

**Taller 08 Refactoring**

**Grupo**

**Diseño de software**

Oscar García

Juan Carlos Pisco

Ramos Pozo

Contenido

[Data Class 3](#_Toc92376407)

[Consecuencias 3](#_Toc92376408)

[Técnicas 3](#_Toc92376409)

[Encapsulate field 4](#_Toc92376410)

[Inline Class 4](#_Toc92376411)

[Temporary Field 5](#_Toc92376412)

[Consecuencias 5](#_Toc92376413)

[Técnicas 5](#_Toc92376414)

[Inline Temp 5](#_Toc92376415)

[Replace Magic Number with Symbolic Constant 5](#_Toc92376416)

[Feature Envy 6](#_Toc92376417)

[Consecuencias 6](#_Toc92376418)

[Técnicas 6](#_Toc92376419)

[Inline Class 6](#_Toc92376420)

[Alternative Classes with Different Interfaces 7](#_Toc92376421)

[Consecuencias 8](#_Toc92376422)

[Técnicas 8](#_Toc92376423)

[Extract Superclass 8](#_Toc92376424)

[Lazy Class 10](#_Toc92376425)

[Consecuencias 10](#_Toc92376426)

[Técnicas 11](#_Toc92376427)

[Replace Delegation with Inheritance 11](#_Toc92376428)

[Large Class 12](#_Toc92376429)

[Consecuencias 12](#_Toc92376430)

[Tecnicas 13](#_Toc92376431)

[Move Method 13](#_Toc92376432)

[Long Parameter List 14](#_Toc92376433)

[Consecuencias 14](#_Toc92376434)

[Técnicas 15](#_Toc92376435)

[Remove Parameter 15](#_Toc92376436)

# Data Class

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## Consecuencias

Estas clases solo contienen funcionalidad que es utilizada por otras clases y no poseen datos en los cuales puedan operar, por lo que al tener clases de esta forma empeora la organización y entendimiento del código.

## Técnicas

### Encapsulate field

Texto

Descripción generada automáticamente

### Inline Class

Imagen que contiene Texto

Descripción generada automáticamente

# Temporary Field

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

## Consecuencias

En este caso el campo de sueldo es creado para usarlo en el algoritmo y no cumple ningún propósito después, esta forma de programar no suele ser muy clara y carece de organización.

## Técnicas

### Inline Temp

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

### Replace Magic Number with Symbolic Constant

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

# Feature Envy

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## Consecuencias

Por lo general siempre que se necesite calcular el sueldo del profesor, se va a necesitar instanciar la clase calcularSueldoProfesor, por lo que ocasionaría duplicación de código, además que visualmente el código se vería desordenado ya que el método que utiliza los datos del profesor no se encuentra con los datos del profesor directamente afectando la legibilidad del código.

## Técnicas

### Inline Class

Tabla

Descripción generada automáticamente

# Alternative Classes with Different Interfaces

Tabla

Descripción generada automáticamente con confianza media

Tabla

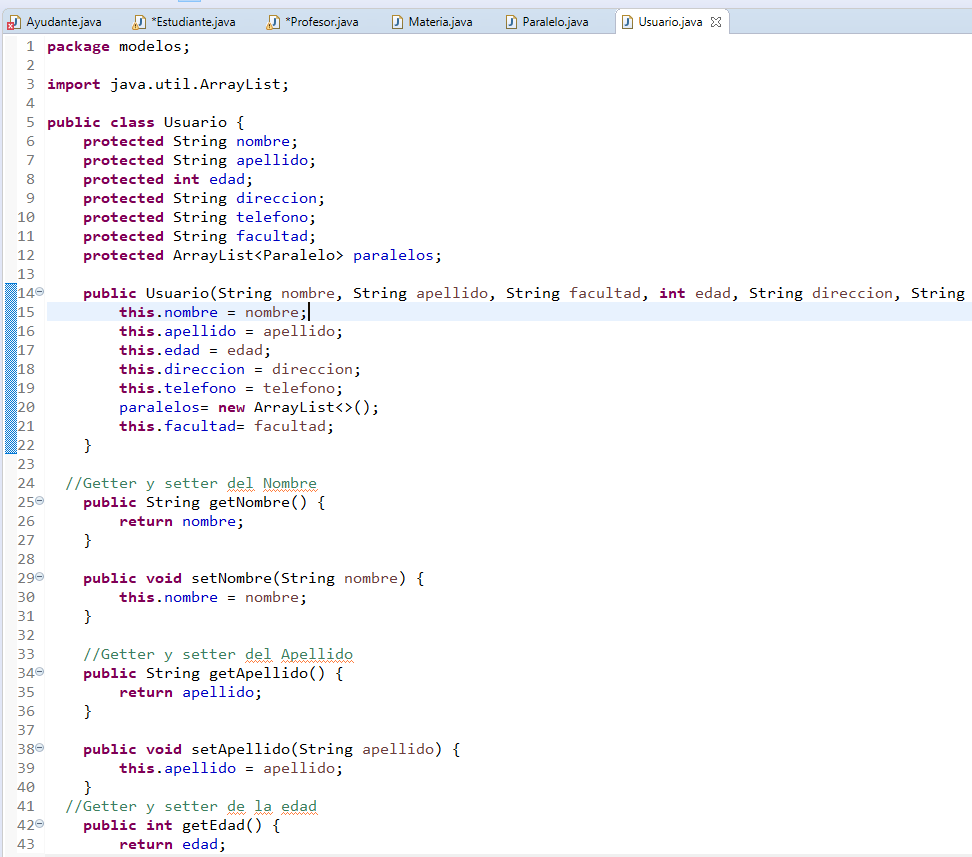
Descripción generada automáticamente con confianza media

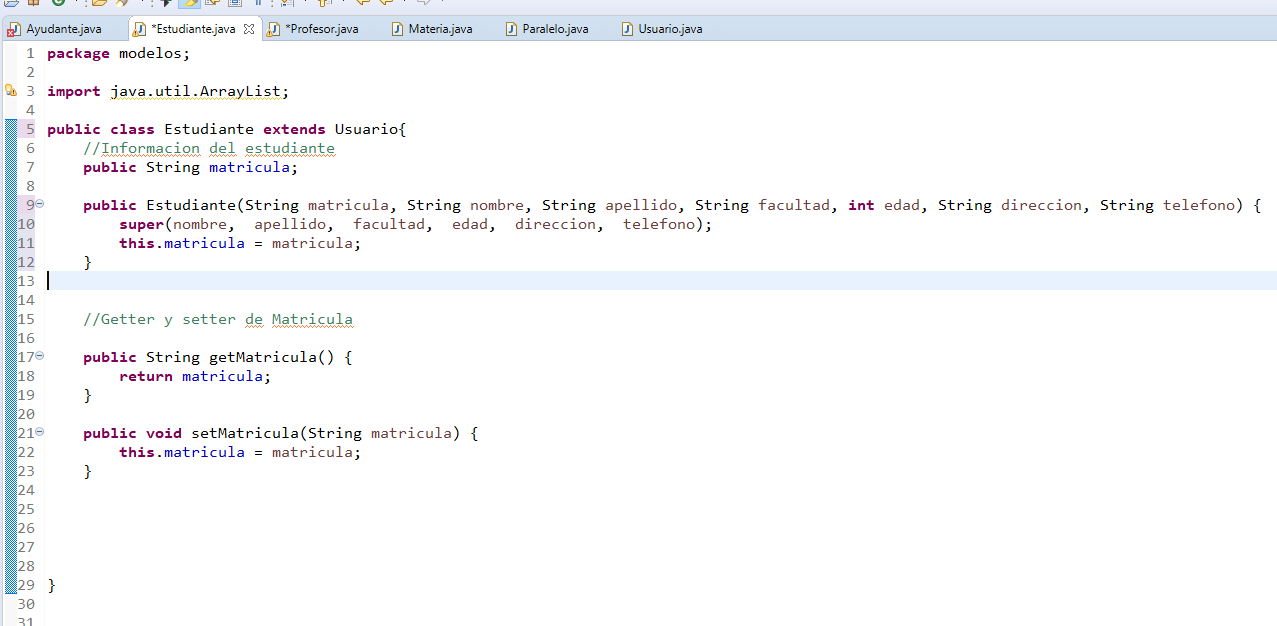
## Consecuencias

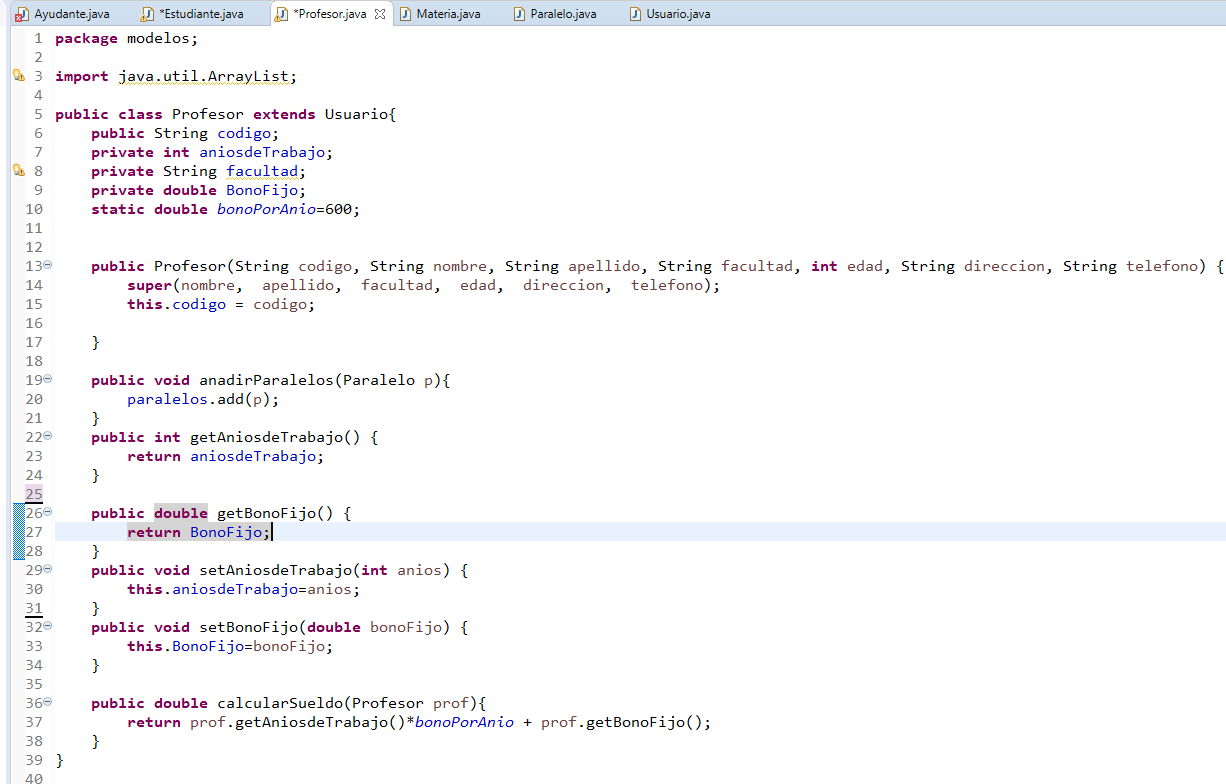
Ocasiona duplicación de código, ya que al haber dos clases o más clases que funcionen de forma similar, estas terminan haciendo uso de los mismos fragmentos de código, además visualmente el código se ve voluminoso, no tan legible y poco entendible.

