

PROYECTO ENTREGABLE CON JAVA: Enunciado

Autor: Jaime Hidalgo Sanchez-Noriega

En este proyecto se va a trabajar con los repositorios de código abierto más famosos del repositorio remoto Github. A través del tratamiento de esta información, contenida un fichero de datos .csv, podremos ver las tendencias actuales de los desarrolladores a la hora de elegir una tecnología para sus proyectos. De igual manera se podrá determinar la tendencias de grandes empresas, como google, facebook, por el uso de repositorios de código abierto para su desempeño en el desarrollo de las nuevas tecnologías.

Se trabajará con un archivo .CSV, el cual se compone de 8 columnas mostrando con la siguiente información: **Username, Repository Name, Description, Last Update Date, Language, Number of starts, Tags, Url**. Un extracto del fichero de datos sería el siguiente:

En este proyecto se usarán los tipos Contenedores y Factorías para recoger los datos y

```
1 username,Repository Name,Description,Last Update Date,Language,Number of Stars,Tags,Url
2 freeCodeCamp,freeCodeCamp,The https://freeCodeCamp.com open source codebase and curriculum
3 twbs,bootstrap,"The most popular HTML, CSS, and JavaScript framework for developing responsive
4 EbookFoundation,free-programming-books, Freely available programming books,2017-06-23;01
5 facebook,react,"A declarative, efficient, and flexible JavaScript library for building user
6 d3,d3,"Bring data to life with SVG, Canvas and HTML. ",2017-05-31;06:03:47,JavaScript,65
7 getify,You-Dont-Know-JS,A book series on JavaScript. @YDKJS on twitter.,2017-06-23;10:42
8 tensorflow,tensorflow,Computation using data flow graphs for scalable machine learning,2
9 sindresorhus,awesome, Curated list of awesome lists,2017-06-24;17:26:02,,60.7k,"lists,aw
10 vuejs,vue,"A progressive, incrementally-adoptable JavaScript framework for building UI o
11 angular,angular.js,AngularJS - HTML enhanced for web apps!,2017-06-24;06:25:23,JavaScript
12 robbyrussell,oh-my-zsh,"A delightful community-driven (with 1,000+ contributors) framewo
13 airbnb,javascript,JavaScript Style Guide,2017-06-24;05:44:24,JavaScript,54.1k,"arrow-fun
14 github,gitignore,A collection of useful .gitignore templates,2017-06-24;08:48:29,,51.6k
```

poder tratarlos de una manera más sencilla en el código, siendo en este caso los siguiente tipos:

Clase base: Repositorio

Tipo Contenedor: Repositorios

Factoría: FactoriaRepositorios

Para las anteriores clases definidas sus respectivas clases de testeos para comprobar que las declaraciones en el código cumplen con la función deseada.

TRATAMIENTOS SECUENCIALES:

1. **getRepositoryByLanguage():**
Método que no recibe ningún parámetro de entrada, crea un diccionario agrupando los nombres de los repositorios y obteniendo como valor el número de apariciones del mismo. Método realizado de manera tradicional.
2. **getAverageStarsbyLanguage():**
Método que no recibe ningún parámetro de entrada, devuelve un diccionario que agrupa los nombres de los lenguajes siendo esta la clave y como valor la media aritmética del número de estrellas ofrecidas por la comunidad de github.
3. **getAllTags():**
Método que no recibe ningún parámetro de entrada, devuelve una lista con todas las etiquetas diferentes que aparecen en cada uno de los repositorios del archivo csv.
4. **counterLenguajes():**
Método que recibe un parámetro de entrada de tipo cadena (tag), que hace referencia a una etiqueta, devolviendo un número entero largo con la cantidad de repositorios que contienen dicha etiqueta.
5. **haveStars():**
Método que recibe dos parámetros de entrada de tipo Integer (minValue) y tipo Language (lan) que hacen referencia a un valor mínimo de estrellas y a un lenguaje de programación específico respectivamente, devuelve una variable booleana que comprueba si todos los repositorios que filtra por lenguaje tiene un número mínimo de estrellas.
6. **existRepoTagQuantityStars():**
Método que recibe tres parámetros de entrada de tipo String (tag), de tipo Integer (quantity) y de tipo Integer (year) que hacen referencia a una etiqueta específica, una cantidad de estrellas y un años respectivamente, con ello el método devuelve una variable booleana que comprueba si existe algún repositorio que esté fechado en un determinado año, contenga una etiqueta y a su vez supere una cantidad de estrellas.
7. **existRepoTagLanguage():**
Método que recibe dos parámetros de entrada de tipo String (tag) y tipo Language (language) que hacen referencia a una etiqueta específica y a un lenguaje de programación, con ello el devolverá una variable booleana comprobando si los todos los repositorios filtrados por una etiqueta están desarrollados en un determinado lenguaje.
8. **getLanguageDistintOfTag():**
Método que recibe un parámetro de entrada de tipo String (tag) que hacer referencia a una etiqueta específica, devolviendo en el return una lista de tipo Language que contiene los lenguajes distintos filtrados por dicha etiqueta de manera ordenada,

siendo este orden el natural definido en el tipo Repositorio.

9. **getNamesByTagLanguage():**

Método que recibe tres parámetros de entrada de tipo String (tag), de tipo Language (language) y de tipo Integer (limit) que hacen referencia a una etiqueta específica, un lenguaje de programación respectivamente, dando como resultado un conjunto de tipo String que contiene los nombres de los repositorios filtrados por la etiqueta y a su vez por el nombre del lenguaje teniendo como límite un número determinado de repositorios.

10. **getRepositoryLanguageByStars():**

Método que recibe un parámetro de entrada de tipo Language (language) que hace referencia a un lenguaje de programación, dando como resultado un diccionario que agrupa por el número de estrellas y obteniendo como valor una lista de repositorios filtrados por el nombre del lenguaje aportado.

11. **getAverageStarsbyLanguageStream():**

Método que recibe no recibe ningún parámetro de entrada, que da como resultado un diccionario agrupando los repositorios por su lenguaje de programación y un valor que hace referencia a la media aritmética del número de estrellas.