

# Juegos de prueba

PROP Grupo 33  
V́ctor Muńoz, Alex Moa, Artur Farriols, Ḿlanie Scat́ena  
Cuatrimestre otońo  
Curso 21-22

# Introducción

Este documento incluye todos los juegos de prueba requeridos para comprobar el correcto funcionamiento de las principales funcionalidades del sistema recomendador.

## Registrar usuario

### Objetivo

Garantizar el debido funcionamiento del caso de uso *Register*.

### Elementos integrados en la prueba

#### Capa de dominio:

- CtrlDomain.java
- CtrlUsers.java
- CtrlDataFactory.java
- User.java

#### Capa de datos:

- CtrlData.java
- DataUtils.java
- Utils.java

#### Capa de presentación:

- LoginUserView.java
- RegisterView.java
- MainView.java
- ProfileView.java

### Ficheros empleados

- *Dataset* de películas.

### Estado del sistema tras la ejecución de la prueba

Tras la ejecución de la prueba, el sistema añadirá un usuario cuyo identificador se corresponderá con el indicado como parámetro.

### Acciones a realizar

Al ejecutar el programa, deberá escogerse el *Dataset* llamado *movies* y pulsar el icono de *Load*. Cuando el usuario se encuentre en la vista *LoginUserView*, deberá pulsar el botón *Register*. Una vez acceda a la vista *RegisterView*, deberá introducir

un identificador válido (proporcionado en el apartado *Input*) y hacer click nuevamente en el botón mencionado anteriormente. Finalmente, tras ser redirigido a la vista *MainView*, simplemente tendrá que acceder a su perfil para comprobar que la información que ha proporcionado ha sido almacenada correctamente por el sistema.

## Input

- Identificador de usuario a introducir: a.

## Output esperado

- Identificador mostrado en el perfil: a.

## Login

### Objetivo

Garantizar el debido funcionamiento del caso de uso *Login*.

## Elementos integrados en la prueba

### Capa de dominio:

- *CtrlDomain.java*
- *CtrlUsers.java*
- *CtrlDataFactory.java*
- *User.java*

### Capa de datos:

- *CtrlData.java*
- *DataUtils.java*
- *Utils.java*

### Capa de presentación:

- *DatasetsView.java*
- *LoginUserView.java*
- *MainView.java*
- *ProfileView.java*

## Ficheros empleados

- *Dataset* de películas.

## Estado del sistema tras la ejecución de la prueba

El estado del sistema no se ve alterado tras la ejecución del juego de prueba.

## Acciones a realizar

Al ejecutar el programa, deberá escogerse el *Dataset* llamado *movies* y pulsar el icono de *Load*. Una vez el usuario se encuentre en la vista *LoginUserView*, es necesario que el usuario introduzca un identificador válido (proporcionado en el apartado *Input*). Finalmente, tras ser redirigido a la vista *MainView*, será preciso que acceda a su perfil y verifique que el identificador mostrado se corresponde con el introducido.

## Input

- Identificador de usuario a introducir: 117838.

## Output esperado

- Identificador mostrado en el perfil: 117838.

## Logout

## Objetivo

Garantizar el debido funcionamiento del caso de uso *Logout*.

## Elementos integrados en la prueba

### Capa de dominio:

- CtrlDomain.java
- CtrlUsers.java
- CtrlDataFactory.java
- User.java

### Capa de datos:

- CtrlData.java
- DataUtils.java
- Utils.java

### Capa de presentación:

- DatasetsView.java
- LoginUserView.java
- MainView.java

## Ficheros empleados

- *Dataset* de películas.

## Estado del sistema tras la ejecución de la prueba

El estado del sistema no se ve alterado tras la ejecución del juego de prueba.

## Acciones a realizar

Al ejecutar el programa, deberá escogerse el *Dataset* llamado *movies* y pulsar el icono de *Load*. Una vez el usuario se encuentre en la vista *LoginUserView*, es necesario que se introduzca un identificador válido (proporcionado en el apartado *Input*). Finalmente, tras hacer click en el icono situado en la parte superior izquierda de la vista, se redirigirá al usuario a la vista *LoginUserView*.

## Input

- Identificador de usuario a introducir: 117838.

## Output esperado

Tras la ejecución del juego de prueba, el sistema no proporcionará *feedback*.

# Collaborative Filtering

## Objetivo

Garantizar el debido funcionamiento del algoritmo *Collaborative Filtering*.

## Elementos integrados en la prueba

### Capa de dominio:

- *CtrlDomain.java*
- *CtrlUsers.java*
- *CtrlDataFactory.java*
- *User.java*
- *Item.java*
- *Algorithm.java*
- *Recommendation.java*
- *NewCollaborativeFiltering.java*

### Capa de datos:

- *CtrlData.java*

- DataUtils.java
- Utils.java

#### **Capa de presentación:**

- DatasetsView.java
- LoginUserView.java
- MainView.java
- RecommendationsView.java

### **Ficheros empleados**

- *Dataset* de películas.

### **Estado del sistema tras la ejecución de la prueba**

El sistema vinculará la recomendación realizada al usuario.

### **Acciones a realizar**

Al ejecutar el programa, deberá escogerse el *Dataset* llamado *movies* y pulsar el icono de *Load*. Una vez el usuario se encuentre en la vista *LoginUserView*, es necesario que se introduzca un identificador válido (proporcionado en el apartado *Input*). Tras ser redirigido a la vista *MainView*, será preciso que escoja el algoritmo a testear y seleccione la precisión (ambos valores están definidos en el apartado *Input*). Solo restará hacer click en el botón *Recommend* y, posteriormente, *See more*. Al cargarse la vista *RecommendationsView*, podrá verificarse que la recomendación ha sido creada y asociada al perfil.

