

Sistema Recomendador

Generado por Doxygen 1.9.3

1 Sistema Recomendador de películas	1
1.1 Contenidos	1
1.2 Proyecto	1
1.2.1 ¿Qué es un sistema recomendador de películas?	1
1.2.2 Acerca del proyecto:	1
1.3 Modo de uso	2
1.4 Hitos	2
1.5 Autores	2
2 Indice de namespaces	3
2.1 Paquetes	3
3 Indice jerárquico	5
3.1 Jerarquía de la clase	5
4 Índice de clases	7
4.1 Lista de clases	7
5 Documentación de namespaces	9
5.1 Paquetes edu.duke	9
5.1.1 Descripción detallada	9
6 Documentación de las clases	11
6.1 Referencia de la Clase es.ull.etsii.app.AllFilters	11
6.1.1 Documentación de las funciones miembro	12
6.1.1.1 satisies()	12
6.2 Referencia de la Clase es.ull.etsii.app.DirectorsFilter	12
6.2.1 Descripción detallada	13
6.2.2 Documentación de las funciones miembro	13
6.2.2.1 satisies()	13
6.3 Referencia de la Clase edu.duke.DirectoryResource	13
6.3.1 Descripción detallada	13
6.3.2 Documentación del constructor y destructor	13
6.3.2.1 DirectoryResource()	13
6.3.3 Documentación de las funciones miembro	14
6.3.3.1 selectedFiles()	14
6.4 Referencia de la Clase es.ull.etsii.app.EfficientRater	14
6.4.1 Documentación de las funciones miembro	15
6.4.1.1 addRating()	15
6.4.1.2 getID()	15
6.4.1.3 getItemsRated()	15
6.4.1.4 getRating()	15
6.4.1.5 hasRating()	15
6.4.1.6 numRatings()	15

6.5 Referencia de la Clase edu.duke.FileResource	15
6.5.1 Descripción detallada	16
6.5.2 Documentación del constructor y destructor	16
6.5.2.1 FileResource() [1/6]	16
6.5.2.2 FileResource() [2/6]	17
6.5.2.3 FileResource() [3/6]	17
6.5.2.4 FileResource() [4/6]	17
6.5.2.5 FileResource() [5/6]	18
6.5.2.6 FileResource() [6/6]	18
6.5.3 Documentación de las funciones miembro	18
6.5.3.1 asString()	18
6.5.3.2 getCSVHeaders()	19
6.5.3.3 getCSVParser() [1/3]	19
6.5.3.4 getCSVParser() [2/3]	19
6.5.3.5 getCSVParser() [3/3]	20
6.5.3.6 lines()	20
6.5.3.7 words()	20
6.5.3.8 write() [1/4]	20
6.5.3.9 write() [2/4]	21
6.5.3.10 write() [3/4]	21
6.5.3.11 write() [4/4]	21
6.6 Referencia de la Interfaz es.ull.etsii.app.Filter	22
6.7 Referencia de la Clase es.ull.etsii.app.FirstRatings	22
6.8 Referencia de la Clase es.ull.etsii.app.FourthRatings	22
6.9 Referencia de la Clase es.ull.etsii.app.GenreFilter	23
6.9.1 Descripción detallada	23
6.9.2 Documentación de las funciones miembro	24
6.9.2.1 satisfies()	24
6.10 Referencia de la Clase edu.duke.ImageResource	24
6.10.1 Descripción detallada	24
6.10.2 Documentación del constructor y destructor	25
6.10.2.1 ImageResource() [1/5]	25
6.10.2.2 ImageResource() [2/5]	25
6.10.2.3 ImageResource() [3/5]	25
6.10.2.4 ImageResource() [4/5]	26
6.10.2.5 ImageResource() [5/5]	26
6.10.3 Documentación de las funciones miembro	26
6.10.3.1 draw()	26
6.10.3.2 getFileName()	26
6.10.3.3 getHeight()	26
6.10.3.4 getPixel()	27
6.10.3.5 getWidth()	27

6.10.3.6 pixels()	27
6.10.3.7 save()	27
6.10.3.8 saveAs()	27
6.10.3.9 setFileName()	28
6.10.3.10 setPixel()	28
6.10.3.11 toString()	28
6.11 Referencia de la Clase es.ull.etsii.app.MinutesFilter	29
6.11.1 Descripción detallada	29
6.11.2 Documentación de las funciones miembro	29
6.11.2.1 satisfies()	30
6.12 Referencia de la Clase es.ull.etsii.app.Movie	30
6.12.1 Descripción detallada	30
6.13 Referencia de la Clase es.ull.etsii.app.MovieDatabase	30
6.14 Referencia de la Clase es.ull.etsii.app.MovieRunnerAverage	31
6.15 Referencia de la Clase es.ull.etsii.app.MovieRunnerSimilarRatings	31
6.16 Referencia de la Clase es.ull.etsii.app.MovieRunnerWithFilters	31
6.17 Referencia de la Clase edu.duke.Pixel	31
6.17.1 Descripción detallada	32
6.17.2 Documentación del constructor y destructor	32
6.17.2.1 Pixel()	32
6.17.3 Documentación de las funciones miembro	32
6.17.3.1 getAlpha()	32
6.17.3.2 getBlue()	32
6.17.3.3 getGreen()	33
6.17.3.4 getRed()	33
6.17.3.5 getX()	33
6.17.3.6 getY()	33
6.17.3.7 setAlpha()	33
6.17.3.8 setBlue()	33
6.17.3.9 setGreen()	34
6.17.3.10 setRed()	34
6.17.3.11 toString()	34
6.18 Referencia de la Clase es.ull.etsii.app.PlainRater	35
6.18.1 Documentación de las funciones miembro	35
6.18.1.1 addRating()	35
6.18.1.2 getID()	36
6.18.1.3 getItemsRated()	36
6.18.1.4 getRating()	36
6.18.1.5 hasRating()	36
6.18.1.6 numRatings()	36
6.19 Referencia de la Clase edu.duke.Point	36
6.19.1 Descripción detallada	36

6.19.2 Documentación del constructor y destructor	37
6.19.2.1 Point()	37
6.19.3 Documentación de las funciones miembro	37
6.19.3.1 distance()	37
6.19.3.2 getX()	37
6.19.3.3 getY()	37
6.19.3.4 toString()	38
6.20 Referencia de la Clase edu.duke.RangeResource	38
6.20.1 Descripción detallada	38
6.20.2 Documentación del constructor y destructor	38
6.20.2.1 RangeResource() [1/4]	38
6.20.2.2 RangeResource() [2/4]	39
6.20.2.3 RangeResource() [3/4]	39
6.20.2.4 RangeResource() [4/4]	40
6.20.3 Documentación de las funciones miembro	40
6.20.3.1 sequence()	40
6.20.3.2 toString()	40
6.21 Referencia de la Interfaz es.ull.etsii.app.Rater	40
6.22 Referencia de la Clase es.ull.etsii.app.RaterDatabase	41
6.23 Referencia de la Clase es.ull.etsii.app.Rating	41
6.23.1 Descripción detallada	42
6.24 Referencia de la Clase es.ull.etsii.app.RecommendationRunner	42
6.24.1 Documentación de las funciones miembro	43
6.24.1.1 getItemsToRate()	43
6.24.1.2 printRecommendationsFor()	43
6.25 Referencia de la Interfaz es.ull.etsii.app.Recommender	44
6.25.1 Descripción detallada	44
6.25.2 Documentación de las funciones miembro	44
6.25.2.1 getItemsToRate()	44
6.25.2.2 printRecommendationsFor()	45
6.26 Referencia de la Clase edu.duke.ResourceException	45
6.26.1 Descripción detallada	46
6.27 Referencia de la Clase es.ull.etsii.app.SecondRatings	46
6.28 Referencia de la Clase edu.duke.Shape	46
6.28.1 Descripción detallada	46
6.28.2 Documentación del constructor y destructor	47
6.28.2.1 Shape() [1/2]	47
6.28.2.2 Shape() [2/2]	47
6.28.3 Documentación de las funciones miembro	47
6.28.3.1 addPoint()	47
6.28.3.2 getLastPoint()	48
6.28.3.3 getPoints()	48

6.29 Referencia de la Clase <code>es.ull.etsii.app.SimpleService</code>	48
6.30 Referencia de la Clase <code>edu.duke.StorageResource</code>	48
6.30.1 Descripción detallada	48
6.30.2 Documentación del constructor y destructor	49
6.30.2.1 <code>StorageResource()</code> [1/2]	49
6.30.2.2 <code>StorageResource()</code> [2/2]	49
6.30.3 Documentación de las funciones miembro	49
6.30.3.1 <code>add()</code>	49
6.30.3.2 <code>clear()</code>	49
6.30.3.3 <code>contains()</code>	50
6.30.3.4 <code>data()</code>	50
6.30.3.5 <code>size()</code>	50
6.31 Referencia de la Clase <code>es.ull.etsii.app.ThirdRatings</code>	50
6.32 Referencia de la Clase <code>es.ull.etsii.app.TrueFilter</code>	51
6.32.1 Descripción detallada	51
6.32.2 Documentación de las funciones miembro	51
6.32.2.1 <code>satisfies()</code>	52
6.33 Referencia de la Clase <code>edu.duke.URLResource</code>	52
6.33.1 Descripción detallada	52
6.33.2 Documentación del constructor y destructor	52
6.33.2.1 <code>URLResource()</code>	52
6.33.3 Documentación de las funciones miembro	53
6.33.3.1 <code>asString()</code>	53
6.33.3.2 <code>getCSVHeaders()</code>	53
6.33.3.3 <code>getCSVParser()</code> [1/3]	53
6.33.3.4 <code>getCSVParser()</code> [2/3]	54
6.33.3.5 <code>getCSVParser()</code> [3/3]	54
6.33.3.6 <code>lines()</code>	54
6.33.3.7 <code>words()</code>	55
6.34 Referencia de la Clase <code>es.ull.etsii.app.YearAfterFilter</code>	55
6.34.1 Descripción detallada	56
6.34.2 Documentación de las funciones miembro	56
6.34.2.1 <code>satisfies()</code>	56
Índice alfabético	57

Capítulo 1

Sistema Recomendador de películas

1.1. Contenidos

- Proyecto
 - ¿Qué es un sistema recomendador de películas?
 - Acerca del proyecto
 - Modo de uso
 - Hitos
 - Autores
-

1.2. Proyecto

1.2.1. ¿Qué es un sistema recomendador de películas?

Un sistema recomendador de películas es una aplicación que utiliza algoritmos de aprendizaje automático para predecir y recomendar películas a un usuario en función de su historial de visualización y preferencias donde se analiza el contenido de cada película (por ejemplo, el género, la trama, etc.). Los sistemas recomendadores de películas se utilizan a menudo en plataformas de streaming de video, como Netflix, HBO, Hulu, etc.. para ayudar a los usuarios a descubrir nuevas películas de interés.

1.2.2. Acerca del proyecto:

Para una documentación en profundidad acerca del proceso de trabajo del grupo en lo que respecta al proyecto en cuestión, se recomienda hacer uso del informe asociado al proyecto.

