**DISEÑO DE SOFTWARE**

**TALLER 8: Refactoring**

**Integrantes:**

**Stephany Carolina Cabeza Tufiño**

**Eduart Sebastian Macias Sanchez**

**Luis Gabriel Cañarte Lucio**

Contenido

[Feature envy: 3](#_Toc111046277)

[Temporary field: 4](#_Toc111046278)

[Large class: 6](#_Toc111046279)

[Lazy class: 7](#_Toc111046280)

[Data clumps: 8](#_Toc111046281)

# Feature envy:

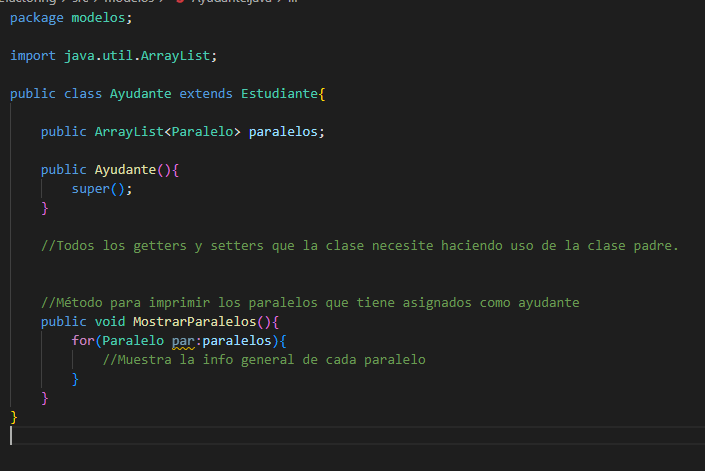
En la clase Ayudante se están usando únicamente los métodos correspondientes a la clase Estudiante, esto incurre en el code smell de Feature envy. Para solucionar este error se aplica la técnica de extraer subclase, con esto se eliminará el atributo estudiante y la clase Ayudante pasará a ser una clase hija de Estudiante.

Antes:

Texto

Descripción generada automáticamente

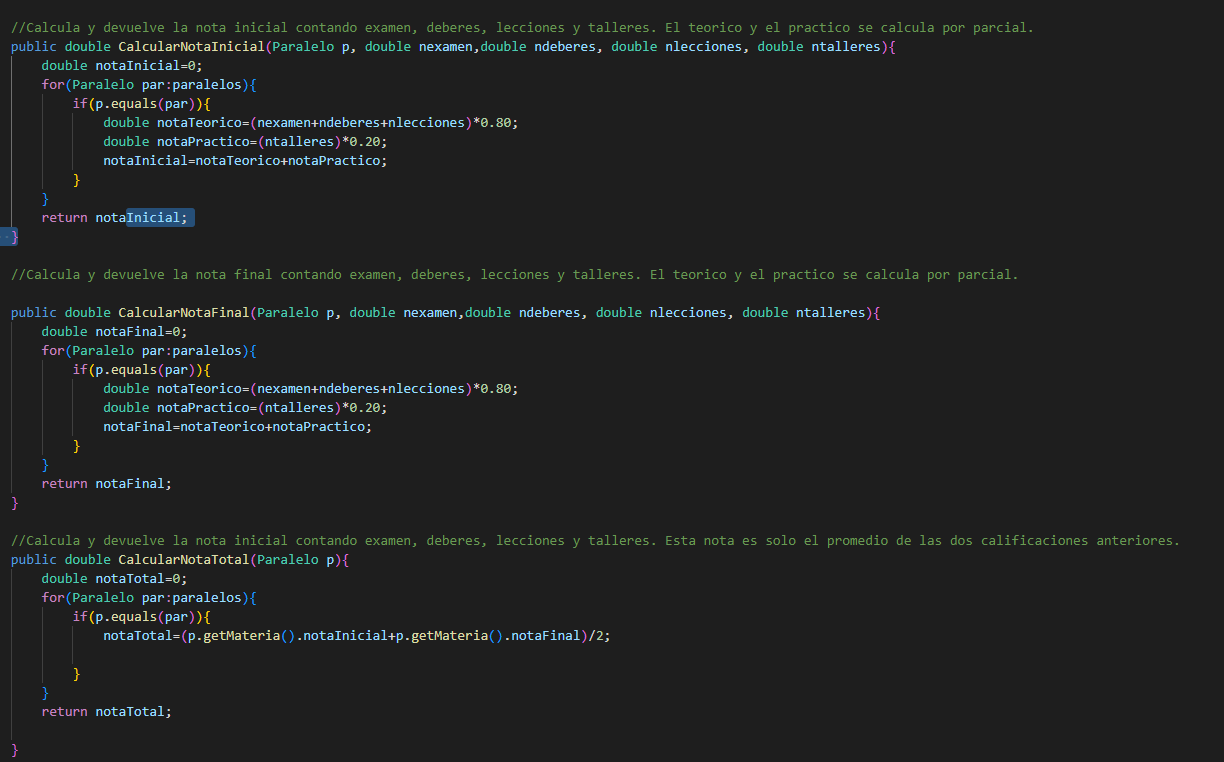
Después:

