

BAD SMELL: Código duplicado

QuestionRetriever>>retrieverQuestions: aUser

| qRet temp followingCol topicsCol newsCol popularTCol averageVotes|

qRet := OrderedCollection new.

option = #social ifTrue:[

followingCol := OrderedCollection new.

aUser following do:[:follow | followingCol addAll: follow questions].

temp := followingCol asSortedCollection:[:a :b | a positiveVotes size > b positiveVotes size].

qRet := temp last: (100 min: temp size).

].

option = #topics ifTrue:[

topicsCol := OrderedCollection new.

aUser topics do:[:topic | topicsCol addAll: topic questions].

temp := topicsCol asSortedCollection:[:a :b | a positiveVotes size > b positiveVotes size].

qRet := temp last: (100 min: temp size).

].

option = #news ifTrue:[

newsCol := OrderedCollection new.

cuoora questions do:[:q | (q timestamp asDate = Date today) ifTrue: [newsCol add: q]].

temp := newsCol asSortedCollection:[:a :b | a positiveVotes size > b positiveVotes size].

qRet := temp last: (100 min: temp size).

].

option = #popularToday ifTrue:[

popularTCol := OrderedCollection new.

cuoora questions do:[:q | (q timestamp asDate = Date today) ifTrue: [popularTCol add: q]].

averageVotes := (cuoora questions sum: [:q | q positiveVotes size]) / popularTCol size.

temp := (popularTCol select:[:q | q positiveVotes size >= averageVotes]) asSortedCollection:[:a :b | a positiveVotes size > b positiveVotes size].

qRet := temp last: (100 min: temp size).

].

^qRet reject:[:q | q user = aUser].

1) Extract Method en #retrieverQuestions: aUser

| qRet followingCol topicsCol newsCol popularTCol averageVotes|

option = #social ifTrue:[

followingCol := OrderedCollection new.

aUser following do:[:follow | followingCol addAll: follow questions].

qRet := sortQuestions: followingCol.

].

option = #topics ifTrue:[

topicsCol := OrderedCollection new.

aUser topics do:[:topic | topicsCol addAll: topic questions].

qRet := sortQuestions: topicsCol.

].

option = #news ifTrue:[

newsCol := OrderedCollection new.

cuoora questions do[:q | (q timestamp asDate = Date today) ifTrue: [newsCol add: q]].

qRet := sortQuestions: newsCol.

].

option = #popularToday ifTrue:[

cuoora questions do[:q | (q timestamp asDate = Date today) ifTrue: [popularTCol add: q]].

averageVotes := (cuoora questions sum: [:q | q positiveVotes size]) / popularTCol size.

qRet := sortQuestions: (popularTCol select[:q | q positiveVotes size >= averageVotes]).

].

^qRet reject[:q | q user = aUser].

QuestionRetriever>> sortQuestions: questionColl

```
| questionReturn |  
    questionReturn := questionColl asSortedCollection:[ :a :b | a    positiveVotes size > b  
positiveVotes size ].  
  
    questionReturn := questionReturn last: (100 min: questionReturn size).  
  
    ^questionReturn.
```

2) Extract Method en #retrieverQuestions: aUser

```
| qRet followingCol topicsCol newsCol popularTCol averageVotes |  
    option = #social ifTrue:[  
        followingCol := OrderedCollection new.  
        aUser following do:[ :follow | followingCol addAll: follow questions ].  
        qRet := sortQuestions: followingCol.  
    ].  
    option = #topics ifTrue:[  
        topicsCol := OrderedCollection new.  
        aUser topics do:[ :topic | topicsCol addAll: topic questions ].  
        qRet := sortQuestions: topicsCol.  
    ].  
    option = #news ifTrue:[  
        newsCol := filterQuestionsToday  
        qRet := sortQuestions: newsCol.  
    ].  
    option = #popularToday ifTrue:[  
        popularTCol := filterQuestionsToday.  
        averageVotes := (cuoora questions sum: [:q | q positiveVotes size ]) / popularTCol size.  
        QRet := sortQuestions: (popularTCol select:[:q | q positiveVotes size >= averageVotes]).  
    ].
```

```
^qRet reject:[:q | q user = aUser].
```

QuestionRetriever>> filterQuestionsToday

```
| tmpCollec |  
tmpCollec := OrderedCollection new.  
cuoora questions do:[:q | (q timestamp asDate = Date today) ifTrue: [tmpCollec add: q]].  
  
^ tmpCollec.
```

BAD SMELL: Método largo.