Tras la realización de diversos estudios y evaluaciones previas sobre metodologías y herramientas que permiten mejorar la calidad y el proceso de construcción de software, se realizó una propuesta para el desarrollo de un proyecto final de forma grupal, con el objetivo de reforzar y unificar los conocimientos adquiridos en la asignatura. A modo de introducción, se pueden destacar las herramientas empleadas de cara a asegurar una correcta metodología de trabajo que asegura cierta calidad en el código siguiendo con las prácticas de integración continua (CI). Estas son:

- SonarCloud
 - Maven
 - Jenkins
 - CircleCI
 - Plugins que apoyan a estas herramientas para distintos propósitos como las creación de tests, generación automática de la documentación (Doxygen), etc.
 - Otras herramientas para el desarrollo del código (IDE) como, por ejemplo, Visual Studio Code y Eclipse.
-

1.3. Modo de uso

Para poder ejecutar el proyecto, se debe tener instalado Maven y Java 8 . Para ello, se recomienda seguir los siguientes pasos:

1. Clonar el repositorio en local

```
$ git clone git@github.com:FabrizzioPerilli/Sistema-Recomendador-.git
```

2. Acceder a la raíz del proyecto

3. Compilar el proyecto

```
$ mvn compile
```

1.4. Hitos

1. Línea Base - Primer Hito realizado el 01/12/2022
 2. Línea Base - Segundo Hito realizado el 22/12/2022
-

1.5. Autores

- Carlos González Mendoza - alu0101025407@ull.edu.es
 - Fabrizzio Daniell Perilli Martín - alu0101138589@ull.edu.es
 - Francisco Javier García González - alu0101030562@ull.edu.es
-
-

Capítulo 2

Indice de namespaces

2.1. Paquetes

Aquí van los paquetes con una breve descripción (si está disponible):

edu.duke	9
------------------------------------	---

Capítulo 3

Índice jerárquico

3.1. Jerarquía de la clase

Esta lista de herencias esta ordenada aproximadamente por orden alfabético:

Comparable	
es.ull.etsii.app.Rating	41
edu.duke.DirectoryResource	13
edu.duke.FileResource	15
es.ull.etsii.app.Filter	22
es.ull.etsii.app.AllFilters	11
es.ull.etsii.app.DirectorsFilter	12
es.ull.etsii.app.GenreFilter	23
es.ull.etsii.app.MinutesFilter	29
es.ull.etsii.app.TrueFilter	51
es.ull.etsii.app.YearAfterFilter	55
es.ull.etsii.app.FirstRatings	22
es.ull.etsii.app.FourthRatings	22
edu.duke.ImageResource	24
es.ull.etsii.app.Movie	30
es.ull.etsii.app.MovieDatabase	30
es.ull.etsii.app.MovieRunnerAverage	31
es.ull.etsii.app.MovieRunnerSimilarRatings	31
es.ull.etsii.app.MovieRunnerWithFilters	31
edu.duke.Pixel	31
edu.duke.Point	36
edu.duke.RangeResource	38
es.ull.etsii.app.Rater	40
es.ull.etsii.app.EfficientRater	14
es.ull.etsii.app.PlainRater	35
es.ull.etsii.app.RaterDatabase	41
es.ull.etsii.app.Recommender	44
es.ull.etsii.app.RecommendationRunner	42
RuntimeException	
edu.duke.ResourceException	45
es.ull.etsii.app.SecondRatings	46
edu.duke.Shape	46
es.ull.etsii.app.SimpleService	48
edu.duke.StorageResource	48
es.ull.etsii.app.ThirdRatings	50
edu.duke.URLResource	52

Capítulo 4

Índice de clases

4.1. Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

es.ull.etsii.app.AllFilters	11
es.ull.etsii.app.DirectorsFilter	12
edu.duke.DirectoryResource	13
es.ull.etsii.app.EfficientRater	14
edu.duke.FileResource	15
es.ull.etsii.app.Filter	22
es.ull.etsii.app.FirstRatings	22
es.ull.etsii.app.FourthRatings	22
es.ull.etsii.app.GenreFilter	23
edu.duke.ImageResource	24
es.ull.etsii.app.MinutesFilter	29
es.ull.etsii.app.Movie	30
es.ull.etsii.app.MovieDatabase	30
es.ull.etsii.app.MovieRunnerAverage	31
es.ull.etsii.app.MovieRunnerSimilarRatings	31
es.ull.etsii.app.MovieRunnerWithFilters	31
edu.duke.Pixel	31
es.ull.etsii.app.PlainRater	35
edu.duke.Point	36
edu.duke.RangeResource	38
es.ull.etsii.app.Rater	40
es.ull.etsii.app.RaterDatabase	41
es.ull.etsii.app.Rating	41
es.ull.etsii.app.RecommendationRunner	42
es.ull.etsii.app.Recommender	44
edu.duke.ResourceException	45
es.ull.etsii.app.SecondRatings	46
edu.duke.Shape	46
es.ull.etsii.app.SimpleService	48
edu.duke.StorageResource	48
es.ull.etsii.app.ThirdRatings	50
es.ull.etsii.app.TrueFilter	51
edu.duke.URLResource	52
es.ull.etsii.app.YearAfterFilter	55

Capítulo 5

Documentación de namespaces

5.1. Paquetes edu.duke

Clases

- class [DirectoryResource](#)
- class [FileResource](#)
- class **FileSelector**
- class **ImageFrame**
- class [ImageResource](#)
- class [Pixel](#)
- class [Point](#)
- class [RangeResource](#)
- class [ResourceException](#)
- class [Shape](#)
- class [StorageResource](#)
- class **TextIterable**
- class [URLResource](#)

5.1.1. Descripción detallada

Resource classes for Duke/Coursera course.

Capítulo 6

Documentación de las clases

6.1. Referencia de la Clase es.ull.etsii.app.AllFilters

Diagrama de herencias de es.ull.etsii.app.AllFilters

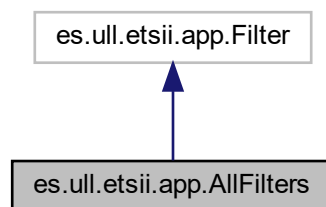
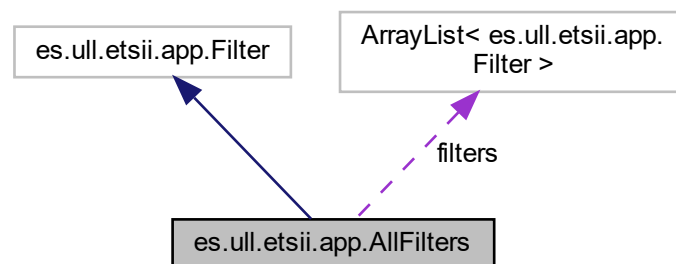


Diagrama de colaboración para es.ull.etsii.app.AllFilters:



Métodos públicos

- void **addFilter** ([Filter](#) f)
- boolean [satisfies](#) (String id)

6.1.1. Documentación de las funciones miembro

6.1.1.1. satisfies()

```
boolean es.ull.etsii.app.AllFilters.satisfies (
    String id )
```

Implementa [es.ull.etsii.app.Filter](#).

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/es/ull/etsii/app/AllFilters.java`

6.2. Referencia de la Clase es.ull.etsii.app.DirectorsFilter

Diagrama de herencias de es.ull.etsii.app.DirectorsFilter

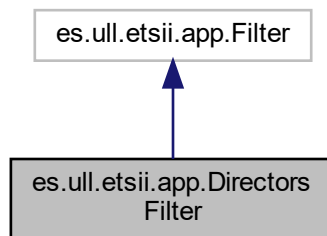
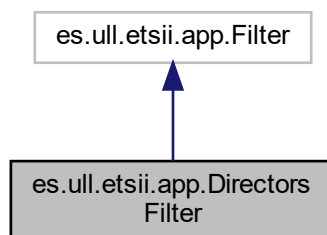


Diagrama de colaboración para es.ull.etsii.app.DirectorsFilter:



Métodos públicos

- **DirectorsFilter** (String directors)
- boolean [satisfies](#) (String id)
- String **getName** ()

6.2.1. Descripción detallada

DirectorsFilters implements [Filter](#) interface to create a `String[]` structure of directors.

Autor

Carlos González Mendoza, Fabrizio Daniell Perilli Martín, Francisco Javier García González

Versión

1.0

6.2.2. Documentación de las funciones miembro

6.2.2.1. satisfies()

```
boolean es.ull.etsii.app.DirectorsFilter.satisfies (
    String id )
```

Implementa [es.ull.etsii.app.Filter](#).

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/es/ull/etsii/app/DirectorsFilter.java`

6.3. Referencia de la Clase edu.duke.DirectoryResource

Métodos públicos

- [DirectoryResource](#) ()
- `Iterable< File > selectedFiles` ()

6.3.1. Descripción detallada

The [DirectoryResource](#) class allows the user to choose one or more files from a directory (or folder) with a file selection dialog box by using the method `selectedFiles`. These files can then be iterated over using a `for` loop.

Example usage:

```
DirectoryResource dr = new DirectoryResource();
for (File f : dr.selectedFiles()) {
    ImageResource ir = new ImageResource(f);
    ir.draw();
}
```

This software is licensed with an Apache 2 license, see <http://www.apache.org/licenses/LICENSE-2.0> for details.

Autor

Duke Software Team

6.3.2. Documentación del constructor y destructor

6.3.2.1. DirectoryResource()

```
edu.duke.DirectoryResource.DirectoryResource ( )
```

Create a [DirectoryResource](#) object.

Creating a [DirectoryResource](#) object does not open a dialog box for selecting files.

6.3.3. Documentación de las funciones miembro

6.3.3.1. selectedFiles()

```
Iterable< File > edu.duke.DirectoryResource.selectedFiles ( )
```

Open a file selection dialog box to allow the user to navigate to a directory and select one or more files from the chosen directory (or folder).

The file selection dialog box opened starts in the current project folder.

Devuelve

an `Iterable` that accesses the chosen files one at a time

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/edu/duke/DirectoryResource.java`

6.4. Referencia de la Clase `es.ull.etsii.app.EfficientRater`

Diagrama de herencias de `es.ull.etsii.app.EfficientRater`

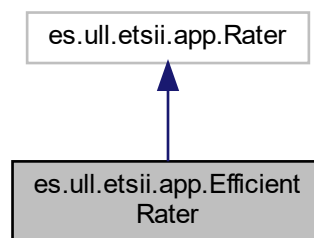
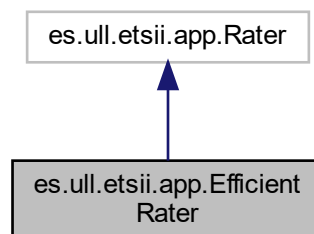


Diagrama de colaboración para `es.ull.etsii.app.EfficientRater`:



Métodos públicos

- **EfficientRater** (String id)

- void [addRating](#) (String item, double rating)
- boolean [hasRating](#) (String item)
- String [getID](#) ()
- double [getRating](#) (String item)
- int [numRatings](#) ()
- ArrayList< String > [getItemsRated](#) ()

6.4.1. Documentación de las funciones miembro

6.4.1.1. addRating()

```
void es.ull.etsii.app.EfficientRater.addRating (
    String item,
    double rating )
```

Implementa [es.ull.etsii.app.Rater](#).

