**ESCUELA SUPERIOR POLITECNICA DEL LITORAL**

**FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION**

**Materia:**

DISEÑO DE SOFTWARE

**Tema:**

CODE SMELLS Y TECNICAS DE REFACTORIZACION

**Integrantes:**

* CORDOVA ANCHUNDIA LEA ALFONSINA
* LOPEZ VACA BRIGGITTE LIZBETH
* SEGURA MERCHAN RONNY HUGO

**Paralelo:**

3

**1S-2020**

**Contenido**

[**CODE SMELLS** 4](#_Toc48231382)

[**1.** **Nombre CS1** 4](#_Toc48231383)

[1.1. Descripción 4](#_Toc48231384)

[1.2. Capturas del Code Smell 4](#_Toc48231385)

[1.3. Consecuencias 4](#_Toc48231386)

[1.4. Técnica(s) de refactorización 4](#_Toc48231387)

[1.5. Capturas del código refactorizado 4](#_Toc48231388)

[**2.** **Nombre CS2** 4](#_Toc48231389)

[2.1. Descripción 4](#_Toc48231390)

[2.2. Capturas del Code Smell 4](#_Toc48231391)

[2.3. Consecuencias 4](#_Toc48231392)

[2.4. Técnica(s) de refactorización 4](#_Toc48231393)

[2.5. Capturas del código refactorizado 4](#_Toc48231394)

[**3.** **Nombre CS3** 4](#_Toc48231395)

[3.1. Descripción 4](#_Toc48231396)

[3.2. Capturas del Code Smell 4](#_Toc48231397)

[3.3. Consecuencias 4](#_Toc48231398)

[3.4. Técnica(s) de refactorización 4](#_Toc48231399)

[3.5. Capturas del código refactorizado 4](#_Toc48231400)

[**4.** **Nombre CS4** 4](#_Toc48231401)

[4.1. Descripción 4](#_Toc48231402)

[4.2. Capturas del Code Smell 4](#_Toc48231403)

[4.3. Consecuencias 4](#_Toc48231404)

[4.4. Técnica(s) de refactorización 4](#_Toc48231405)

[4.5. Capturas del código refactorizado 4](#_Toc48231406)

[**5.** **Nombre CS5** 4](#_Toc48231407)

[5.1. Descripción 4](#_Toc48231408)

[5.2. Capturas del Code Smell 4](#_Toc48231409)

[5.3. Consecuencias 4](#_Toc48231410)

[5.4. Técnica(s) de refactorización 4](#_Toc48231411)

[5.5. Capturas del código refactorizado 4](#_Toc48231412)

[**6.** **Nombre CS6** 4](#_Toc48231413)

[6.1. Descripción 4](#_Toc48231414)

[6.2. Capturas del Code Smell 4](#_Toc48231415)

[6.3. Consecuencias 4](#_Toc48231416)

[6.4. Técnica(s) de refactorización 4](#_Toc48231417)

[6.5. Capturas del código refactorizado 4](#_Toc48231418)