QuestionRetreiver>> retrieverQuestions: aUser

```
| qRet followingCol topicsCol newsCol popularTCol averageVotes |  
option = #social ifTrue:[  
    followingCol := OrderedCollection new.  
    aUser following do:[ :follow | followingCol addAll: follow questions ].  
    qRet:= sortQuestions: followingCol.  
].  
option = #topics ifTrue:[  
    topicsCol := OrderedCollection new.  
    aUser topics do:[ :topic | topicsCol addAll: topic questions ].  
    qRet := sortQuestions: topicsCol.  
].  
option = #news ifTrue:[  
    newsCol := filterQuestionsToday  
    qRet := sortQuestions: newsCol.  
].
```

```

option = #popularToday ifTrue:[
    popularTCol := filterQuestionsToday.
    averageVotes := (cuoora questions sum: [:q | q positiveVotes size ]) / popularTCol size.
    qRet := sortQuestions: (popularTCol select:[:q | q positiveVotes size >= averageVotes]).
].
^qRet reject:[:q | q user = aUser].

```

3) Replace Temp With Query: **#retrieverQuestions: aUser**

QuestionRetriever>> retrieverQuestions: aUser

```

| qRet followingCol topicsCol newsCol popularTCol |
option = #social ifTrue:[
    followingCol := OrderedCollection new.
    aUser following do: [:follow | followingCol addAll: follow questions ].
    qRet := sortQuestions: followingCol.
].
option = #topics ifTrue:[
    topicsCol := OrderedCollection new.
    aUser topics do: [:topic | topicsCol addAll: topic questions ].
    qRet := sortQuestions: topicsCol.
].
option = #news ifTrue:[
    newsCol := filterQuestionsToday
    QRet := sortQuestions: newsCol.
].
option = #popularToday ifTrue:[
    popularTCol := filterQuestionsToday.
    QRet := sortQuestions: (popularTCol select:[:q | q positiveVotes size >= averageVotes
popularTCol])).

```

```
].  
^qRet reject:[:q | q user = aUser].
```

QuestionRetriever>> averageVotes: coll

```
^(cuoora questions sum: [:q | q positiveVotes size ]) / coll size.
```

4) Create subclass questionSocial and Push down method en **#retrieverQuestions: aUser**

QuestionRetriever>> retrieverQuestions: aUser

```
| qRet topicsCol newsCol popularTCol |  
option = #topics ifTrue:[  
    topicsCol := OrderedCollection new.  
    aUser topics do:[ :topic | topicsCol addAll: topic questions ].  
    qRet := sortQuestions: topicsCol.  
].  
option = #news ifTrue:[  
    newsCol := filterQuestionsToday  
    QRet := sortQuestions: newsCol.  
].  
option = #popularToday ifTrue:[  
    popularTCol := filterQuestionsToday.  
    qRet := sortQuestions: (popularTCol select:[:q | q positiveVotes size >= averageVotes  
popularTCol]).  
].  
^qRet reject:[:q | q user = aUser].
```

QuestionSocial>> retrieverQuestions: aUser

```
| followingCol |  
followingCol := OrderedCollection new.  
aUser following do:[ :follow | followingCol addAll: follow questions ].  
  
^(self sortQuestions:followingCol) reject:[:q | q user = self ].
```

5) Create subclass questionTopics and Push down method en
#retrieverQuestions: aUser