6.4.1.2. getID()

```
String es.ull.etsii.app.EfficientRater.getID ( )
```

Implementa [es.ull.etsii.app.Rater](#).

6.4.1.3. getItemsRated()

```
ArrayList< String > es.ull.etsii.app.EfficientRater.getItemsRated ( )
```

Implementa [es.ull.etsii.app.Rater](#).

6.4.1.4. getRating()

```
double es.ull.etsii.app.EfficientRater.getRating (
    String item )
```

Implementa [es.ull.etsii.app.Rater](#).

6.4.1.5. hasRating()

```
boolean es.ull.etsii.app.EfficientRater.hasRating (
    String item )
```

Implementa [es.ull.etsii.app.Rater](#).

6.4.1.6. numRatings()

```
int es.ull.etsii.app.EfficientRater.numRatings ( )
```

Implementa [es.ull.etsii.app.Rater](#).

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/es/ull/etsii/app/EfficientRater.java

6.5. Referencia de la Clase edu.duke.FileResource

Métodos públicos

- [FileResource](#) ()
- [FileResource](#) (File file)

- `FileResource` (String filename)
- `FileResource` (boolean writable)
- `FileResource` (File file, boolean writable)
- `FileResource` (String filename, boolean writable)
- `Iterable< String > lines ()`
- `Iterable< String > words ()`
- `String asString ()`
- `CSVParser getCSVParser ()`
- `CSVParser getCSVParser (boolean withHeader)`
- `CSVParser getCSVParser (boolean withHeader, String delimiter)`
- `Iterable< String > getCSVHeaders (CSVParser parser)`
- `void write (String s)`
- `void write (StorageResource list)`
- `void write (String[] list)`
- `void write (ArrayList< String > list)`

6.5.1. Descripción detallada

The `FileResource` class represents a file and allows access to its contents a line at a time, using the method `lines`, or a word at a time, using the method `words`. These strings can then be iterated over using a `for` loop. Example usage:

```
FileResource fr = new FileResource();
for (String s : fr.words()) {
    // print or process s
}
```

If each line of the file represents separated data values, because its a CSV file, then the user can get a `getCSVParser` object to access that data more directly, using one of the `getCSVParser` methods.

Example CSV usage:

```
FileResource fr = new FileResource("food.csv");
for (CSVRecord record : fr.getCSVParser()) {
    // print or process fields in record
    String name = record.get("Name");
    // other processing
}
```

This software is licensed with an Apache 2 license, see <http://www.apache.org/licenses/LICENSE-2.0> for details.

Autor

Duke Software Team

6.5.2. Documentación del constructor y destructor

6.5.2.1. `FileResource()` [1/6]

`edu.duke.FileResource.FileResource ()`

Create a `FileResource` object that opens the file chosen by the user using a file selection dialog box.

Excepciones

<i>exception</i>	if no file is selected by the user
------------------	------------------------------------

6.5.2.2. FileResource() [2/6]

```
edu.duke.FileResource.FileResource (
    File file )
```

Create a [FileResource](#) object that opens a file represented by the File object passed as a parameter. Useful, for example, when used in conjunction with the [DirectoryResource](#) class.

Parámetros

<i>file</i>	the file to be represented by this resource
-------------	---

Excepciones

<i>exception</i>	if the file cannot be accessed
------------------	--------------------------------

6.5.2.3. FileResource() [3/6]

```
edu.duke.FileResource.FileResource (
    String filename )
```

Create a [FileResource](#) object that opens a file whose name is passed as a parameter. The named file should be on the current class path to be found.

Parámetros

<i>filename</i>	the name of the file to be opened
-----------------	-----------------------------------

Excepciones

<i>exception</i>	if the filename cannot be accessed
------------------	------------------------------------

6.5.2.4. FileResource() [4/6]

```
edu.duke.FileResource.FileResource (
    boolean writable )
```

Create a [FileResource](#) object that opens the file chosen by the user using a file selection dialog box, possibly to write to it.

If the user wants to change the contents of the open file by using the method `write` to add new strings to it, pass `true` as the second parameter. Otherwise it is assumed the user will only iterate over the existing contents of the file.

Parámetros

<i>writable</i>	allow changes to this file only if true
-----------------	---

Excepciones

<i>exception</i>	if no file is selected by the user
------------------	------------------------------------

6.5.2.5. FileResource() [5/6]

```
edu.duke.FileResource.FileResource (
    File file,
    boolean writable )
```

Create a [FileResource](#) object that opens a file represented by the `File` object passed as a parameter, possibly to write to it.

If the user wants to change the contents of the open file by using the method `write` to add new strings to it, pass `true` as the second parameter. Otherwise it is assumed the user will only iterate over the existing contents of the file.

Useful, for example, when used in conjunction with the [DirectoryResource](#) class.

Parámetros

<i>file</i>	the file to be represented by this resource
<i>writable</i>	allow changes to this file only if true

Excepciones

<i>exception</i>	if the file cannot be accessed
------------------	--------------------------------

6.5.2.6. FileResource() [6/6]

```
edu.duke.FileResource.FileResource (
    String filename,
    boolean writable )
```

Create a [FileResource](#) object that opens a file whose name is passed as a parameter, possibly to write to it.

If the user wants to change the contents of the open file by using the method `write` to add new strings to it, pass `true` as the second parameter. Otherwise it is assumed the user will only iterate over the existing contents of the file.

The named file should be on the current class path to be found.

Parámetros

<i>filename</i>	the name of the file to be opened
<i>writable</i>	allow changes to this file only if true

Excepciones

<i>exception</i>	if the filename cannot be accessed
------------------	------------------------------------

6.5.3. Documentación de las funciones miembro

6.5.3.1. asString()

```
String edu.duke.FileResource.asString ( )
```

Return entire contents of this opened file as one string.

Devuelve

a `String` that is the contents of the open file

6.5.3.2. getCSVHeaders()

```
Iterable< String > edu.duke.FileResource.getCSVHeaders (
    CSVParser parser )
```

Allows access to the column names of the header row of a CSV file (the first line in the file) one at a time. If the CSV file did not have a header row, then an empty `Iterator` is returned.

Parámetros

<i>parser</i>	the <code>CSVParser</code> that has been created for this file
---------------	--

Devuelve

an `Iterable` that allows access one header name at a time

6.5.3.3. getCSVParser() [1/3]

```
CSVParser edu.duke.FileResource.getCSVParser ( )
```

Returns a `CSVParser` object to access the contents of an open file.

Each line of the file should be formatted as data separated by commas and with a header row to describe the column names.

Devuelve

a `CSVParser` that can provide access to the records in the file one at a time

Excepciones

<i>exception</i>	if this file does not represent a CSV formatted data
------------------	--

6.5.3.4. getCSVParser() [2/3]

```
CSVParser edu.duke.FileResource.getCSVParser (
    boolean withHeader )
```

Returns a `CSVParser` object to access the contents of an open file, possibly without a header row.

Each line of the file should be formatted as data separated by commas and with/without a header row to describe the column names.

Parámetros

<i>withHeader</i>	uses first row of data as a header row only if true
-------------------	---

Devuelve

a `CSVParser` that can provide access to the records in the file one at a time

Excepciones

<i>exception</i>	if this file does not represent a CSV formatted data
------------------	--

6.5.3.5. getCSVParser() [3/3]

```
CSVParser edu.duke.FileResource.getCSVParser (
    boolean withHeader,
    String delimiter )
```

Returns a `CSVParser` object to access the contents of an open file, possibly without a header row and a different data delimiter than a comma.

Each line of the file should be formatted as data separated by the delimiter passed as a parameter and with/without a header row to describe the column names. This is useful if the data is separated by some character other than a comma.

Parámetros

<i>withHeader</i>	uses first row of data as a header row only if true
<i>delimiter</i>	a single character that separates one field of data from another

Devuelve

a `CSVParser` that can provide access to the records in the file one at a time

Excepciones

<i>exception</i>	if this file does not represent a CSV formatted data
<i>exception</i>	if <code>delimiter.length() != 1</code>

6.5.3.6. lines()

```
Iterable< String > edu.duke.FileResource.lines ( )
```

Allow access to this opened file one line at a time.

Devuelve

an `Iterable` that will allow access to contents of opened file one line at a time.

6.5.3.7. words()

```
Iterable< String > edu.duke.FileResource.words ( )
```

Allow access to this opened file one word at a time, where words are separated by white-space. This means any form of spaces, like tabs or newlines, can delimit words.

Devuelve

an `Iterable` that will allow access to contents of opened file one word at a time.

6.5.3.8. write() [1/4]

```
void edu.duke.FileResource.write (
    ArrayList< String > list )
```

Writes a list of strings to the end of this file, one element per line.

Parámetros

<i>list</i>	the strings to saved to the file
-------------	----------------------------------

6.5.3.9. write() [2/4]

```
void edu.duke.FileResource.write (
    StorageResource list )
```

Writes a list of strings to the end of this file, one element per line.

Parámetros

<i>list</i>	the strings to saved to the file
-------------	----------------------------------

6.5.3.10. write() [3/4]

```
void edu.duke.FileResource.write (
    String s )
```

Writes a string to the end of this file.

Parámetros

<i>s</i>	the string to saved to the file
----------	---------------------------------

6.5.3.11. write() [4/4]

```
void edu.duke.FileResource.write (
    String[] list )
```

Writes a list of strings to the end of this file, one element per line.

Parámetros

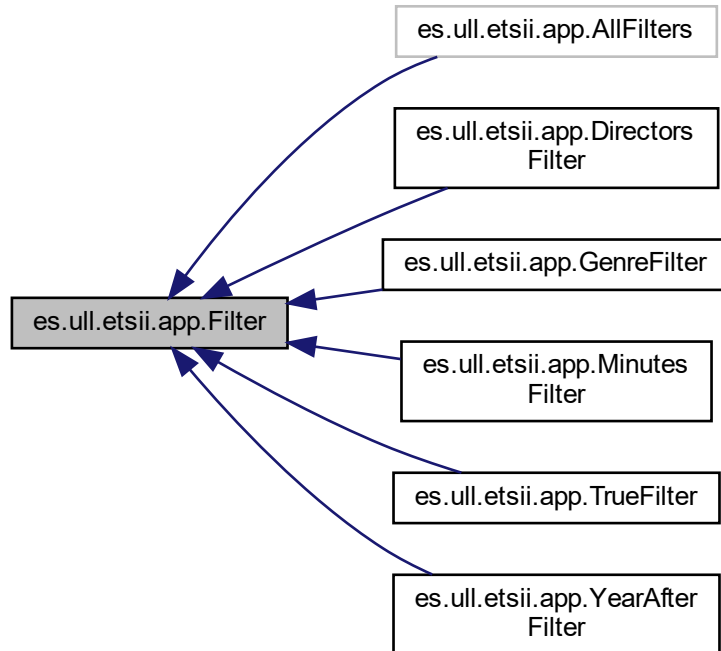
<i>list</i>	the strings to saved to the file
-------------	----------------------------------

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/duke/FileResource.java

6.6. Referencia de la Interfaz es.ull.etsii.app.Filter

Diagrama de herencias de es.ull.etsii.app.Filter



Métodos públicos

- boolean **satisfies** (String id)