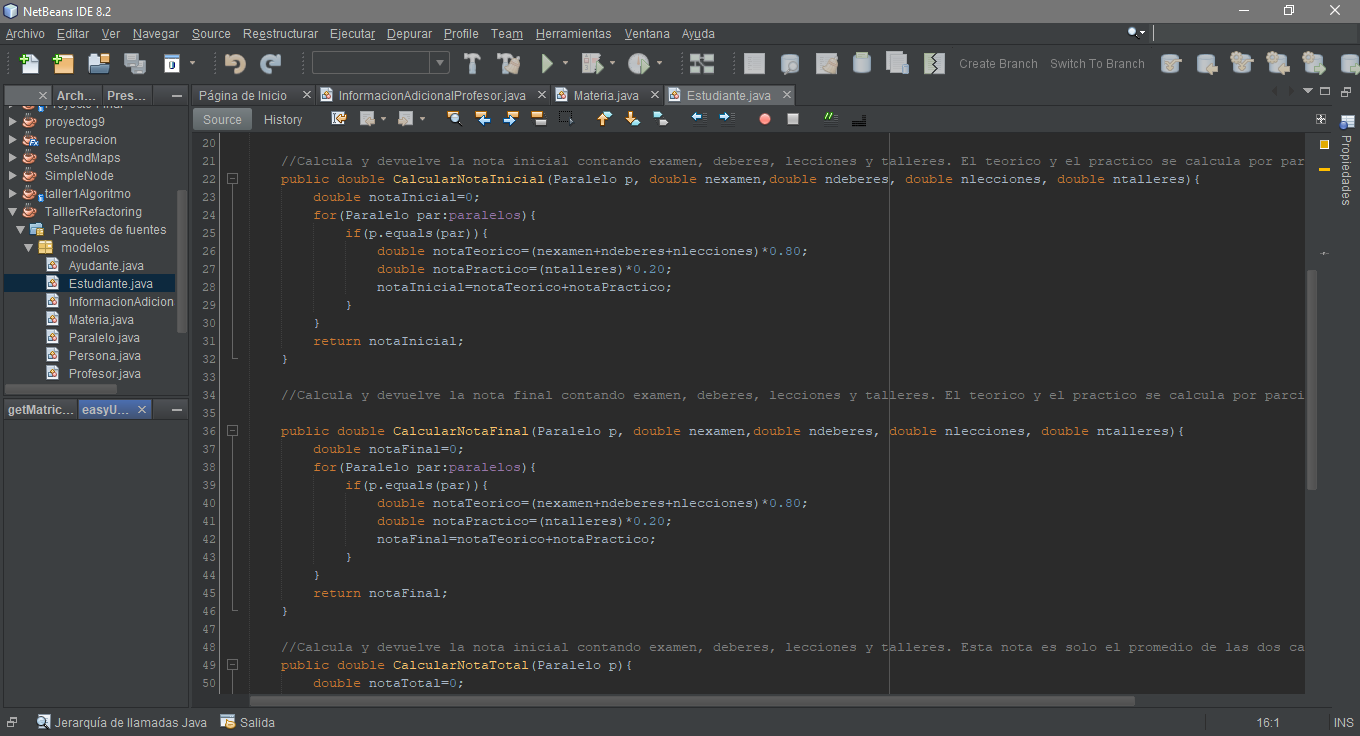
**CODE SMELLS**

1. **Nombre CS1** 
   1. Descripción
   2. Capturas del Code Smell
   3. Consecuencias
   4. Técnica(s) de refactorización
   5. Capturas del código refactorizado
2. **Nombre CS2**
   1. Descripción
   2. Capturas del Code Smell
   3. Consecuencias
   4. Técnica(s) de refactorización
   5. Capturas del código refactorizado
3. **Nombre CS3**
   1. Descripción
   2. Capturas del Code Smell
   3. Consecuencias
   4. Técnica(s) de refactorización
   5. Capturas del código refactorizado
4. **Nombre CS4**
   1. Descripción
   2. Capturas del Code Smell
   3. Consecuencias
   4. Técnica(s) de refactorización
   5. Capturas del código refactorizado
5. **Nombre Long Parameter List**
   1. Descripción

Este olor implica que los métodos contengan más de cuatro parámetros, lo cual se hace que esta lista se vuelva difícil de usar o contradictoria a medida que se añadan.

Lo que se pretende con su solución es pasar lo necesario como parámetro en un método o utilizar los datos del objeto que contenga lo necesario.

