

## Reentrega Hito 2

### Trabajo de Promoción OO2

Integrantes: Andrade, Ivana Micaela (Legajo: 12932/6) y Posada, Ignacio (Legajo: 12956/4)

#### 2 ) Bad Smells:

Code Smell	Impl. provista	Impl. propia	Refactoring
Duplicate code	Si, StreamTweetReceiver >> basicReport , reportAverageHashtags;, reportAverageWords;, reportTweetCount.	Si, Filtro >> colección, filtro.	Aplicar Extract Method del codigo repetido en Impl. Provista. Aplicar Form Template Method en impl. Propia.
Large class	Si, la clase StreamTweetReceiver.	No.	Aplicar Extract method.
Dead Code	Si, StreamTweetReceiver >> basicReport .	No.	Eliminar el codigo que ya no se utiliza. Si es una clase, reusarla con nuevos metodos. En la clase provista, la solucion es darle uso al codigo.
Long method	Si, StreamTweetReceiver >> reportAverageHashtags;, reportAverageWords: .	Si, ProcesarTweets >> masSeguidores.	Realizar Extract Method de las diferentes etapas de procesamiento.
Data class	No.	No.	Realizar Extract Method de las metodos que usan los datos de la clase.
Feature envy	No.	No.	Mover los metodos que usan los servicios de la clase. (Extract method)
Long parameters list	No.	No.	Achicar la cantidad de parametros usando alguna de las 3 tecnicas: Replace Parameter with Method ,Preserve Whole Object o Introduce Parameter Object .
Switch statements	No.	No.	Eliminar los condicionales usando Polimorfismo.

Excessively long identifiers	No.	No.	Renombrar los nombres de variable o metodos con palabras mas cortas sin perder representatividad.
Excessively short identifiers	Si: algunas variables de instancia en la clase StreamTweetReceiver y una variable temporal en un metodo de clase de la misma.	Si, la variable de instancia palTot en la clase ProcesarFlujoDeTweet.	Renombrar los nombres con valores mas representativos sin que sea demasiado largo.

#### 4) Pasos para agregar un nuevo filtro:

Para agregar un nuevo filtro, se debe crear una subclase de la clase Filtro, que a su vez es subclase de la clase Tarea.

Se tiene que redefinir el metodo #filtro:. Este metodo contiene solamente la condicion del filtro que desea crear. Recordar que estos filtros son genericos, eso quiere decir que no se crea un filtro por un dato en especifico, sino un filtro sobre algun campo del tweet. Por ejemplo, no se puede crear un filtro por una fecha de creacion en especifico, sino que en su lugar se crea uno filtro que reciba como parametro la fecha. Esa variables que indica el criterio del filtro se llama 'criterio'.

Asegurarse que el metodo #filtro: devuelva un valor booleano.

#### Pasos para agregar una nueva tarea:

Para agregar una nueva tarea, se debe crear una subclase de la clase Tarea.

En el caso de que la nueva tarea fuera similar a alguna clase de tarea existente, deberia agregarse como subclase de esa misma.

En cualquiera de los casos, el metodo que se tiene que redefinir es el #tarea:. En ese metodo se realiza el labor correspondiente a esa tarea. Se pueden agregar otros metodos si fuese necesario modularizar, pero recordar que esos metodos deben ser llamados desde #tarea:.

#### 5)

Con las siguiente linea se instancia la aplicación ejemplo → miObj:=Ejemplo nuevo.

MiObj comenzar: '#Hashtag'.

Con la siguiente linea se frena el flujo de tweets → miObj parar .

Ambos codigos se ejecutan en el playground por separado.