La documentación para este interfaz fue generada a partir del siguiente fichero:

- src/main/java/es/ull/etsii/app/Filter.java

6.7. Referencia de la Clase es.ull.etsii.app.FirstRatings

Métodos públicos

- ArrayList< [Movie](#) > **loadMovies** (String filename)
- void **testLoadMovies** ()
- ArrayList< [EfficientRater](#) > **loadRaters** (String filename)
- ArrayList< String > **getRaterWithMostRatings** (ArrayList< [EfficientRater](#) > raters)
- void **testLoadRaters** ()

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/es/ull/etsii/app/FirstRatings.java

6.8. Referencia de la Clase es.ull.etsii.app.FourthRatings

Métodos públicos

- ArrayList< [Rating](#) > **getAverageRatings** (int minimalRaters)

- ArrayList< [Rating](#) > **getAverageRatingsByFilter** (int minimalRaters, [Filter](#) filterCriteria)
- double **dotProduct** ([Rater](#) me, [Rater](#) r)
- ArrayList< [Rating](#) > **getSimilarRatings** (String id, int numSimilarRaters, int minimalRaters)
- ArrayList< [Rating](#) > **getSimilarRatingsByFilter** (String id, int numSimilarRaters, int minimalRaters, [Filter](#) filterCriteria)

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/es/ull/etsii/app/FourthRatings.java

6.9. Referencia de la Clase es.ull.etsii.app.GenreFilter

Diagrama de herencias de es.ull.etsii.app.GenreFilter

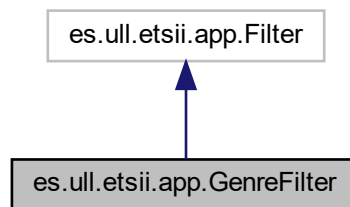
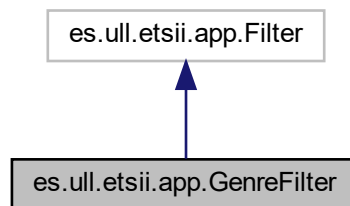


Diagrama de colaboración para es.ull.etsii.app.GenreFilter:



Métodos públicos

- **GenreFilter** (String withGenre)
- boolean [satisfies](#) (String id)
- String **getName** ()

6.9.1. Descripción detallada

[GenreFilter](#) implements [Filter](#) interface to return true if a movie satisfies this genre (movies may have several genres).

Autor

Carlos González Mendoza, Fabrizio Daniell Perilli Martín, Francisco Javier García González

Versión

1.0

6.9.2. Documentación de las funciones miembro

6.9.2.1. satisfies()

```
boolean es.ull.etsii.app.GenreFilter.satisfies (
    String id )
```

Implementa [es.ull.etsii.app.Filter](#).

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/es/ull/etsii/app/GenreFilter.java`

6.10. Referencia de la Clase edu.duke.ImageResource

Métodos públicos

- [ImageResource](#) ()
- [ImageResource](#) (int width, int height)
- [ImageResource](#) (String fileName)
- [ImageResource](#) (File file)
- [ImageResource](#) ([ImageResource](#) other)
- int [getWidth](#) ()
- int [getHeight](#) ()
- [Iterable](#)< [Pixel](#) > [pixels](#) ()
- void [draw](#) ()
- String [getFileName](#) ()
- void [setFileName](#) (String name)
- [Pixel](#) [getPixel](#) (int x, int y)
- void [setPixel](#) (int x, int y, [Pixel](#) p)
- String [toString](#) ()
- void [save](#) ()
- void [saveAs](#) ()

6.10.1. Descripción detallada

The [ImageResource](#) class represents an image as a grid of [Pixel](#) objects and allows access to all of them, using the method `pixels`. These pixels can then be iterated over using a `for` loop.

Note, no changes made to the pixels in this image affect the original image opened unless you use the method `save`.

Example usage:

```
ImageResource image = new ImageResource();
for (Pixel p : image.pixels()) {
    int red = p.getRed();
    int green = p.getGreen();
    int blue = p.getBlue();
    int average = (red + green + blue) / 3;
    p.setRed(average);
    p.setGreen(average);
    p.setBlue(average);
}
image.draw();
```


This is open-source software released under the terms of the GPL (<http://www.gnu.org/licenses/gpl.html>).

6.10.2. Documentación del constructor y destructor

6.10.2.1. ImageResource() [1/5]

```
edu.duke.ImageResource.ImageResource ( )
```

Create an `ImageResource` object that represents the file chosen by the user using a file selection dialog box.

Excepciones

<i>exception</i>	if no file is selected by the user
------------------	------------------------------------

6.10.2.2. ImageResource() [2/5]

```
edu.duke.ImageResource.ImageResource (
    int width,
    int height )
```

Create an `ImageResource` object whose size is the width and height passed as parameters and whose pixels are all black.

Parámetros

<i>width</i>	the width of the image in pixels
<i>height</i>	the height of the image in pixels

Excepciones

<i>exception</i>	if the width or height are not positive values
------------------	--

6.10.2.3. ImageResource() [3/5]

```
edu.duke.ImageResource.ImageResource (
    String fileName )
```

Create an `ImageResource` object from the file name passed as a parameter.

Parámetros

<i>fileName</i>	the name of the file
-----------------	----------------------

Excepciones

<i>exception</i>	if the file cannot be accessed or is not in an image format
------------------	---

6.10.2.4. `ImageResource()` [4/5]

```
edu.duke.ImageResource.ImageResource (
    File file )
```

Create an `ImageResource` object from a file given as a parameter.

Parámetros

<i>file</i>	the file representing an image
-------------	--------------------------------

Excepciones

<i>exception</i>	if the file cannot be accessed or is not in an image format
------------------	---

6.10.2.5. `ImageResource()` [5/5]

```
edu.duke.ImageResource.ImageResource (
    ImageResource other )
```

Create an `ImageResource` object that is a copy of another image.

Parámetros

<i>other</i>	the original image being copied
--------------	---------------------------------

6.10.3. Documentación de las funciones miembro

6.10.3.1. `draw()`

```
void edu.duke.ImageResource.draw ( )
```

Displays this image in a separate window.

6.10.3.2. `getFileName()`

```
String edu.duke.ImageResource.getFileName ( )
```

Returns the file name associated with this image.

Devuelve

the name of the file used to create this image or an empty string if it was created as a sized image

6.10.3.3. `getHeight()`

```
int edu.duke.ImageResource.getHeight ( )
```

Returns the height of the image in pixels.

Devuelve

the image's height in pixels

6.10.3.4. getPixel()

```
Pixel edu.duke.ImageResource.getPixel (
    int x,
    int y )
```

Returns the pixel at the (x, y) coordinates passed as a parameter.

Parámetros

<i>x</i>	the column position of the pixel
<i>y</i>	the row position of the pixel

Devuelve

the [Pixel](#) at the given (x, y) coordinates

6.10.3.5. getWidth()

```
int edu.duke.ImageResource.getWidth ( )
```

Returns the width of the image in pixels.

Devuelve

the image's width in pixels

6.10.3.6. pixels()

```
Iterable< Pixel > edu.duke.ImageResource.pixels ( )
```

Allow access to this image one pixel at a time.

Devuelve

an `Iterable` that will allow access to each pixel in this image

6.10.3.7. save()

```
void edu.duke.ImageResource.save ( )
```

Saves the image as a JPEG using its current file name or opens a file selection dialog box to allow the user to choose a name if no file name set (for example if this image was created as a blank sized image).

Excepciones

<i>exception</i>	if the current filename cannot be accessed for saving
------------------	---

6.10.3.8. saveAs()

```
void edu.duke.ImageResource.saveAs ( )
```

Saves the image as a JPEG by opening a file selection dialog box to allow the user to choose the new name for the file.

Excepciones

<i>exception</i>	if no file is selected by the user
------------------	------------------------------------

6.10.3.9. `setFileName()`

```
void edu.duke.ImageResource.setFileName (
    String name )
```

Resets the file name associated with this image.

Useful, for example, when saving the results of changes to this image in a different file than the original.

Parámetros

<i>name</i>	the new name for the file
-------------	---------------------------

6.10.3.10. `setPixel()`

```
void edu.duke.ImageResource.setPixel (
    int x,
    int y,
    Pixel p )
```

Resets the pixel at the given (x, y) coordinates but does not redraw it.

Parámetros

<i>x</i>	the column position of the pixel
<i>y</i>	the row position of the pixel
<i>p</i>	the new pixel values to use

6.10.3.11. `toString()`

```
String edu.duke.ImageResource.toString ( )
```

Returns a string representation of the image (file name, width, and height).

Devuelve

a string representation of the image

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/edu/duke/ImageResource.java`

6.11. Referencia de la Clase es.ull.etsii.app.MinutesFilter

Diagrama de herencias de es.ull.etsii.app.MinutesFilter

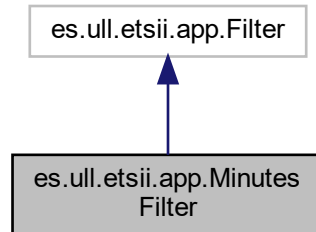
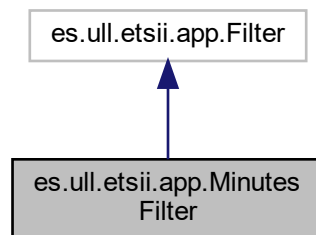


Diagrama de colaboración para es.ull.etsii.app.MinutesFilter:



Métodos públicos

- **MinutesFilter** (int minMinutes, int maxMinutes)
- boolean [satisfies](#) (String id)
- String [getName](#) ()

6.11.1. Descripción detallada

[MinutesFilter](#) implements filter interface to return true if a movie's running time is at least min minutes and no more than max minutes.

Autor

Carlos González Mendoza, Fabrizio Daniell Perilli Martín, Francisco Javier García González

Versión

1.0

6.11.2. Documentación de las funciones miembro

6.11.2.1. satisfies()

```
boolean es.ull.etsii.app.MinutesFilter.satisfies (  
    String id )
```

Implementa [es.ull.etsii.app.Filter](#).

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/es/ull/etsii/app/MinutesFilter.java`

6.12. Referencia de la Clase `es.ull.etsii.app.Movie`

Métodos públicos

- **Movie** (String anID, String aTitle, String aYear, String theGenres)
- **Movie** (String anID, String aTitle, String aYear, String theGenres, String aDirector, String aCountry, String aPoster, int theMinutes)
- String **getID** ()
- String **getTitle** ()
- int **getYear** ()
- String **getGenres** ()
- String **getCountry** ()
- String **getDirector** ()
- String **getPoster** ()
- int **getMinutes** ()
- String **toString** ()

6.12.1. Descripción detallada

[Movie](#) represents movies and all their attributes and methods.