* 1. Capturas del Code Smell



* 1. Consecuencias

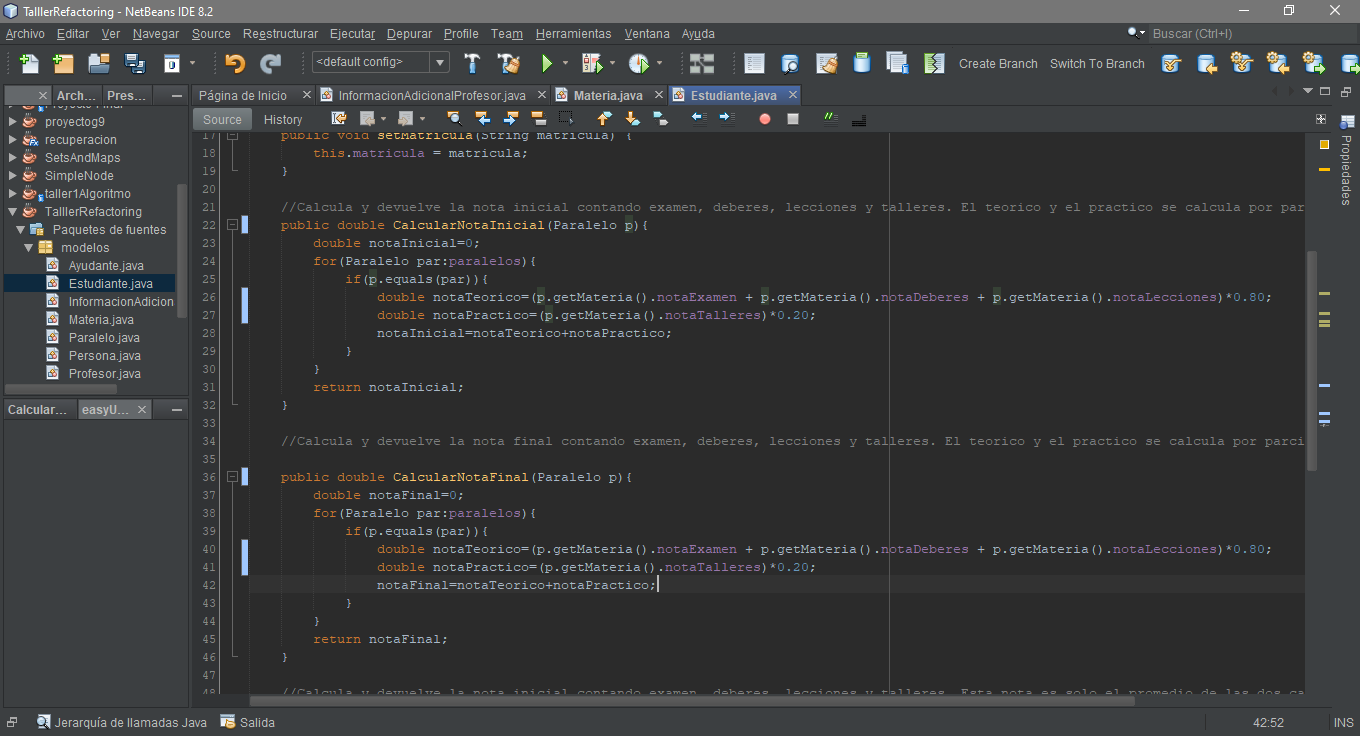
Este olor hace que el código no sea tan legible y su solución con refactorización implicaría que se genere código duplicado previamente invertido.

* 1. Técnica(s) de refactorización

Para la refactorización se aplico **Perserver Whole Object** en lo cual consiste remplazar una la lista de parámetros que recibe por solo el objeto que maneje los datos necesarios.

También se aplico Introduce **Parameter Object** donde los datos similares que son enviados como parámetros son remplazados por un objeto. Al aplicar esta refactorización se movieron los datos a otra clase por ende puede generar a una clase de datos, también como los métodos son dependientes de otra clase se los puede mover, para ello se usa **Move Method**.

