**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

**DISEÑO DE SOFTWARE**

**TALLER DE REFACTORIZACIÓN**

**Integrantes**

Carlier Durango César

Carrasco Morán ENRIQUE

Castro Ascencio KARLA

**I TÉRMINO 2020**

Table of Contents

[Code Smells Refactorizados 3](#__RefHeading___Toc283_3987738830)

[Speculative Generality: Clase Profesor 3](#__RefHeading___Toc285_3987738830)

[Explicación, Asunciones y Consecuencias 3](#__RefHeading___Toc287_3987738830)

[Captura del código Inicial 3](#__RefHeading___Toc289_3987738830)

[Técnica de Refactorización 3](#__RefHeading___Toc321_3987738830)

[Código refactorizado 3](#__RefHeading___Toc323_3987738830)

[Lazy Class: Clase calcularSueldoProfesor 4](#__RefHeading___Toc291_3987738830)

[Explicación, Asunciones y Consecuencias 4](#__RefHeading___Toc293_3987738830)

[Captura del código Inicial 4](#__RefHeading___Toc295_3987738830)

[Técnica de Refactorización 4](#__RefHeading___Toc325_3987738830)

[Código refactorizado 4](#__RefHeading___Toc327_3987738830)

[Duplicated Code 5](#__RefHeading___Toc297_3987738830)

[Explicación, Asunciones y Consecuencias 5](#__RefHeading___Toc299_3987738830)

[Captura del código Inicial 5](#__RefHeading___Toc301_3987738830)

[Técnica de Refactorización 5](#__RefHeading___Toc329_3987738830)

[Código refactorizado 5](#__RefHeading___Toc331_3987738830)

[Innappropiade Intimacy 6](#__RefHeading___Toc303_3987738830)

[Explicación, Asunciones y Consecuencias 6](#__RefHeading___Toc305_3987738830)

[Captura del código Inicial 6](#__RefHeading___Toc307_3987738830)

[Técnica de Refactorización 6](#__RefHeading___Toc333_3987738830)

[Código refactorizado 6](#__RefHeading___Toc335_3987738830)

[Middle Man 7](#__RefHeading___Toc309_3987738830)

[Explicación, Asunciones y Consecuencias 7](#__RefHeading___Toc311_3987738830)

[Captura del código Inicial 7](#__RefHeading___Toc313_3987738830)

[Técnica de Refactorización 7](#__RefHeading___Toc337_3987738830)

[Código refactorizado 7](#__RefHeading___Toc339_3987738830)

[Long Paramether List 8](#__RefHeading___Toc315_3987738830)

[Explicación, Asunciones y Consecuencias 8](#__RefHeading___Toc317_3987738830)

[Captura del código Inicial 8](#__RefHeading___Toc319_3987738830)

[Técnica de Refactorización 8](#__RefHeading___Toc341_3987738830)

[Código refactorizado 8](#__RefHeading___Toc343_3987738830)