Autor

Carlos González Mendoza, Fabrizio Daniell Perilli Martín, Francisco Javier García González

Versión

1.0

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/es/ull/etsii/app/Movie.java`

6.13. Referencia de la Clase `es.ull.etsii.app.MovieDatabase`

Métodos públicos estáticos

- static void **initialize** (String moviefile)
- static boolean **containsID** (String id)
- static int **getYear** (String id)
- static String **getGenres** (String id)
- static String **getTitle** (String id)
- static [Movie](#) **getMovie** (String id)
- static String **getPoster** (String id)
- static int **getMinutes** (String id)
- static String **getCountry** (String id)
- static String **getDirector** (String id)
- static int **size** ()
- static ArrayList< String > **filterBy** ([Filter](#) f)

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/es/ull/etsii/app/MovieDatabase.java`

6.14. Referencia de la Clase es.ull.etsii.app.MovieRunnerAverage

Métodos públicos

- void **printAverageRatings** ()
- void **getAverageRatingOneMovie** ()

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/es/ull/etsii/app/MovieRunnerAverage.java

6.15. Referencia de la Clase es.ull.etsii.app.MovieRunnerSimilarRatings

Métodos públicos

- void **printAverageRating** ()
- void **printAverageRatingByYearAfterAndGenre** ()
- void **printSimilarRatings** ()
- void **printSimilarRatingsByGenre** ()
- void **printSimilarRatingsByDirector** ()
- void **printSimilarRatingsByGenreAndMinutes** ()
- void **printSimilarRatingsByYearAfterAndMinutes** ()

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/es/ull/etsii/app/MovieRunnerSimilarRatings.java

6.16. Referencia de la Clase es.ull.etsii.app.MovieRunnerWithFilters

Métodos públicos

- void **printAverageRating** ()
- void **printAverageRatingByYear** ()
- void **printAverageRatingByGenre** ()
- void **printAverageRatingByMinutes** ()
- void **printAverageRatingByDirector** ()
- void **printAverageRatingByYearAfterAndGenre** ()
- void **printAverageRatingsByDirectorsAndMinutes** ()

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/es/ull/etsii/app/MovieRunnerWithFilters.java

6.17. Referencia de la Clase edu.duke.Pixel

Métodos públicos

- [Pixel](#) ([Pixel](#) other)
- int [getX](#) ()
- int [getY](#) ()
- int [getRed](#) ()
- int [getGreen](#) ()
- int [getBlue](#) ()
- int [getAlpha](#) ()
- void [setRed](#) (int r)
- void [setGreen](#) (int g)
- void [setBlue](#) (int b)
- void [setAlpha](#) (int a)
- String [toString](#) ()

6.17.1. Descripción detallada

The `Pixel` class represents a color as its component values of red, green, blue, as well as alpha (for transparency). Each of the component values of a pixel must have a value between 0 and 255. If a value is given outside that range, it is changed to be within that range. As such, a negative value would be set to 0 and a value greater than 255 would be set to 255.

Example usage:

```
ImageResource image = new ImageResource();
for (Pixel p : image.pixels()) {
    int red = p.getRed();
    int green = p.getGreen();
    int blue = p.getBlue();
    int average = (red + green + blue) / 3;
    p.setRed(average);
    p.setGreen(average);
    p.setBlue(average);
}
image.draw();
```

This is open-source software released under the terms of the GPL (<http://www.gnu.org/licenses/gpl.html>).

6.17.2. Documentación del constructor y destructor

6.17.2.1. Pixel()

```
edu.duke.Pixel.Pixel (
    Pixel other )
```

Creates a new `Pixel` from with the same values as the other pixel passed as a parameter.

Parámetros

<i>other</i>	another pixel
--------------	---------------

6.17.3. Documentación de las funciones miembro

6.17.3.1. getAlpha()

```
int edu.duke.Pixel.getAlpha ( )
```

Returns the value of the pixel's alpha (or transparency) component.

Devuelve

the pixel's alpha value within the range 0-255

6.17.3.2. getBlue()

```
int edu.duke.Pixel.getBlue ( )
```

Returns the value of the pixel's blue component.

Devuelve

the pixel's blue value within the range 0-255

6.17.3.3. getGreen()

```
int edu.duke.Pixel.getGreen ( )
```

Returns the value of the pixel's green component.

Devuelve

the pixel's green value within the range 0-255

6.17.3.4. getRed()

```
int edu.duke.Pixel.getRed ( )
```

Returns the value of the pixel's red component.

Devuelve

the pixel's red value within the range 0-255

6.17.3.5. getX()

```
int edu.duke.Pixel.getX ( )
```

Returns the pixel's x-coordinate within the image.

Devuelve

the x-coordinate of this pixel.

6.17.3.6. getY()

```
int edu.duke.Pixel.getY ( )
```

Returns the pixel's y-coordinate within the image.

Devuelve

the y-coordinate of this pixel.

6.17.3.7. setAlpha()

```
void edu.duke.Pixel.setAlpha (
    int a )
```

Resets the value of the pixel's alpha (or transparency) component to the value passed as a parameter. If it is not in the range of 0-255 it is changed to be in that range.

Parámetros

<i>a</i>	the alpha value
----------	-----------------

6.17.3.8. setBlue()

```
void edu.duke.Pixel.setBlue (
    int b )
```

Resets the value of the pixel's blue component to the value passed as a parameter. If it is not in the range of 0-255 it is changed to be in that range.

Parámetros

<i>b</i>	the blue value
----------	----------------

6.17.3.9. setGreen()

```
void edu.duke.Pixel.setGreen (
    int g )
```

Resets the value of the pixel's green component to the value passed as a parameter. If it is not in the range of 0-255 it is changed to be in that range.

Parámetros

<i>g</i>	the green value
----------	-----------------

6.17.3.10. setRed()

```
void edu.duke.Pixel.setRed (
    int r )
```

Resets the value of the pixel's red component to the value passed as a parameter. If it is not in the range of 0-255 it is changed to be in that range.

Parámetros

<i>r</i>	the red value
----------	---------------

6.17.3.11. toString()

```
String edu.duke.Pixel.toString ( )
```

Returns the string representation of the of the pixel.

Devuelve

a string containing the RGB values

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/duke/Pixel.java

6.18. Referencia de la Clase es.ull.etsii.app.PlainRater

Diagrama de herencias de es.ull.etsii.app.PlainRater

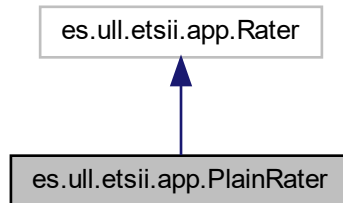
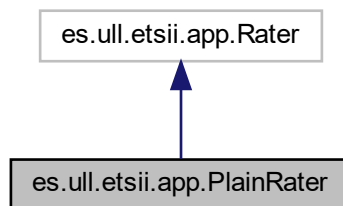


Diagrama de colaboración para es.ull.etsii.app.PlainRater:



Métodos públicos

- **PlainRater** (String id)
- void [addRating](#) (String item, double rating)
- boolean [hasRating](#) (String item)
- String [getID](#) ()
- double [getRating](#) (String item)
- int [numRatings](#) ()
- ArrayList< String > [getItemsRated](#) ()

6.18.1. Documentación de las funciones miembro

6.18.1.1. addRating()

```
void es.ull.etsii.app.PlainRater.addRating (
    String item,
    double rating )
```

Implementa [es.ull.etsii.app.Rater](#).

6.18.1.2. `getID()`

`String es.ull.etsii.app.PlainRater.getID ()`
Implementa [es.ull.etsii.app.Rater](#).

6.18.1.3. `getItemsRated()`

`ArrayList< String > es.ull.etsii.app.PlainRater.getItemsRated ()`
Implementa [es.ull.etsii.app.Rater](#).

6.18.1.4. `getRating()`

`double es.ull.etsii.app.PlainRater.getRating (`
 `String item)`
Implementa [es.ull.etsii.app.Rater](#).

6.18.1.5. `hasRating()`

`boolean es.ull.etsii.app.PlainRater.hasRating (`
 `String item)`
Implementa [es.ull.etsii.app.Rater](#).

6.18.1.6. `numRatings()`

`int es.ull.etsii.app.PlainRater.numRatings ()`
Implementa [es.ull.etsii.app.Rater](#).

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/es/ull/etsii/app/PlainRater.java`

6.19. Referencia de la Clase `edu.duke.Point`

Métodos públicos

- [Point](#) (int startx, int starty)
- int [getX](#) ()
- int [getY](#) ()
- double [distance](#) ([Point](#) otherPt)
- String [toString](#) ()

Métodos públicos estáticos

- static void [main](#) (String[] args)

6.19.1. Descripción detallada

The [Point](#) class represents a two-dimensional location, constructed from (x,y) coordinates with some methods for access and the capability to calculate the distance from this point to another point.

Example usage:

```
Point a = new Point(3, 4);
Point b = new Point(2, 5);
double dist = a.distance(b);
```

This software is licensed with an Apache 2 license, see <http://www.apache.org/licenses/LICENSE-2.0> for details.

Autor

Duke Coursera Team

6.19.2. Documentación del constructor y destructor**6.19.2.1. Point()**

```
edu.duke.Point.Point (
    int startx,
    int starty )
```

Create a `Point` object from x and y coordinates.**Parámetros**

<i>startx</i>	is the x-coordinate
<i>starty</i>	is the y-coordinate

6.19.3. Documentación de las funciones miembro**6.19.3.1. distance()**

```
double edu.duke.Point.distance (
    Point otherPt )
```

Calculate and return the Euclidean distance from this point to another point.

Parámetros

<i>otherPt</i>	the other point to which distance is calculated
----------------	---

Devuelve

the distance from this point to otherPt

6.19.3.2. getX()

```
int edu.duke.Point.getX ( )
```

Returns the x coordinate of this point.

Devuelve

x coordinate

6.19.3.3. getY()

```
int edu.duke.Point.getY ( )
```

Returns the y coordinate of this point.

Devuelve

y coordinate

6.19.3.4. toString()

`String edu.duke.Point.toString ()`

Returns a string representation of this point.

Devuelve

(x,y) for this point.

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/edu/duke/Point.java`

6.20. Referencia de la Clase `edu.duke.RangeResource`

Métodos públicos

- `RangeResource` (int end)
- `RangeResource` (int start, int end)
- `RangeResource` (int start, int end, int increment)
- `RangeResource` (`RangeResource` other)
- `String toString ()`
- `Iterable< Integer > sequence ()`

6.20.1. Descripción detallada

The `RangeResource` class represents a range of integer (or whole) numbers and allows access to these numbers using the method `sequence`. The integers can then be iterated over using a `for` loop

Example usage:

```
// this prints the square number sequence for the first 9 numbers:
// 1, 4, 9, 16, 25, 36, 49, 64, 81
RangeResource rr = new RangeResource(1, 10);
for (int value : rr.sequence()) {
    System.out.println(value * value);
}
```

Another example usage:

```
// this prints the odd numbers in decreasing order within the given range:
// 37, 35, 33, 31, 29, 27, 25, 23
RangeResource rr = new RangeResource(37, 21, -2);
for (int value : rr.sequence()) {
    System.out.println(value);
}
```

This software is licensed with an Apache 2 license, see <http://www.apache.org/licenses/LICENSE-2.0> for details.

Autor

Duke Software Team

6.20.2. Documentación del constructor y destructor

6.20.2.1. `RangeResource()` [1/4]

`edu.duke.RangeResource.RangeResource (`
 int end)

Create a `RangeResource` object, starting at 0 and going up to but not including end, [0 - end), that increments by 1.