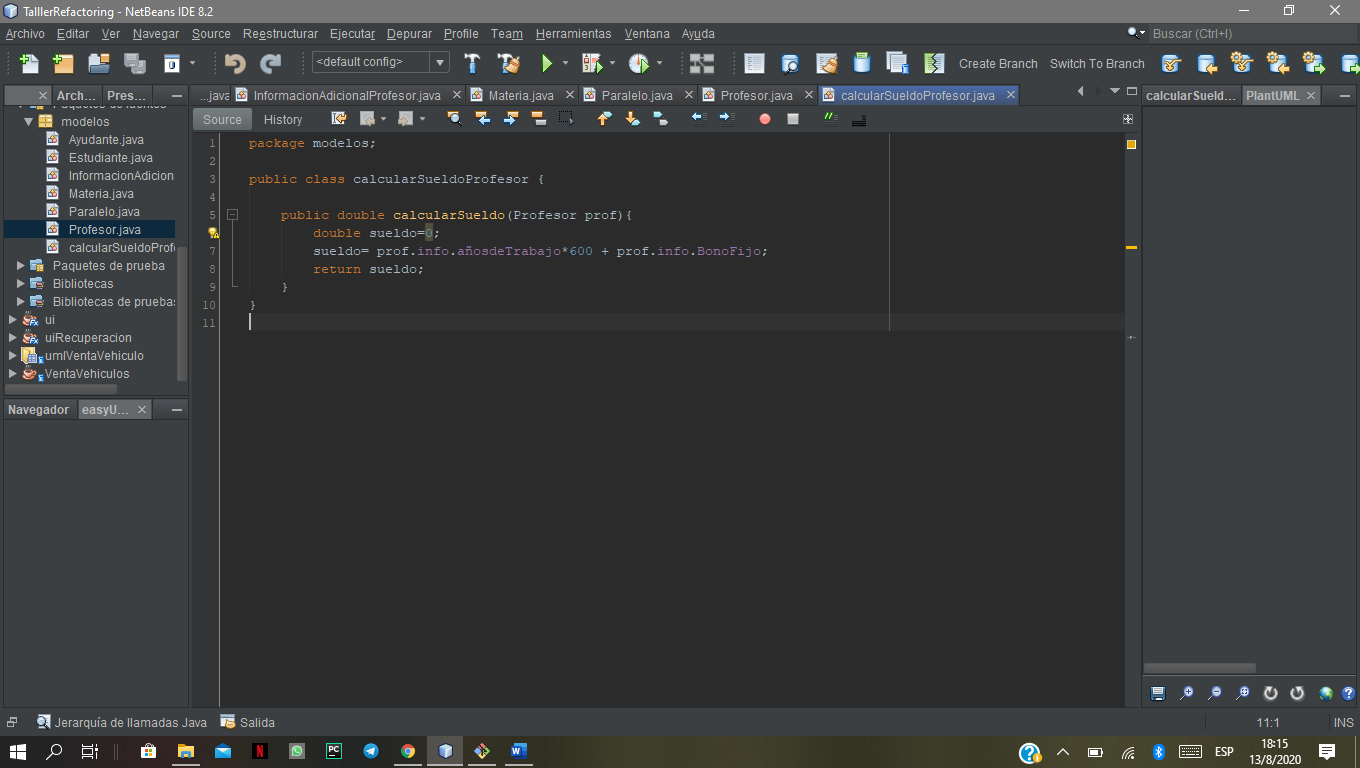
* 1. Capturas del código refactorizado



1. **Nombre Speculative Generality**
   1. Descripción

Este olor trata cuando existen clases con características que se anticipan y que nunca se implementan, por ende, el código se hace difícil de entender. Este olor hace que la clase CalcularSueldoProfesor no se ha utilizado.

* 1. Capturas del Code Smell



* 1. Consecuencias

Si no se refactoriza el olor hará que el código sea más robusto y por ende su soporte sea complicado.

* 1. Técnica(s) de refactorización

Para la refactorización se aplicó **Inline Class**, para delegar la función de esta clase en otra, para que esta clase pueda ser eliminada y por ende tener un código más ligero.

**Inline Class** implica que una clase dentro del código no cumpla ninguna función por ende se deben mover todas las funciones a otra clase.

* 1. Capturas del código refactorizado