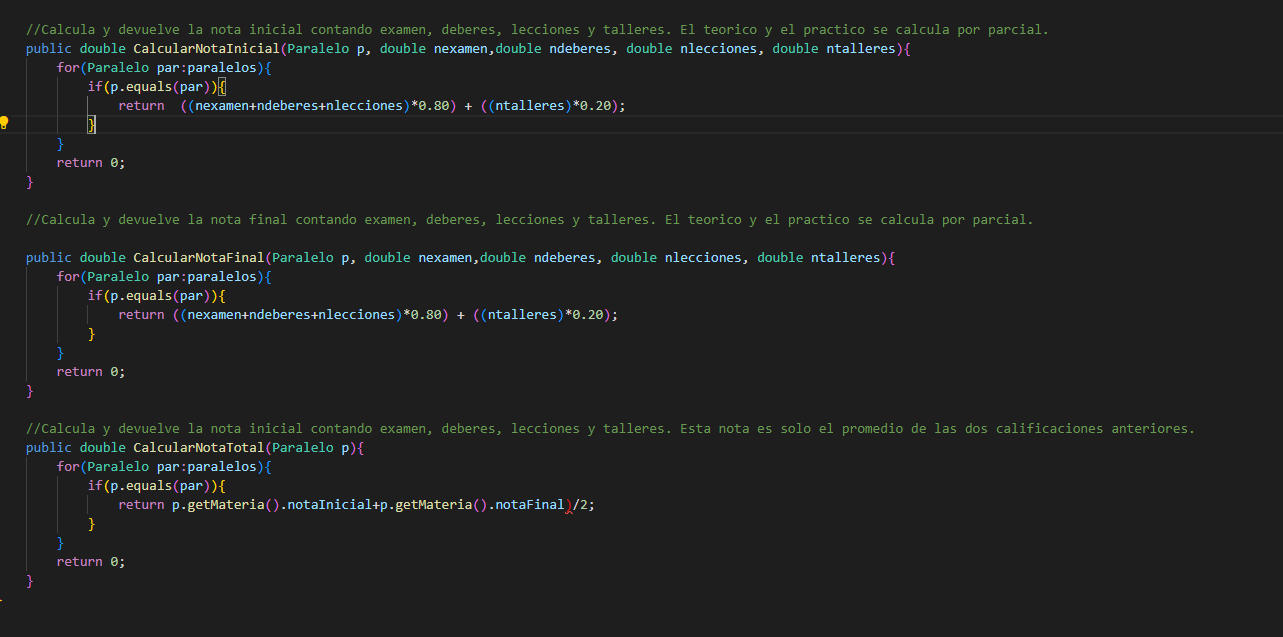


# Temporary field:

En los métodos CalcularNotaInicial, CalcularNotaFinal, CalcularNotaTotal, se están creando variables que se usan únicamente dentro de cada uno de estos métodos lo que incurre en el code smell de Temporary field. Mantener estas variables dentro de los métodos puede hacer que el código sea mas complicado de entender. Para resolver esto se hará uso de la técnica de Inline temp.

Antes:



Después:

En el método CalcularSueldo de la clase calcularSueldoProfesor también se encontró un temporary field.

Antes:

Texto

Descripción generada automáticamente

Después:

Texto

Descripción generada automáticamente

# Large class:

Los métodos CalcularNotaInicial, CalcularNotaFinal, CalcularNotaTotal, no permiten a la clase Estudiante cumplir con el principio de Single responsibility ya que estos métodos no tienen necesidad de estar en esta clase. Para solucionarlo se aplica la técnica de Extract class.

Antes:

Texto

Descripción generada automáticamente

Después:

Texto

Descripción generada automáticamente

# Lazy class:

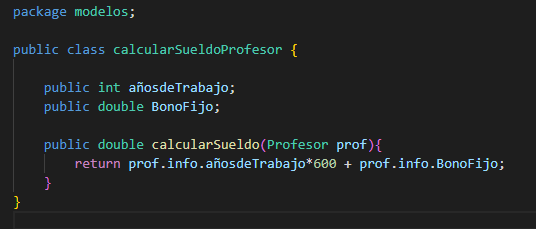
La clase InformacioAdicionalPrefesor y la clase calcularSueldoProfesor tienen relación entre ambas y no existe una necesidad de que estén separadas, ya que esto solo complica el entendimiento del programa. Para solucionar esto se aplica la técnica de Move Fields.

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteAntes:

Después:



# Data clumps:

Dentro de la clase Materia existen atributos que tienen relación entre ellos, pero no con los otros atributos por lo cual la mejor solución seria crear un objeto con estos datos. Para esto se usa la técnica de extract class.

Texto

Descripción generada automáticamenteAntes:

Después:

Texto

Descripción generada automáticamente

Repositorio: