
PRIS – Source Monitor

Curso 2020/2021

Juan José Pallarés Sánchez

Índice

- 1. Introducción 1
- 2. Baseline 1..... 1

1. Introducción

En este documento se recoge la revisión realizada sobre el código de “Versionado” para la práctica de SourceMonitor.

2. Baseline 1

En primer lugar y como se puede observar en la gráfica (Fig. 1) a lo largo de las diferentes versiones se observa como las líneas de código han ido decayendo a la par que la complejidad.

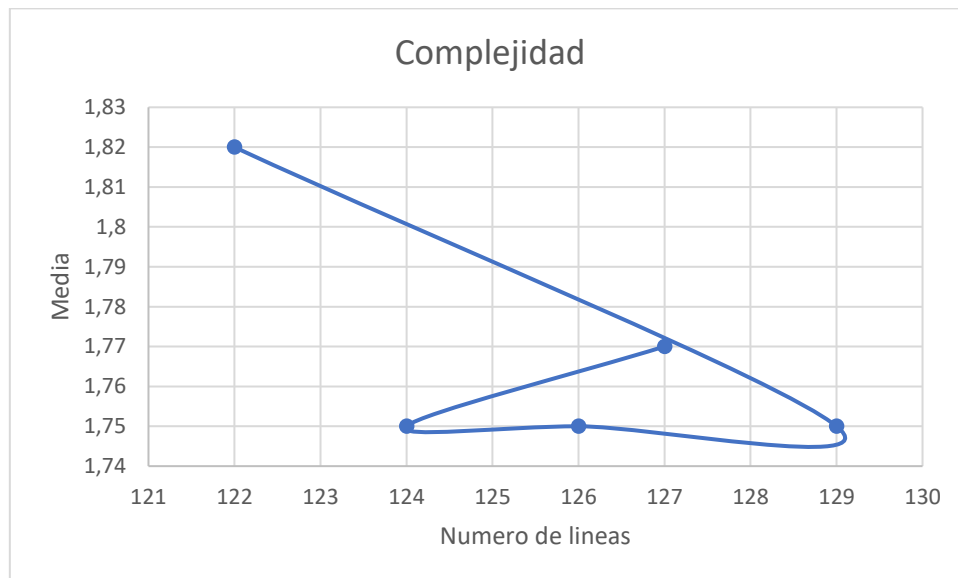


Fig. 1 Complejidad

Esto destaca el hecho de la eliminación en la primera versión de algunos métodos que consigue reducir la complejidad y líneas del código. Donde podemos destacar que se ha trasladado el siguiente código a una clase mejor ubicada dejando “Rental” como clase de modelo para abstraer y entender mejor el código.

```
private double amountOf(Rental aRental) {  
    double thisAmount = 0;  
    switch (aRental.getMovie().getPriceCode()) {  
        case Movie.REGULAR:  
            thisAmount += 2;  
            if (aRental.getDaysRented() > 2){  
                thisAmount += (aRental.getDaysRented() - 2) * 1.5;  
            }  
            break;  
        case Movie.NEW_RELEASE:  
            thisAmount += aRental.getDaysRented() * 3;  
            break;  
        case Movie.CHILDRENS:  
            thisAmount += 1.5;  
            if (aRental.getDaysRented() > 3){  
                thisAmount += (aRental.getDaysRented() - 3) * 1.5;  
            }  
    }  
}
```

```

        break;
    default: break;
}
return thisAmount;
}

```

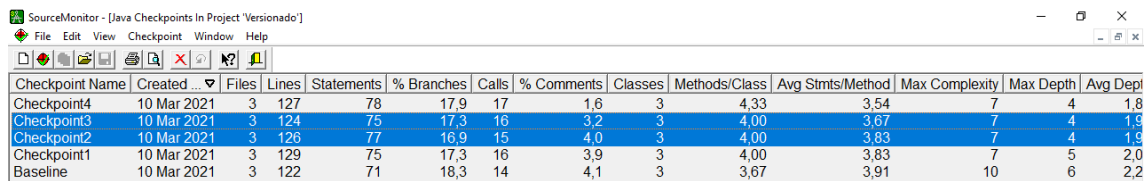
Durante el transcurso de las versiones de la 2 a la 3 se ha vuelto a implementar el método en Rental devolviéndole la complejidad a esta clase, pero se mantiene el método “getFrequentRenterPoints” dentro fuera de esta.

```

public int getFrequentRenterPoints(){
if ((getMovie().getPriceCode() == Movie.NEW_RELEASE)
&& getDaysRented() > 1) return 2;
else
    return 1;
}

```

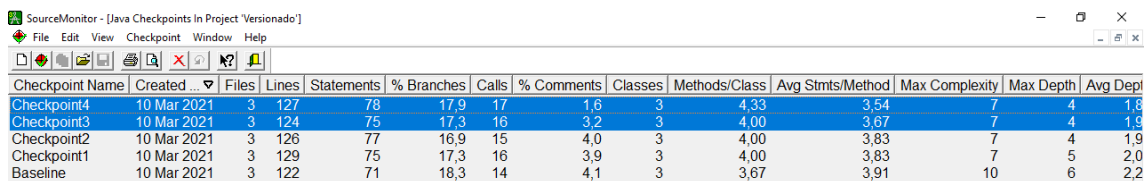
Durante la siguiente versión se ha eliminado la renta del código haciendo reducir el número de líneas de código otra vez, pero sigue manteniendo la complejidad. Seguramente se hayan solventado



Checkpoint Name	Created	Files	Lines	Statements	% Branches	Calls	% Comments	Classes	Methods/Class	Avg Strmts/Method	Max Complexity	Max Depth	Avg Depth
Checkpoint4	10 Mar 2021	3	127	78	17.9	17	1.6	3	4.33	3.54	7	4	1.8
Checkpoint3	10 Mar 2021	3	124	75	17.3	16	3.2	3	4.00	3.67	7	4	1.9
Checkpoint2	10 Mar 2021	3	126	77	16.9	15	4.0	3	4.00	3.83	7	4	1.9
Checkpoint1	10 Mar 2021	3	129	75	17.3	16	3.9	3	4.00	3.83	7	5	2.0
Baseline	10 Mar 2021	3	122	71	18.3	14	4.1	3	3.67	3.91	10	6	2.2

Fig. 2 Source Monitor Check 3-4

En la última versión se ha devuelto el método “getFrequentRenterPoints” a la clase rental. Además, se ha mejorado el método statement eliminado los puntos de frecuencia devolviéndolos a la clase de rental.



Checkpoint Name	Created	Files	Lines	Statements	% Branches	Calls	% Comments	Classes	Methods/Class	Avg Strmts/Method	Max Complexity	Max Depth	Avg Depth
Checkpoint4	10 Mar 2021	3	127	78	17.9	17	1.6	3	4.33	3.54	7	4	1.8
Checkpoint3	10 Mar 2021	3	124	75	17.3	16	3.2	3	4.00	3.67	7	4	1.9
Checkpoint2	10 Mar 2021	3	126	77	16.9	15	4.0	3	4.00	3.83	7	4	1.9
Checkpoint1	10 Mar 2021	3	129	75	17.3	16	3.9	3	4.00	3.83	7	5	2.0
Baseline	10 Mar 2021	3	122	71	18.3	14	4.1	3	3.67	3.91	10	6	2.2

Fig. 3 Source Monitor Check 4-5

Por último los diferentes cambios también pueden verse reflejados en el proyecto a través de la grafica de líneas que ofrece SourceMonitor