## Técnicas

### Extract Superclass







# Lazy Class

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

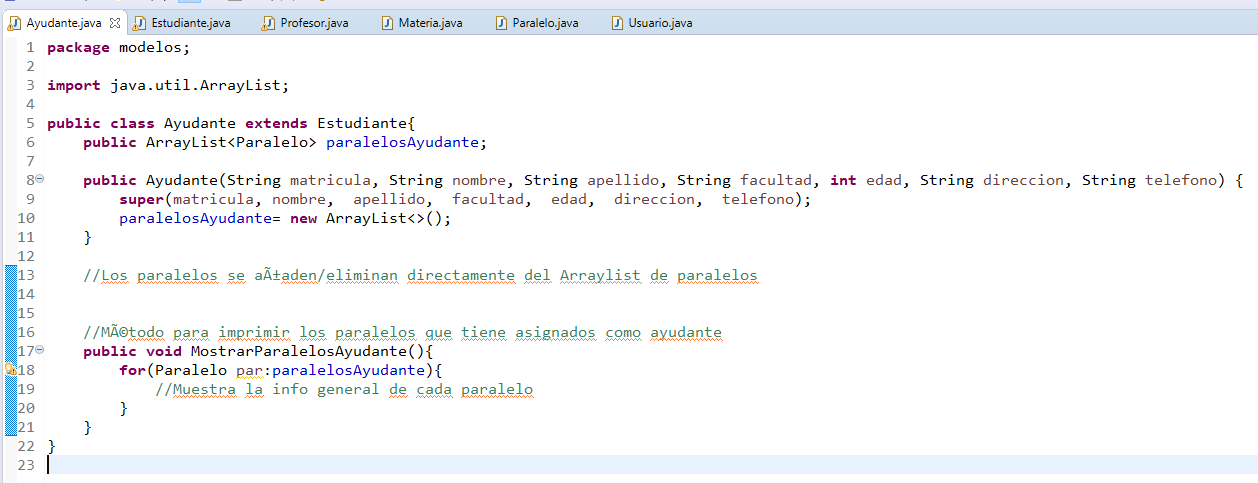
Descripción generada automáticamente

## Consecuencias

Por lo general, comprender y mantener clases toma tiempo y recursos, así que si una clase, como en el caso de Ayudante, no tiene suficientes responsabilidades hay que buscar una forma de que sea útil o hay que eliminarla.