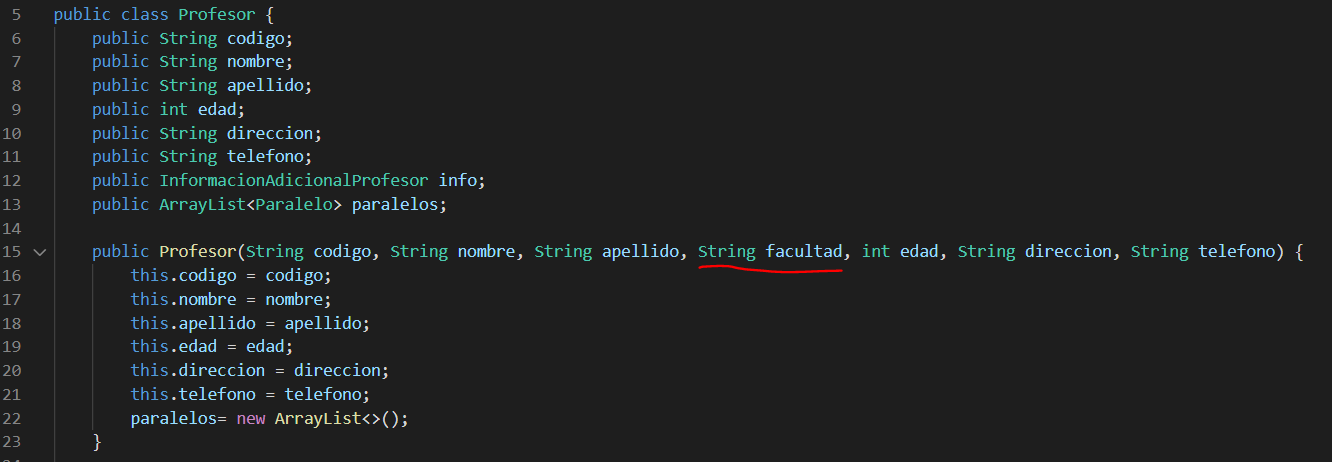
# Code Smells Refactorizados

## Speculative Generality: Clase Profesor

### Explicación, Asunciones y Consecuencias

Existe un parámetro inutilizado, <String Facultad> en el constructor de la clase Profesor. Es posible que este parámetro haya sido creado para una futura implementación de facultades para agrupar profesores y/o materias, además de que este dato se guarda por su cuenta en la clase InformacionAdicionalProfesor. Si esta continua en el código podría llegar a confundir y hacer más difícil el soporte si en algún momento se desea llamar a este constructor.

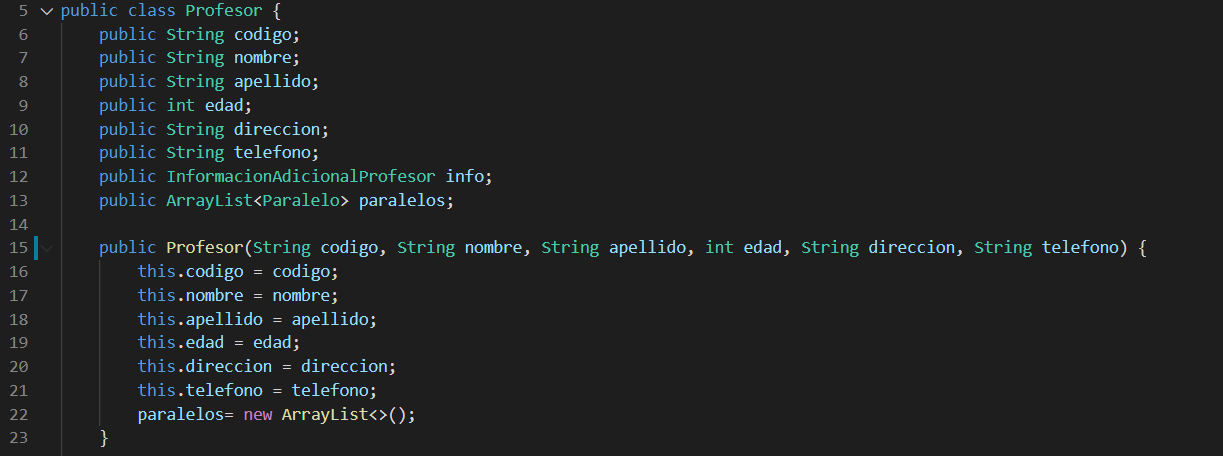
### Captura del código Inicial

Constructor en la línea 15, parámetro ”String facultad”, nunca es usado en el constructor ni es atributo de la clase.

### Técnica de Refactorización

Remove Parameter: Eliminamos el parámetro inutilizado del método original, y buscamos en el código cualquier referencia a este constructor y lo modificamos

### Código refactorizado

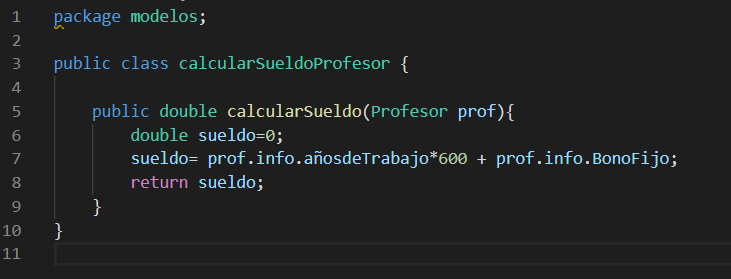


## Lazy Class: Clase calcularSueldoProfesor

### Explicación, Asunciones y Consecuencias

La clase calcularSueldoProfesor no tiene ninguna responsabilidad real más que un método para calcular un valor consiguiendo datos de otra clase. Dejarla dentro del código solo causará mayor cantidad de clases y un mantenimiento más difícil.

