a otras clases. Impacta en la reusabilidad y comprensibilidad del sistema
Feature Envy	Son métodos que están más interesados en los datos de otras clases que en los propios. Esto podría ser una señal de que el método está fuera de lugar y debe ser movido a otra clase.
Data Class	Son clases que almacenan datos, no tienen funcionalidad compleja, o directamente su funcionalidad es nula, y otras clases dependen en gran medida de sus datos. Es un síntoma de una mala encapsulación de datos y funcionalidad.
Brain Method	Es un método difícil de mantener o entender. Tienden a centralizar la funcionalidad de una clase, de la misma manera que una God Class centraliza toda la funcionalidad de todo un sistema.
Brain Class	Es una Clase compleja que tiende a contener una cantidad excesiva de responsabilidades, por lo general contiene varios métodos afectados por el code smell Brain Method. Se diferencia de God Class por no acceder a datos de otras clases de manera abusiva.
Significant Duplication	Hace referencia a código duplicado que resulte significativo. Esto quiere decir que sea suficientemente grande.
Intensive Coupling	Son métodos con demasiadas dependencias que se encuentran dispersas en una o pocas clases.
Dispersed Coupling	Son métodos con demasiadas dependencias que se encuentran dispersas en muchas clases. Los métodos invocados suelen ser muy simples.
Shotgun Surgery	Se trata de la situación inversa al Dispersed Coupling. Hace referencia a cuando el cambio en un método genera muchos pequeños cambios en otros métodos y clases.
Refused Bequest	Hace referencia a cuando, en una jerarquía, un padre define métodos protegidos que no son utilizados por sus hijos. Esto es un indicio de que algo es incorrecto en la clasificación de relación entre clases.
Tradition Breaker	Hace referencia a una clase que no especializa la clase de la cual hereda,

Analysis

Significant Duplication

El code smell, hace referencia a código duplicado que resulte significativo. Esto quiere decir que sea suficientemente grande

Descripción y detección

1

Detectamos código repetido en los CRUDs tanto del back-End como del Front-End

2

Detectamos que por la similitud en algunas clases, hace que se copie gran parte los métodos de una clase a la otra, esto genera más código repetido e inútil

Analisis

God Class

El code smell God Class se refiere a clases que tienden a centralizar la inteligencia del sistema.

Descripción y detección

1

God Class fue detectado en una parte de nuestro CRUD.

2

Un solo archivo contenía todos los módulos y métodos. No hay una buena distribución de funciones.

3

Lo correcto es separarlos en scripts según a las funciones que cada uno tiene.

Analysis

Intensive Coupling

El code smell Intensive Coupling, se refiere a cuando un método que llama a varios métodos que se implementan en una o pocas clases.

Descripción y detección

1

Los métodos para la implementación de registros de tutoría generarían este code smell, ya que generarán una dispersión en las operaciones de esta clase

2

Los métodos para las operaciones de cambio de tutor también generarían una dispersión en las operaciones de la clase, esto por validaciones requeridas tanto en el estudiante como en los tutores(anterior y nuevo)

Resumen:

Code Smell	Resumen
God Class	Si fue encontrado
Feature Envy	No fue encontrado
Data Class	No fue encontrado
Brain Method	No fue encontrado
Brain Class	No fue encontrado
Significant Duplication	Si fue encontrado
Intensive Coupling	Si fue encontrado
Dispersed Coupling	Si fue encontrado
Shotgun Surgery	Si fue encontrado
Refused Bequest	Si fue encontrado
Tradition Breaker	Si fue encontrado

Bibliografía

<https://www.ridaa.unicen.edu.ar/xmlui/bitstream/handle/123456789/1724/Trabajo%20Final%20Malavolta.pdf?sequence=1&isAllowed=y>