## Técnicas

### Replace Delegation with Inheritance



# Large Class

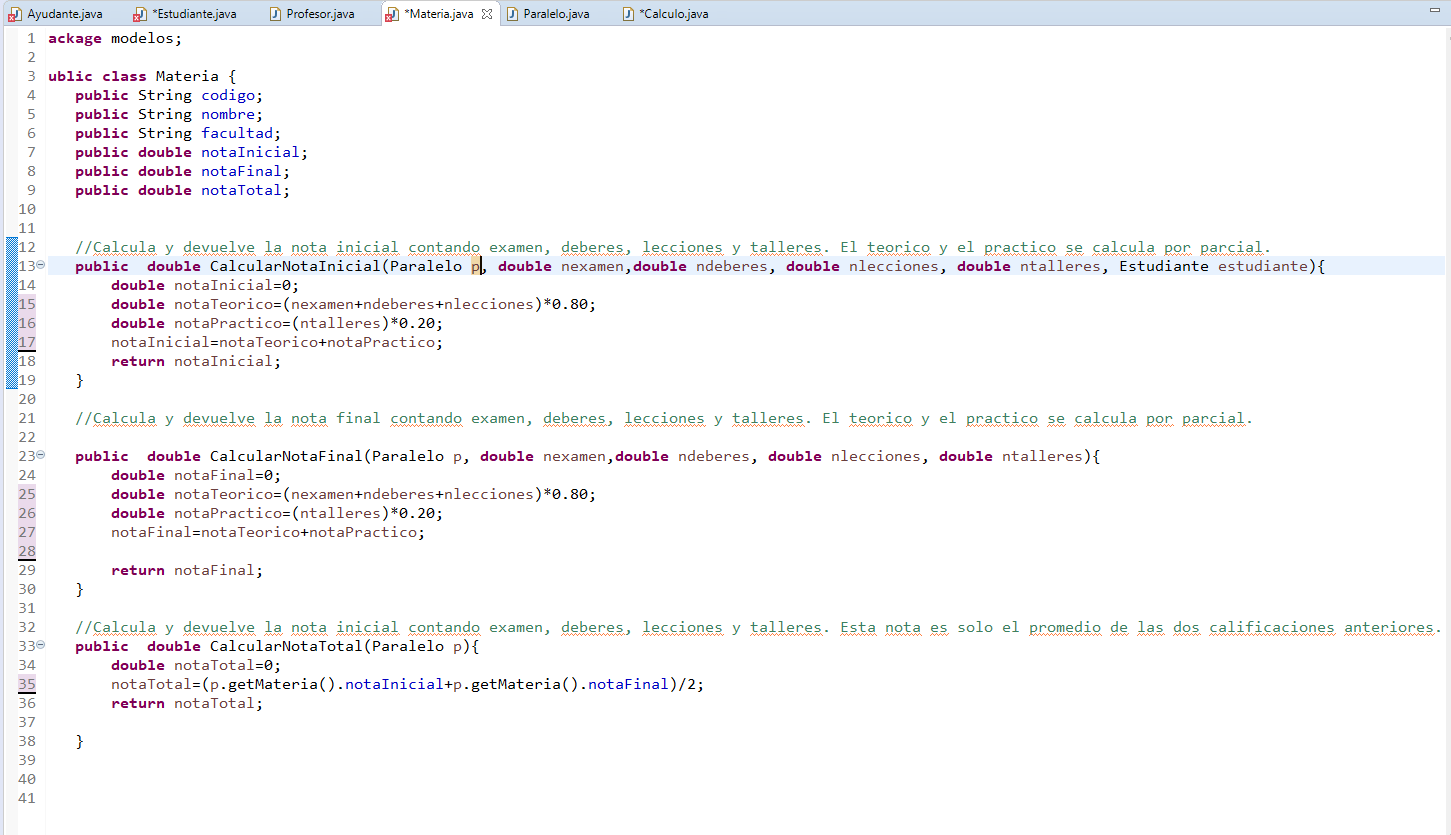


## Consecuencias

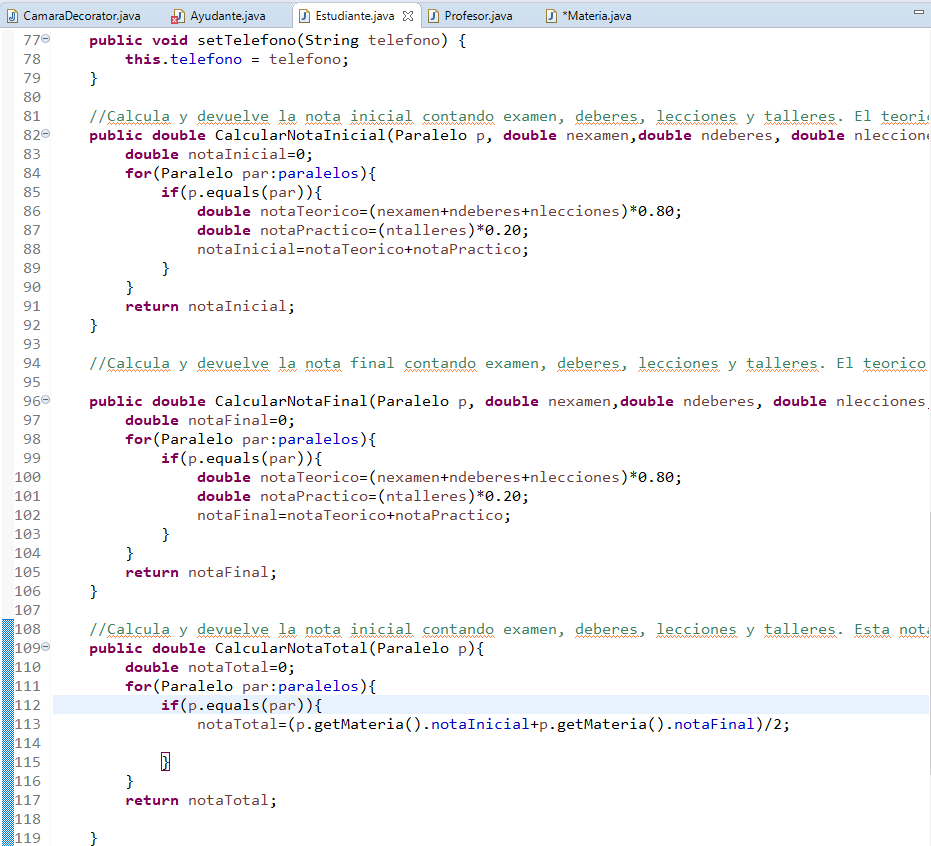
Las clases usualmente se van llenando de código a medida que el programa evoluciona, y una práctica muy común para evitar complicar las cosas es seguir añadiendo funcionalidades a una misma clase, en el caso de estudiante, una vez se agreguen más y más funcionalidades esto va a causar que el código se vuelva más complicado de entender, forzando a los desarrolladores a tener que recordar más aspectos derivados de esta misma clase.

## Tecnicas

### Move Method



# Long Parameter List



## Consecuencias

Una larga lista de parámetros es probablemente ocasionada luego de que varios tipos de algoritmos se mezclen en uno solo, o en el intento de hacer las clases sean más independientes con respecto a otras clases, estas listas se vuelven muy difíciles de comprender ya que a medida que crecen se vuelven contradictorias y difíciles de utilizar, por lo que es más apropiado remover en lo posible estos parámetros, en la forma que se vea necesaria.

## Técnicas

### Remove Parameter