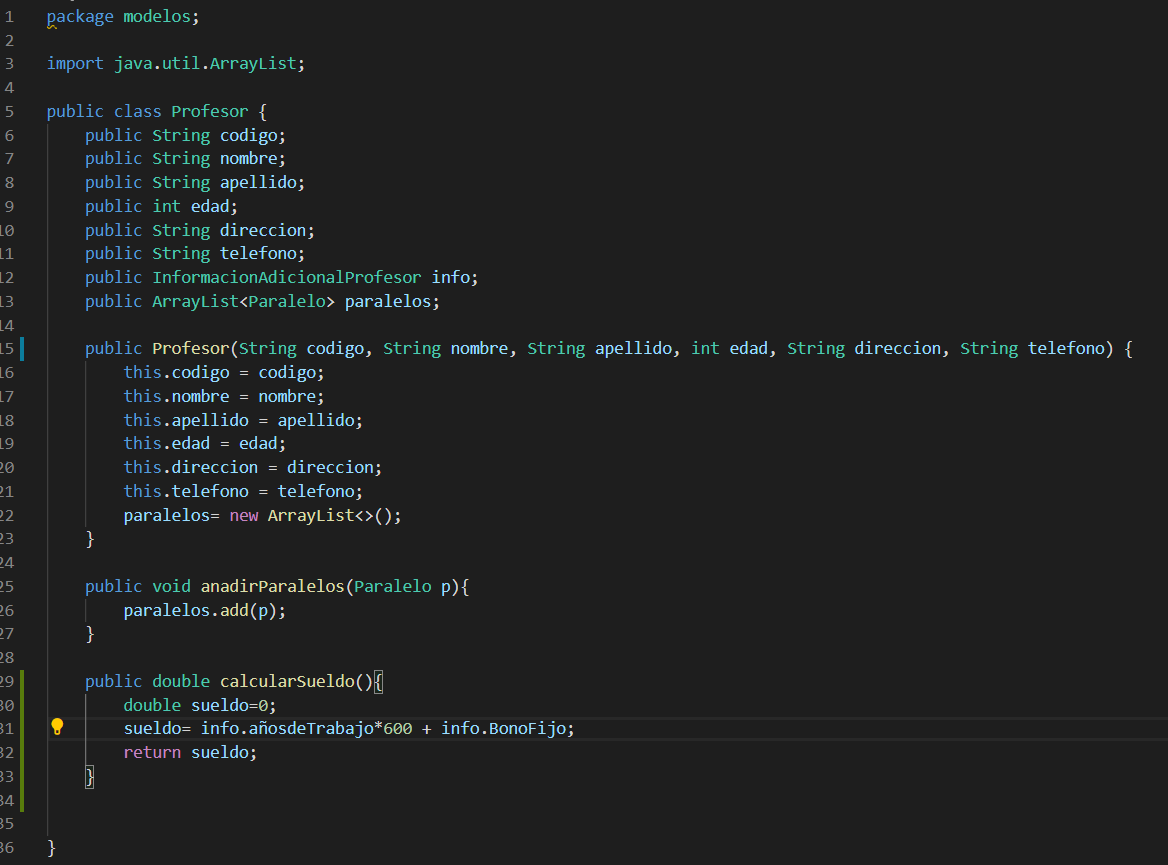
### Captura del código Inicial



### Técnica de Refactorización

Inline Class. Se moverán todos los atributos y métodos a una clase que pueda tener esa responsabilidad, en este caso, lo pasaremos a la clase Profesor, y se eliminará esta clase, refactorizando los lugares donde haya sido referenciada.