Parámetros

<i>end</i>	when to stop the range, not included as one of the values
------------	---

Excepciones

<i>exception</i>	if the end is negative
------------------	------------------------

6.20.2.2. RangeResource() [2/4]

```
edu.duke.RangeResource.RangeResource (
    int start,
    int end )
```

Create a [RangeResource](#) object, starting at start and going up to but not including end, [start - end), that increments by 1.

Parámetros

<i>start</i>	the first value in the range, included as one of the values
<i>end</i>	when to stop the range, not included as one of the values

Excepciones

<i>exception</i>	if the end is less than the start
------------------	-----------------------------------

6.20.2.3. RangeResource() [3/4]

```
edu.duke.RangeResource.RangeResource (
    int start,
    int end,
    int increment )
```

Create a [RangeResource](#) object, starting at start and going up to but not including end, [start - end), that increments by the amount passed as a parameter.

Parámetros

<i>start</i>	the first value in the range, included as one of the values
<i>end</i>	when to stop the range, not included as one of the values
<i>increment</i>	how much to add to get the next value in the range's sequence

Excepciones

<i>exception</i>	if increment is negative when the end is greater than the start
<i>exception</i>	if increment is positive when the end is less than the start
<i>exception</i>	if increment is 0

6.20.2.4. RangeResource() [4/4]

```
edu.duke.RangeResource.RangeResource (
    RangeResource other )
```

Create an `RangeResource` object that is a copy of another range.

Parámetros

<code>other</code>	the original range being copied
--------------------	---------------------------------

6.20.3. Documentación de las funciones miembro

6.20.3.1. sequence()

```
Iterable< Integer > edu.duke.RangeResource.sequence ( )
```

Allow access to the numbers in this range one at a time.

Devuelve

an `Iterable` that will allow access to each number in this range

6.20.3.2. toString()

```
String edu.duke.RangeResource.toString ( )
```

Return string representation of this range, with each value in the sequence separated by a comma.

Devuelve

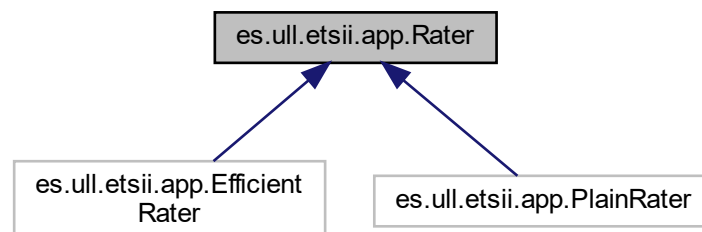
a `String` containing all the values in the range

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/edu/duke/RangeResource.java`

6.21. Referencia de la Interfaz `es.ull.etsii.app.Rater`

Diagrama de herencias de `es.ull.etsii.app.Rater`



Métodos públicos

- void **addRating** (String item, double rating)
- boolean **hasRating** (String item)
- String **getID** ()
- double **getRating** (String item)
- int **numRatings** ()
- ArrayList< String > **getItemsRated** ()

La documentación para este interfaz fue generada a partir del siguiente fichero:

- src/main/java/es/ull/etsii/app/Rater.java

6.22. Referencia de la Clase es.ull.etsii.app.RaterDatabase

Métodos públicos estáticos

- static void **initialize** (String filename)
- static void **addRatings** (String filename)
- static void **addRaterRating** (String raterID, String movieID, double rating)
- static **Rater** **getRater** (String id)
- static ArrayList< **Rater** > **getRaters** ()
- static int **size** ()

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/es/ull/etsii/app/RaterDatabase.java

6.23. Referencia de la Clase es.ull.etsii.app.Rating

Diagrama de herencias de es.ull.etsii.app.Rating

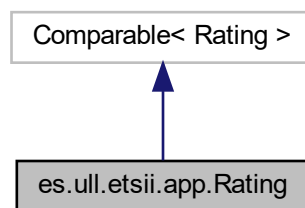
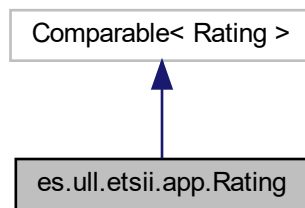


Diagrama de colaboración para `es.ull.etsii.app.Rating`:



Métodos públicos

- **Rating** (String anItem, double aValue)
- String **getItem** ()
- double **getValue** ()
- String **toString** ()
- int **compareTo** ([Rating](#) other)

6.23.1. Descripción detallada

[Rating](#) is also a POJO class for storing the data about one rating of an item.

Autor

Carlos González Mendoza, Fabrizio Daniell Perilli Martín, Francisco Javier García González

Versión

1.0

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/es/ull/etsii/app/Rating.java`

6.24. Referencia de la Clase `es.ull.etsii.app.RecommendationRunner`

Diagrama de herencias de `es.ull.etsii.app.RecommendationRunner`

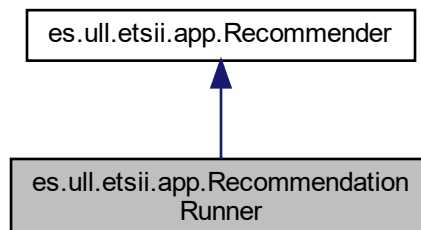
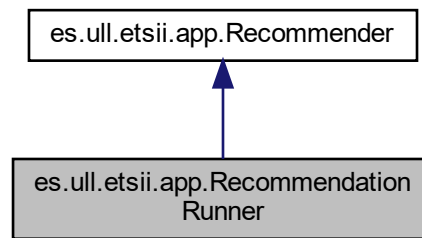


Diagrama de colaboración para es.ull.etsii.app.RecommendationRunner:



Métodos públicos

- `ArrayList< String > getItemsToRate ()`
- `void printRecommendationsFor (String webRaterID)`

6.24.1. Documentación de las funciones miembro

6.24.1.1. `getItemsToRate()`

```
ArrayList< String > es.ull.etsii.app.RecommendationRunner.getItemsToRate ( )
```

This method returns a list of movie IDs that will be used to look up the movies in the [MovieDatabase](#) and present them to users to rate.

The movies returned in the list will be displayed on a web page, so the number you choose may affect how long the page takes to load and how willing users are to rate the movies. For example, 10-20 should be fine, 50 or more would be too many.

There are no restrictions on the method you use to generate this list of movies: the most recent movies, movies from a specific genre, randomly chosen movies, or simply your favorite movies.

The ratings for these movies will make the profile for a new [Rater](#) that will be used to compare to for finding recommendations.

Implementa [es.ull.etsii.app.Recommender](#).

6.24.1.2. `printRecommendationsFor()`

```
void es.ull.etsii.app.RecommendationRunner.printRecommendationsFor (
    String webRaterID )
```

This method returns nothing, but prints out an HTML table of the movies recommended for the given rater.

The HTML printed will be displayed on a web page, so the number you choose to display may affect how long the page takes to load. For example, you may want to limit the number printed to only the top 20-50 movies recommended or to movies not rated by the given rater.

You may also include CSS styling for your table using the `<style>` tag before you print the table. There are no restrictions on which movies you print, what order you print them in, or what information you include about each movie.

Parámetros

<i>webRaterID</i>	the ID of a new Rater that has been already added to the RaterDatabase with ratings for the movies returned by the method <code>getItemsToRate</code>
-------------------	---

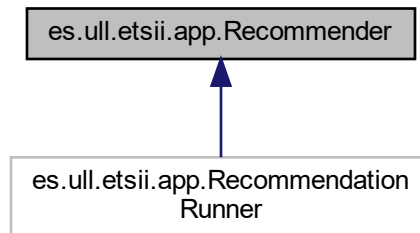
Implementa [es.ull.etsii.app.Recommender](#).

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/es/ull/etsii/app/RecommendationRunner.java`

6.25. Referencia de la Interfaz `es.ull.etsii.app.Recommender`

Diagrama de herencias de `es.ull.etsii.app.Recommender`



Métodos públicos

- `ArrayList< String > getItemsToRate ()`
- `void printRecommendationsFor (String webRaterID)`

6.25.1. Descripción detallada

Implement this interface to allow your code to be integrated with our web site.

When users first visit the recommender website, our code will call the method `getItemsToRate ()` to get a list of movies to display on the web page for users to rate.

When a user submits their ratings, our code will call the method `printRecommendationsFor` to get your recommendations based on the user's ratings. The ID given to this method is for a new [Rater](#) that we have already added to the [RaterDatabase](#) with ratings for the movies returned by the first method. Whatever is printed from that method will be displayed on the web page: HTML, plain text, or debugging information.

6.25.2. Documentación de las funciones miembro

6.25.2.1. `getItemsToRate()`

```
ArrayList< String > es.ull.etsii.app.Recommender.getItemsToRate ( )
```

This method returns a list of movie IDs that will be used to look up the movies in the [MovieDatabase](#) and present them to users to rate.

The movies returned in the list will be displayed on a web page, so the number you choose may affect how long the page takes to load and how willing users are to rate the movies. For example, 10-20 should be fine, 50 or more would be too many.

There are no restrictions on the method you use to generate this list of movies: the most recent movies, movies from a specific genre, randomly chosen movies, or simply your favorite movies.

The ratings for these movies will make the profile for a new [Rater](#) that will be used to compare to for finding recommendations.

Implementado en [es.ull.etsii.app.RecommendationRunner](#).

6.25.2.2. printRecommendationsFor()

```
void es.ull.etsii.app.Recommender.printRecommendationsFor (
    String webRaterID )
```

This method returns nothing, but prints out an HTML table of the movies recommended for the given rater.

The HTML printed will be displayed on a web page, so the number you choose to display may affect how long the page takes to load. For example, you may want to limit the number printed to only the top 20-50 movies recommended or to movies not rated by the given rater.

You may also include CSS styling for your table using the `<style>` tag before you print the table. There are no restrictions on which movies you print, what order you print them in, or what information you include about each movie.

Parámetros

<i>webRaterID</i>	the ID of a new Rater that has been already added to the RaterDatabase with ratings for the movies returned by the method <code>getItemsToRate</code>
-------------------	---

Implementado en [es.ull.etsii.app.RecommendationRunner](#).

La documentación para este interfaz fue generada a partir del siguiente fichero:

- `src/main/java/es/ull/etsii/app/Recommender.java`

6.26. Referencia de la Clase edu.duke.ResourceException

Diagrama de herencias de edu.duke.ResourceException

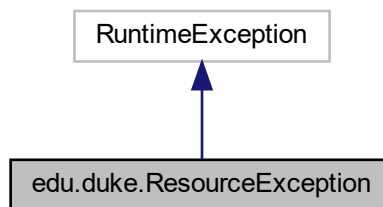
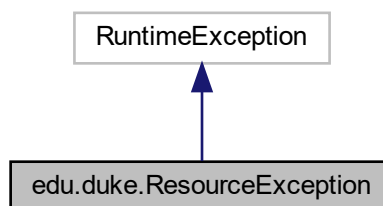


Diagrama de colaboración para edu.duke.ResourceException:



Métodos públicos

- **ResourceException** (String message)
- **ResourceException** (String message, Throwable cause)

6.26.1. Descripción detallada

A general exception for Resource objects.