QuestionRetriever>> retrieverQuestions: aUser

```
| qRet newsCol popularTCol |  
option = #news ifTrue:[  
    newsCol := filterQuestionsToday  
    QRet := sortQuestions: newsCol.  
].  
option = #popularToday ifTrue:[  
    popularTCol := filterQuestionsToday.  
    qRet := sortQuestions: (popularTCol select:[:q | q positiveVotes size >= averageVotes  
popularTCol])).  
].  
^qRet reject:[:q | q user = aUser].
```

QuestionTopics>> retrieverQuestions: aUser

```
| topicsCol |  
topicsCol := OrderedCollection new.  
aUser topics do:[ :topic | topicsCol addAll: topic questions ].
```

```
^(self sortQuestions:topicsCol) reject:[ :q | q user = self ].
```

6) Create subclass questionNews and Push down method en **#retrieverQuestions: aUser**

```
QuestionRetriever>> retrieverQuestions: aUser
```

```
| qRet popularTCol |  
    option = #popularToday ifTrue:[  
        popularTCol := filterQuestionsToday.  
        qRet := sortQuestions: (popularTCol select:[ :q | q positiveVotes size >= averageVotes  
popularTCol])).  
    ].  
    ^qRet reject:[ :q | q user = aUser].
```

```
QuestionNews>> retrieverQuestions: aUser
```

```
| newsCol |  
newsCol := self filterQuestionsToday..  
    ^ (self sortQuestions: newsCol) reject: [ :q | q user = aUser ]
```

7) Create subclass questionPopularToday and Push down method en **#retrieverQuestions: aUser**

```
QuestionRetriever>> retrieveQuestions: aUser
```

```
self subclassResponsibility.
```

```
QuestionPopularToday >> retrieverQuestions: aUser
```



```

    | popularTCol |
    popularTCol := self filterQuestionsToday.
    popularTCol := (popularTCol select[:q | q positiveVotes size >= self averageVotes: popularTCol
    ]).

    ^ (self sortQuestions:popularTCol) reject[:q | q user = aUser ].

```

Los modulos de testeo fueron afectados y se realizaron los siguientes cambios:

Código actual:

QuestionRetrieverTest>> SetUp

...

...

socialRetriever := QuestionRetriever new: cuoora and: #social.

topicsRetriever := QuestionRetriever new: cuoora and: #topics.

newsRetriever := QuestionRetriever new: cuoora and: #news.

popularTodayRetriever := QuestionRetriever new: cuoora and: #popularToday.

Codigo resultante:

QuestionRetrieverTest>> SetUp

...

...

socialRetriever := **questionSocial** new: cuoora.

topicsRetriever := **questionTopics** new: cuoora.

newsRetriever := **questionNews** new: cuoora.

popularTodayRetriever := **questionPopularToday** new: cuoora.

BAD SMELL: CÓDIGO DUPLICADO

En las clases Answer y Question se duplican todos los get y set de las variables:

“timestamp”, “user”, “votes”, “description”.

Se creó una nueva clase padre llamada Publication que contiene las 4 variables con los get y set de cada una.

Answer	Question
<div><div>addVote:</div><div>description</div><div>description:</div><div>▲ initialize</div><div>● negativeVotes</div><div>● positiveVotes</div><div>question:</div><div>timestamp</div><div>timestamp:</div><div>user</div><div>user:</div><div>votes</div></div>	<div><div>addTopic:</div><div>addVote:</div><div>description</div><div>description:</div><div>▲ initialize</div><div>● negativeVotes</div><div>● positiveVotes</div><div>timestamp</div><div>timestamp:</div><div>title</div><div>title:</div><div>topics</div><div>user</div><div>user:</div><div>votes</div></div>

Move method a clase Publication:

addVote: aVote

votes add: aVote

votes

^ votes

user: anObject

user := anObject

user

^ user

timestamp: anObject

timestamp := anObject

timestamp

^ timestamp

description: anObject

description := anObject

description

^ description

negativeVotes

| r |

r := OrderedCollection new.

votes do:[:vote | vote isLike ifFalse:[r add: vote]].

^r

positiveVotes

| r |

r := OrderedCollection new.

votes do:[:vote | vote isLike ifTrue:[r add: vote]].

^r