### Código refactorizado

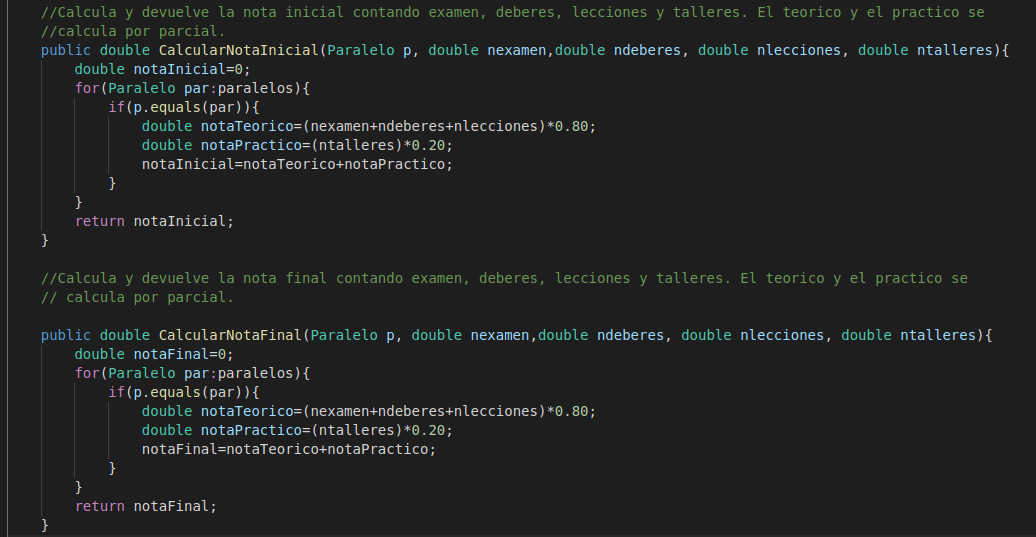


## Duplicated Code

### Explicación, Asunciones y Consecuencias

Existen 2 métodos dentro de la clase Estudiante, CalcularNotaInicial y CalcularNotaFinal , tienen exactamente la misma funcionalidad y reciben la misma cantidad de parámetros, esto genera una duplicación de código y aumento innecesario de líneas en la clase.

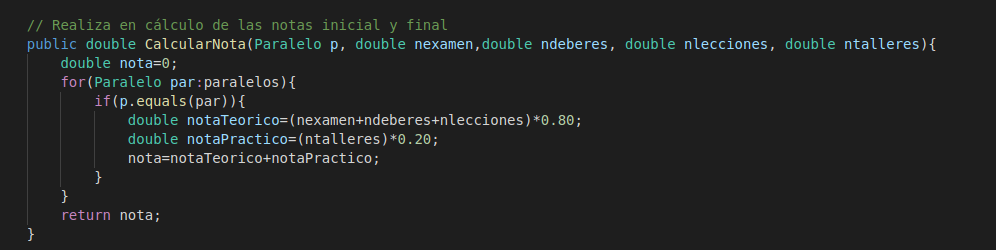
### Captura del código Inicial



### Técnica de Refactorización

Extract Method: Mover este código a un nuevo método independiente y reemplace el código antiguo con una llamada al método.

### Código refactorizado

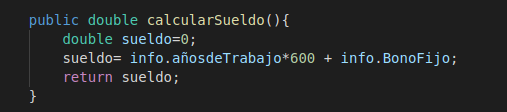


## Innappropiade Intimacy

### Explicación, Asunciones y Consecuencias

La clase Profesor usa atributos de la clase InformacionAdicionalProfesor generando este mal olor.

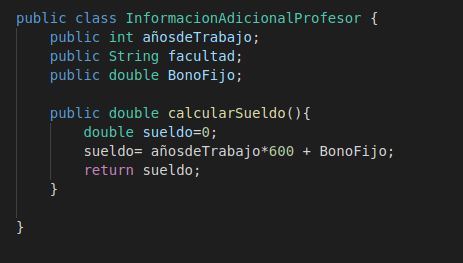
### Captura del código Inicial



### Técnica de Refactorización

Move Method: Crear un nuevo método en la clase que más usa el método, luego mover el código del método anterior allí. Convertir el código del método original en una referencia al nuevo método en la otra clase o elimínelo por completo y en nuestro caso lo eliminamos.