Note, this is a RuntimeException so that it does not HAVE to be caught, but still gives useful information about the error.

This software is licensed with an Apache 2 license, see <http://www.apache.org/licenses/LICENSE-2.0> for details.

Autor

Duke Software Team

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/edu/duke/ResourceException.java

6.27. Referencia de la Clase es.ull.etsii.app.SecondRatings

Métodos públicos

- **SecondRatings** (String moviefile, String ratingsfile)
- int **getMovieSize** ()
- int **getRaterSize** ()
- double **getAverageByID** (String id, int minimalRaters)
- ArrayList< [Rating](#) > **getAverageRatings** (int minimalRaters)
- String **getTitle** (String id)
- String **getID** (String title)

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/es/ull/etsii/app/SecondRatings.java

6.28. Referencia de la Clase edu.duke.Shape

Métodos públicos

- [Shape](#) ()
- [Shape](#) ([FileResource](#) file)
- void [addPoint](#) ([Point](#) p)
- [Point](#) [getLastPoint](#) ()
- Iterable< [Point](#) > [getPoints](#) ()

6.28.1. Descripción detallada

The [Shape](#) class represents a collection of [Points](#) that can easily model a polygonal shape. Points can be added in order via code or given using a file, one point per line separated by a comma. The points can then be accessed in the order they were added one at a time, using the method [getPoints](#).

Example usage:

```
Point a = new Point(3, 4);
Point b = new Point(2, 5);
Shape s = new Shape();
s.addPoint(a);
s.addPoint(b);
s.addPoint(new Point(1, 1));
```

```
for (Point p : s.getPoints()) {
    System.out.println(p);
}
```

A `Shape` can also be created using file in the following format:

```
3, 4
2, 5
1, 1
```

Example file usage:

```
FileResource fr = new FileResource();
Shape s = new Shape(fr);
for (Point p : s.getPoints()) {
    System.out.println(p);
}
```

This software is licensed with an Apache 2 license, see <http://www.apache.org/licenses/LICENSE-2.0> for details.

Autor

Duke Coursera Team

6.28.2. Documentación del constructor y destructor

6.28.2.1. Shape() [1/2]

```
edu.duke.Shape.Shape ( )
```

Create an empty `Shape` object, one with no points.

6.28.2.2. Shape() [2/2]

```
edu.duke.Shape.Shape (
    FileResource file )
```

Create a `Shape` object from a file with x,y coordinates of points, in order, one pair of x,y per line. Each x,y are comma-separated, e.g.,

```
3, 4
-2, 5
```

whitespace on line is skipped.

Parámetros

<i>file</i>	is the <code>FileResource</code> accessible and bound to a file with one pair of points per line
-------------	--

6.28.3. Documentación de las funciones miembro

6.28.3.1. addPoint()

```
void edu.duke.Shape.addPoint (
    Point p )
```

Add a point to this shape/polygon.

The order in which points are added defines the order in which points are accessed.

Parámetros

<i>p</i>	is the Point added to this shape
----------	--

6.28.3.2. getLastPoint()

```
Point edu.duke.Shape.getLastPoint ( )
```

Return the last point added to this shape/polygon

Devuelve

the last [Point](#) added

6.28.3.3. getPoints()

```
Iterable< Point > edu.duke.Shape.getPoints ( )
```

Allow access to this shape one point at a time.

Devuelve

an Iterable that allows access to points

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/edu/duke/Shape.java`

6.29. Referencia de la Clase es.ull.etsii.app.SimpleService**Métodos públicos**

- String **helloService** (String msg)

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/es/ull/etsii/app/SimpleService.java`

6.30. Referencia de la Clase edu.duke.StorageResource**Métodos públicos**

- [StorageResource](#) ()
- [StorageResource](#) ([StorageResource](#) other)
- void [clear](#) ()
- void [add](#) (String s)
- int [size](#) ()
- boolean [contains](#) (String s)
- Iterable< String > [data](#) ()

6.30.1. Descripción detallada

The [StorageResource](#) class stores any number of `String` objects and allows access to these stored values one at a time, using the method `data`. These strings can then be iterated over in the order they were added using a `for` loop.

This class mirrors an `ArrayList<String>` in some functionality, but is simpler to use and fits the Duke/Coursersa model of creating and using iterables.

Example usage:


```
FileResource fr = new FileResource();
StorageResource store = new StorageResource();
for (String s : fr.words()) {
    store.add(s);
}
// can process store here, e.g.,
// get number of strings stored
int x = store.size();
for (String s : store.data()) {
    // print or process s
}
```

This software is licensed with an Apache 2 license, see <http://www.apache.org/licenses/LICENSE-2.0> for details.

Autor

Duke Software Team

6.30.2. Documentación del constructor y destructor

6.30.2.1. StorageResource() [1/2]

```
edu.duke.StorageResource.StorageResource ( )
```

Create an empty `StorageResource` object

6.30.2.2. StorageResource() [2/2]

```
edu.duke.StorageResource.StorageResource (
    StorageResource other )
```

Create an `StorageResource` object that is a copy of another list.

Parámetros

<i>other</i>	the original list being copied
--------------	--------------------------------

6.30.3. Documentación de las funciones miembro

6.30.3.1. add()

```
void edu.duke.StorageResource.add (
    String s )
```

Adds a string to this storage object.

Parámetros

<i>s</i>	the value added
----------	-----------------

6.30.3.2. clear()

```
void edu.duke.StorageResource.clear ( )
```

Remove all strings from this object so that `.size() == 0`.

6.30.3.3. contains()

```
boolean edu.duke.StorageResource.contains (
    String s )
```

Determines if a string is stored in this object.

Parámetros

s	string searched for
---	---------------------

Devuelve

true if and only if s is stored in this object

6.30.3.4. data()

```
Iterable< String > edu.duke.StorageResource.data ( )
```

Create and return an iterable for all strings in this object.

Devuelve

an `Iterable` that allows access to each string in the order stored

6.30.3.5. size()

```
int edu.duke.StorageResource.size ( )
```

Returns the number of strings added/stored in this object.

Devuelve

the number of strings stored in the object

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/edu/duke/StorageResource.java`

6.31. Referencia de la Clase `es.ull.etsii.app.ThirdRatings`**Métodos públicos**

- **ThirdRatings** (String ratingsfile)
- int **getRaterSize** ()
- ArrayList< [Rating](#) > **getAverageRatings** (int minimalRaters)
- ArrayList< [Rating](#) > **getAverageRatingsByFilter** (int minimalRaters, [Filter](#) filterCriteria)

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/es/ull/etsii/app/ThirdRatings.java`

6.32. Referencia de la Clase es.ull.etsii.app.TrueFilter

Diagrama de herencias de es.ull.etsii.app.TrueFilter

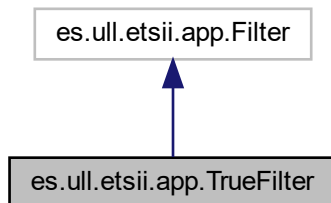
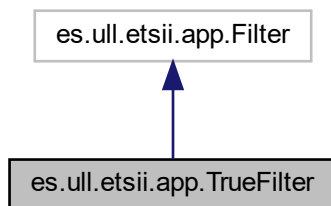


Diagrama de colaboración para es.ull.etsii.app.TrueFilter:



Métodos públicos

- boolean [satisfies](#) (String id)

6.32.1. Descripción detallada

[TrueFilter](#) implements [Filter](#) interface to select every movie from [MovieDatabase](#). It's satisfies method always returns true.

Autor

Carlos González Mendoza, Fabrizio Daniell Perilli Martín, Francisco Javier García González

Versión

1.0

6.32.2. Documentación de las funciones miembro

6.32.2.1. satisfies()

```
boolean es.ull.etsii.app.TrueFilter.satisfies (
    String id )
```

Implementa [es.ull.etsii.app.Filter](#).

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/es/ull/etsii/app/TrueFilter.java`

6.33. Referencia de la Clase edu.duke.URLResource

Métodos públicos

- [URLResource](#) (String name)
- `Iterable< String > lines ()`
- `Iterable< String > words ()`
- `String asString ()`
- `CSVParser getCSVParser ()`
- `CSVParser getCSVParser (boolean withHeader)`
- `CSVParser getCSVParser (boolean withHeader, String delimiter)`
- `Iterable< String > getCSVHeaders (CSVParser parser)`

6.33.1. Descripción detallada

The [URLResource](#) class opens a connection to a URL and allows access to the contents of the web page a line at a time, using the method `lines`, or a word at a time, using the method `words`. These strings can then be iterated over using a `for` loop.

Example usage:

```
URLResource ur = new URLResource("http://www.dukelearntoprogram.com/");
for (String s : ur.lines()) {
    // print or process s
}
```

If each line of the web page represents separated data values, because its a CSV file, then the user can get a `getCSVParser` object to access that data more directly, using one of the `getCSVParser` methods.

Example CSV usage:

```
URLResource ur = new URLResource("http://www.dukelearntoprogram.com/course2/java/food.csv");
for (CSVRecord record : ur.getCSVParser()) {
    // print or process fields in record
    String name = record.get("Name");
    // other processing
}
```

This software is licensed with an Apache 2 license, see <http://www.apache.org/licenses/LICENSE-2.0> for details.

Autor

Duke Software Team

6.33.2. Documentación del constructor y destructor

6.33.2.1. URLResource()

```
edu.duke.URLResource.URLResource (
    String name )
```

Create a [URLResource](#) object bound to the web page whose URL is given as the parameter.

Constructing the object opens a connection and reads the contents of the web page.

Parámetros

<i>name</i>	is the name of the URL, it must start with "http" or "https"
-------------	--

Excepciones

<i>exception</i>	if the URL does not start with "http" or "https"
------------------	--

6.33.3. Documentación de las funciones miembro

6.33.3.1. asString()

```
String edu.duke.URLResource.asString ( )
```

Return entire open web page as one string.

Devuelve

a `String` that is the contents of the open web page

6.33.3.2. getCSVHeaders()

```
Iterable< String > edu.duke.URLResource.getCSVHeaders (
    CSVParser parser )
```

Allows access to the column names of the header row of a CSV file (the first line in the file) one at a time. If the CSV file did not have a header row, then an empty `Iterator` is returned.

Parámetros

<i>parser</i>	the <code>CSVParser</code> that has been created for this web page
---------------	--

Devuelve

an `Iterable` that allows access one header name at a time

6.33.3.3. getCSVParser() [1/3]

```
CSVParser edu.duke.URLResource.getCSVParser ( )
```

Returns a `CSVParser` object to access the contents of an open web page.

Each line of the web page should be formatted as data separated by commas and with a header row to describe the column names.

Devuelve

a `CSVParser` that can provide access to the records in the web page one at a time

Excepciones

<i>exception</i>	if this web page does not represent a CSV formatted data
------------------	--

6.33.3.4. getCSVParser() [2/3]

```
CSVParser edu.duke.URLResource.getCSVParser (
    boolean withHeader )
```

Returns a `CSVParser` object to access the contents of an open web page, possibly without a header row. Each line of the web page should be formatted as data separated by commas and with/without a header row to describe the column names.

