

# Ejemplo de refactoring

DAM M5-UF2

Gastón Santiago Rossi

14/05/2019

Índice de contenido

Introducción.....3

Ejercicio 1. Refactoring.....3

Conclusión.....4

# Introducción

En este ejercicio veremos un caso de refactorización de código y en concreto de que tipo de refactorización nos referimos.

## Ejercicio 1. Refactoring

Es obvio el problema que tiene este código, si miras este código tardarías en entender que es lo que hace, ya que las variables tienen nombres muy abstractos que no aportan nada de información.

```
import java.io.*;

public class Ejercicio2_GastonRossi{
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader (new InputStreamReader(System.in));
        System.out.println("Numeros divisibles");
        System.out.print("Introduce un numero: ");
        int X = Integer.parseInt(reader.readLine());
        int i = 0;
        int Y = 0;
        System.out.print("Los multiples son: ");
        for (i = 1; i < X + 1; i = i + 1){
            Y = X % i;
            if (i == X){
                System.out.print(i);
            }
            else if (result == 0){
                System.out.print(i + ", ");
            }
        }
    }
}
```

Aquí se puede ver como es mucho mas simple entender el código, en este caso refactorizamos el nombre de todas las variables.

```
import java.io.*;

public class Ejercicio2_GastonRossi{
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader (new InputStreamReader(System.in));
        System.out.println("Numeros divisibles");
        System.out.print("Introduce un numero: ");
        int Numero = Integer.parseInt(reader.readLine());
        int Resultado;
        System.out.print("Los multiples son: ");
        for (int PosiblesMultiples = 1; PosiblesMultiples < Numero + 1; PosiblesMultiples++){
            Resultado = Numero % PosiblesMultiples;
            if (PosiblesMultiples == Numero){
                System.out.print(PosiblesMultiples);
            }
            else if (Resultado == 0) {
                System.out.print(PosiblesMultiples + ", ");
            }
        }
    }
}
```

# Conclusión

El refactoring a mi criterio debería ser algo que se aplica cada vez que observamos un código, sí cada vez que entras en un código cambias algo y lo dejas mejor, tu código no hará más que mejorar y seguir mejorando.

Creo que el refactoring es esencial para cualquier programador.