### Código refactorizado

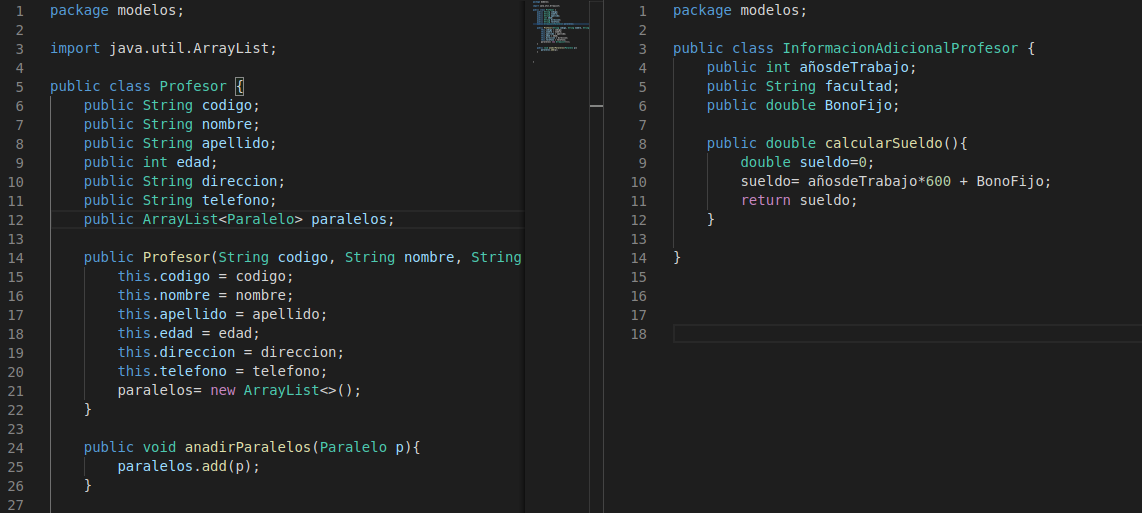


## Middle Man

### Explicación, Asunciones y Consecuencias

La clase InformacionAdicionalProfesor genera un único cálculo, el del sueldo, si la mayortía o total de informción general y base del Profesor se encuentra en una clase diferent esta no tiene razón de existencia.

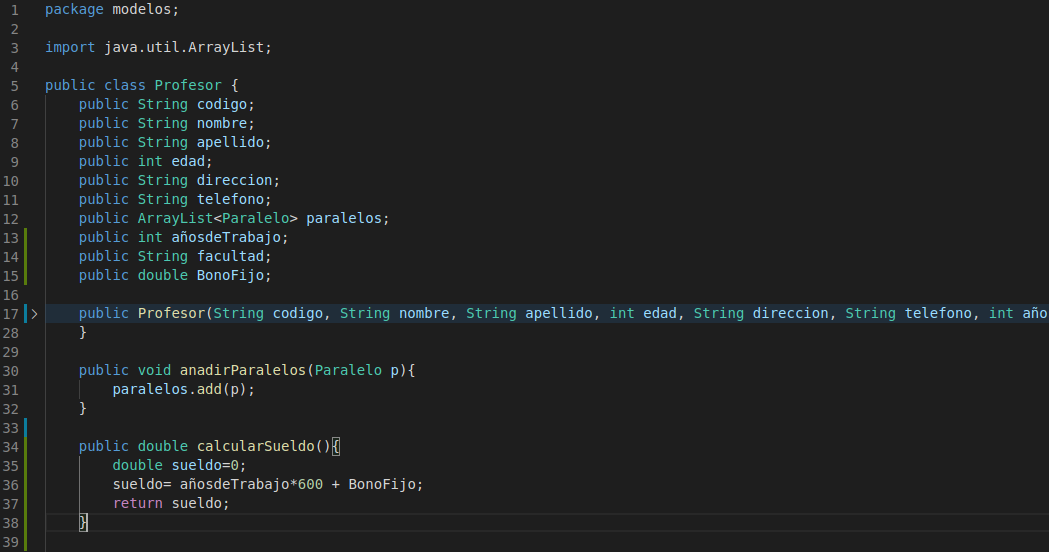
### Captura del código Inicial



### Técnica de Refactorización

Remove middle man: con esta técnica lo que buscamos en eliminar elementos intermediaron que añaden complejidad al código base, al eliminar la clase innecesaria los atributos y el único método que contenía se trasladan a la clase base Profesor.

### Código refactorizado

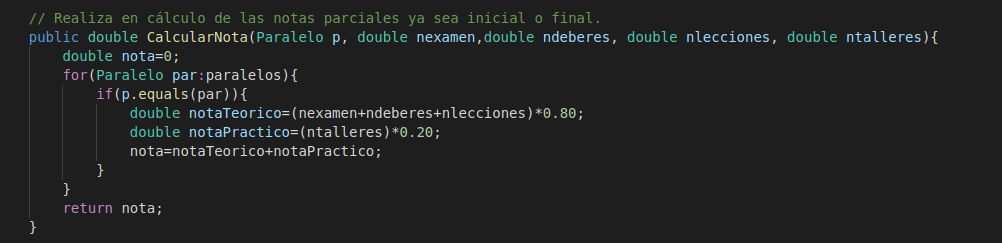


## Long Paramether List

### Explicación, Asunciones y Consecuencias

El método calculoNota definido en la clase Estudiante recibe una muy larga lista de parámetros.

### Captura del código Inicial



### Técnica de Refactorización

Introduce parameter object: en lugar de recibir los 4 parámetros enviaremos un objeto Notas que contendrá las diferentes notas.

### Código refactorizado