Parámetros

<i>withHeader</i>	uses first row of data as a header row only if true
-------------------	---

Devuelve

a `CSVParser` that can provide access to the records in the web page one at a time

Excepciones

<i>exception</i>	if this web page does not represent a CSV formatted data
------------------	--

6.33.3.5. getCSVParser() [3/3]

```
CSVParser edu.duke.URLResource.getCSVParser (
    boolean withHeader,
    String delimiter )
```

Returns a `CSVParser` object to access the contents of an open web page, possibly without a header row and a different data delimiter than a comma.

Each line of the web page should be formatted as data separated by the delimiter passed as a parameter and with/without a header row to describe the column names. This is useful if the data is separated by some character other than a comma.

Parámetros

<i>withHeader</i>	uses first row of data as a header row only if true
<i>delimiter</i>	a single character that separates one field of data from another

Devuelve

a `CSVParser` that can provide access to the records in the web page one at a time

Excepciones

<i>exception</i>	if this web page does not represent a CSV formatted data
<i>exception</i>	if <code>delimiter.length() != 1</code>

6.33.3.6. lines()

```
Iterable< String > edu.duke.URLResource.lines ( )
```

Allow access to open web page one line at a time.

Devuelve

an `Iterable` that allows access one line at a time

6.33.3.7. words()

```
Iterable< String > edu.duke.URLResource.words ( )
```

Allow access to this open web page one word at a time, where words are separated by white-space. This means any form of spaces, like tabs or newlines, can delimit words.

Devuelve

an `Iterable` that allows access one word at a time

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/edu/duke/URLResource.java`

6.34. Referencia de la Clase es.ull.etsii.app.YearAfterFilter

Diagrama de herencias de es.ull.etsii.app.YearAfterFilter

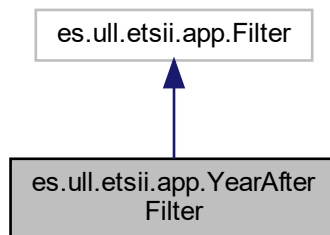
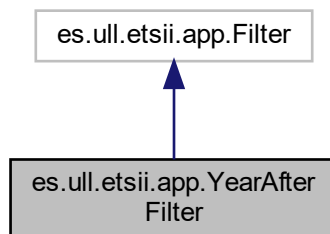


Diagrama de colaboración para es.ull.etsii.app.YearAfterFilter:

**Métodos públicos**

- **YearAfterFilter** (int year)
- boolean [satisfies](#) (String id)

6.34.1. Descripción detallada

[YearAfterFilter](#) implements [Filter](#) interface to calculate the number of movies in the database that have at least a minimal number of ratings and came out in a particular year or later.

Autor

Carlos González Mendoza, Fabrizio Daniell Perilli Martín, Francisco Javier García González

Versión

1.0

6.34.2. Documentación de las funciones miembro

6.34.2.1. satisfies()

```
boolean es.ull.etsii.app.YearAfterFilter.satisfies (  
    String id )
```

Implementa [es.ull.etsii.app.Filter](#).

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/es/ull/etsii/app/YearAfterFilter.java`

Índice alfabético

- add
 - edu.duke.StorageResource, 49
- addPoint
 - edu.duke.Shape, 47
- addRating
 - es.ull.etsii.app.EfficientRater, 15
 - es.ull.etsii.app.PlainRater, 35
- asString
 - edu.duke.FileResource, 18
 - edu.duke.URLResource, 53
- clear
 - edu.duke.StorageResource, 49
- contains
 - edu.duke.StorageResource, 49
- data
 - edu.duke.StorageResource, 50
- DirectoryResource
 - edu.duke.DirectoryResource, 13
- distance
 - edu.duke.Point, 37
- draw
 - edu.duke.ImageResource, 26
- edu.duke, 9
- edu.duke.DirectoryResource, 13
 - DirectoryResource, 13
 - selectedFiles, 14
- edu.duke.FileResource, 15
 - asString, 18
 - FileResource, 16–18
 - getCSVHeaders, 19
 - getCSVParser, 19, 20
 - lines, 20
 - words, 20
 - write, 20, 21
- edu.duke.ImageResource, 24
 - draw, 26
 - getFileName, 26
 - getHeight, 26
 - getPixel, 26
 - getWidth, 27
 - ImageResource, 25, 26
 - pixels, 27
 - save, 27
 - saveAs, 27
 - setFileName, 28
 - setPixel, 28
 - toString, 28
- edu.duke.Pixel, 31
 - getAlpha, 32
 - getBlue, 32
 - getGreen, 32
 - getRed, 33
 - getX, 33
 - getY, 33
 - Pixel, 32
 - setAlpha, 33
 - setBlue, 33
 - setGreen, 34
 - setRed, 34
 - toString, 34
- edu.duke.Point, 36
 - distance, 37
 - getX, 37
 - getY, 37
 - Point, 37
 - toString, 37
- edu.duke.RangeResource, 38
 - RangeResource, 38, 39
 - sequence, 40
 - toString, 40
- edu.duke.ResourceException, 45
- edu.duke.Shape, 46
 - addPoint, 47
 - getLastPoint, 48
 - getPoints, 48
 - Shape, 47
- edu.duke.StorageResource, 48
 - add, 49
 - clear, 49
 - contains, 49
 - data, 50
 - size, 50
 - StorageResource, 49
- edu.duke.URLResource, 52
 - asString, 53
 - getCSVHeaders, 53
 - getCSVParser, 53, 54
 - lines, 54
 - URLResource, 52
 - words, 55
- es.ull.etsii.app.AllFilters, 11
 - satisfies, 12
- es.ull.etsii.app.DirectorsFilter, 12
 - satisfies, 13
- es.ull.etsii.app.EfficientRater, 14
 - addRating, 15

- getID, 15
 - getItemsRated, 15
 - getRating, 15
 - hasRating, 15
 - numRatings, 15
 - es.ull.etsii.app.Filter, 22
 - es.ull.etsii.app.FirstRatings, 22
 - es.ull.etsii.app.FourthRatings, 22
 - es.ull.etsii.app.GenreFilter, 23
 - satisfies, 24
 - es.ull.etsii.app.MinutesFilter, 29
 - satisfies, 29
 - es.ull.etsii.app.Movie, 30
 - es.ull.etsii.app.MovieDatabase, 30
 - es.ull.etsii.app.MovieRunnerAverage, 31
 - es.ull.etsii.app.MovieRunnerSimilarRatings, 31
 - es.ull.etsii.app.MovieRunnerWithFilters, 31
 - es.ull.etsii.app.PlainRater, 35
 - addRating, 35
 - getID, 35
 - getItemsRated, 36
 - getRating, 36
 - hasRating, 36
 - numRatings, 36
 - es.ull.etsii.app.Rater, 40
 - es.ull.etsii.app.RaterDatabase, 41
 - es.ull.etsii.app.Rating, 41
 - es.ull.etsii.app.RecommendationRunner, 42
 - getItemsToRate, 43
 - printRecommendationsFor, 43
 - es.ull.etsii.app.Recommender, 44
 - getItemsToRate, 44
 - printRecommendationsFor, 44
 - es.ull.etsii.app.SecondRatings, 46
 - es.ull.etsii.app.SimpleService, 48
 - es.ull.etsii.app.ThirdRatings, 50
 - es.ull.etsii.app.TrueFilter, 51
 - satisfies, 51
 - es.ull.etsii.app.YearAfterFilter, 55
 - satisfies, 56
- FileResource
- edu.duke.FileResource, 16–18
- getAlpha
- edu.duke.Pixel, 32
- getBlue
- edu.duke.Pixel, 32
- getCSVHeaders
- edu.duke.FileResource, 19
 - edu.duke.URLResource, 53
- getCSVParser
- edu.duke.FileResource, 19, 20
 - edu.duke.URLResource, 53, 54
- getFileName
- edu.duke.ImageResource, 26
- getGreen
- edu.duke.Pixel, 32
- getHeight
- edu.duke.ImageResource, 26
- getID
- es.ull.etsii.app.EfficientRater, 15
 - es.ull.etsii.app.PlainRater, 35
- getItemsRated
- es.ull.etsii.app.EfficientRater, 15
 - es.ull.etsii.app.PlainRater, 36
- getItemsToRate
- es.ull.etsii.app.RecommendationRunner, 43
 - es.ull.etsii.app.Recommender, 44
- getLastPoint
- edu.duke.Shape, 48
- getPixel
- edu.duke.ImageResource, 26
- getPoints
- edu.duke.Shape, 48
- getRating
- es.ull.etsii.app.EfficientRater, 15
 - es.ull.etsii.app.PlainRater, 36
- getRed
- edu.duke.Pixel, 33
- getWidth
- edu.duke.ImageResource, 27
- getX
- edu.duke.Pixel, 33
 - edu.duke.Point, 37
- getY
- edu.duke.Pixel, 33
 - edu.duke.Point, 37
- hasRating
- es.ull.etsii.app.EfficientRater, 15
 - es.ull.etsii.app.PlainRater, 36
- ImageResource
- edu.duke.ImageResource, 25, 26
- lines
- edu.duke.FileResource, 20
 - edu.duke.URLResource, 54
- numRatings
- es.ull.etsii.app.EfficientRater, 15
 - es.ull.etsii.app.PlainRater, 36
- Pixel
- edu.duke.Pixel, 32
- pixels
- edu.duke.ImageResource, 27
- Point
- edu.duke.Point, 37
- printRecommendationsFor
- es.ull.etsii.app.RecommendationRunner, 43
 - es.ull.etsii.app.Recommender, 44
- RangeResource
- edu.duke.RangeResource, 38, 39
- satisfies
- es.ull.etsii.app.AllFilters, 12

- es.ull.etsii.app.DirectorsFilter, [13](#)
- es.ull.etsii.app.GenreFilter, [24](#)
- es.ull.etsii.app.MinutesFilter, [29](#)
- es.ull.etsii.app.TrueFilter, [51](#)
- es.ull.etsii.app.YearAfterFilter, [56](#)
- save
 - edu.duke.ImageResource, [27](#)
- saveAs
 - edu.duke.ImageResource, [27](#)
- selectedFiles
 - edu.duke.DirectoryResource, [14](#)
- sequence
 - edu.duke.RangeResource, [40](#)
- setAlpha
 - edu.duke.Pixel, [33](#)
- setBlue
 - edu.duke.Pixel, [33](#)
- setFileName
 - edu.duke.ImageResource, [28](#)
- setGreen
 - edu.duke.Pixel, [34](#)
- setPixel
 - edu.duke.ImageResource, [28](#)
- setRed
 - edu.duke.Pixel, [34](#)
- Shape
 - edu.duke.Shape, [47](#)
- size
 - edu.duke.StorageResource, [50](#)
- StorageResource
 - edu.duke.StorageResource, [49](#)
- toString
 - edu.duke.ImageResource, [28](#)
 - edu.duke.Pixel, [34](#)
 - edu.duke.Point, [37](#)
 - edu.duke.RangeResource, [40](#)
- URLResource
 - edu.duke.URLResource, [52](#)
- words
 - edu.duke.FileResource, [20](#)
 - edu.duke.URLResource, [55](#)
- write
 - edu.duke.FileResource, [20](#), [21](#)