### **Input**

- Identificador de usuario a introducir: 117838.
- Algoritmo a escoger: *CollaborativeFiltering*.
- Precisión a escoger: *Imprecise*.

### **Output esperado**

- Identificador de la recomendación.
- Título del ítem.
- Valoración estimada por el algoritmo.
- Puntuación que el usuario asigna a la recomendación (en este caso 0).

# Content-Based Filtering

## Objetivo

Garantizar el debido funcionamiento del algoritmo *Content-Based Filtering*.

## Elementos integrados en la prueba

### Capa de dominio:

- CtrlDomain.java
- CtrlUsers.java
- CtrlDataFactory.java
- User.java
- Item.java
- Algorithm.java
- Recommendation.java
- ContentBasedFiltering.java

### Capa de datos:

- CtrlData.java
- DataUtils.java
- Utils.java

### Capa de presentación:

- DatasetsView.java
- LoginUserView.java
- MainView.java
- RecommendationsView.java

## Ficheros empleados

- *Dataset* de películas.

## Estado del sistema tras la ejecución de la prueba

El sistema vinculará la recomendación realizada al usuario.

## Acciones a realizar

Al ejecutar el programa, deberá escogerse el *Dataset* llamado *movies* y pulsar el icono de *Load*. Una vez el usuario se encuentre en la vista *LoginUserView*, es necesario que se introduzca un identificador válido (proporcionado en el apartado *Input*). Tras ser redirigido a la vista *MainView*, será preciso que escoja el algoritmo a testear y seleccione la precisión (ambos valores están definidos en el apartado *Input*). Solo restará hacer click en el botón *Recommend* y, posteriormente, *See*

*more*. Al cargar la vista *RecommendationsView*, podrá verificarse que la recomendación ha sido creada y asociada al perfil.

## Input

- Identificador de usuario a introducir: 117838.
- Algoritmo a escoger: *ContentBasedFiltering*.
- Precisión a escoger: *Imprecise*.

## Output esperado

- Identificador de la recomendación.
- Título del ítem.
- Valoración estimada por el algoritmo.
- Puntuación que el usuario asigna a la recomendación (en este caso 0).

# Hybrid Approach Filtering

## Objetivo

Garantizar el debido funcionamiento del algoritmo *Hybrid Approach Filtering*.

## Elementos integrados en la prueba

### Capa de dominio:

- CtrlDomain.java
- CtrlUsers.java
- CtrlDataFactory.java
- User.java
- Item.java
- Algorithm.java
- Recommendation.java
- ContentBasedFiltering.java

### Capa de datos:

- CtrlData.java
- DataUtils.java
- Utils.java

### Capa de presentación:

- DatasetsView.java
- LoginUserView.java
- MainView.java
- RecommendationsView.java



## Ficheros empleados

- *Dataset* de películas.

## Estado del sistema tras la ejecución de la prueba

El sistema vinculará la recomendación realizada al usuario.

## Acciones a realizar

Al ejecutar el programa, deberá escogerse el *Dataset* llamado *movies* y pulsar el icono de *Load*. Una vez el usuario se encuentre en la vista *LoginUserView*, es necesario que se introduzca un identificador válido (proporcionado en el apartado *Input*). Tras ser redirigido a la vista *MainView*, será preciso que escoja el algoritmo a testear y seleccione la precisión (ambos valores están definidos en el apartado *Input*). Solo restará hacer click en el botón *Recommend* y, posteriormente, *See more*. Al cargar la vista *RecommendationsView*, podrá verificarse que la recomendación ha sido creada y asociada al perfil.

## Input

- Identificador de usuario a introducir: 117838.
- Algoritmo a escoger: *HybridApproach*.
- Precisión a escoger: *Imprecise*.

## Output esperado

- Identificador de la recomendación.
- Título del ítem.
- Valoración estimada por el algoritmo.
- Puntuación que el usuario asigna a la recomendación (en este caso 0).

## Editar Valoración de un Ítem

### Objetivo

Garantizar el debido funcionamiento del caso de uso *Rate Item*.

## Elementos integrados en la prueba

- *CtrlDomain.java*
- *CtrlUsers.java*

- CtrlDataFactory.java
- User.java
- Item.java

#### **Capa de datos:**

- CtrlData.java
- DataUtils.java
- Utils.java

#### **Capa de presentación:**

- DatasetsView.java
- LoginUserView.java
- MainView.java
- ItemRateView

## Ficheros empleados

- *Dataset* de películas.

## Estado del sistema tras la ejecución de la prueba

El sistema asignará la puntuación proporcionada por el usuario al ítem indicado.

## Acciones a realizar

Al ejecutar el programa, deberá escogerse el *Dataset* llamado *movies* y pulsar el icono de *Load*. Una vez el usuario se encuentre en la vista *LoginUserView*, es necesario que se introduzca un identificador válido (proporcionado en el apartado *Input*). Tras ser redirigido a la vista *MainView*, pulsará el segundo botón de la parte inferior, lo cual lo conducirá a la vista *ItemRatesView*. Allí deberá clicar el icono del lápiz de cualquier ítem de la lista inferior. Al realizar dicha acción, recibirá un *Pop Up* en el que deberá indicar una nueva puntuación válida. Finalmente, el sistema modificará el *rating* del ítem.

## Input

- Identificador de usuario a introducir: 117838.
- Puntuación: 5.0.

## Output esperado

- Puntuación: 5.0.
- El documento `DATA/datasets/movies/temp/ratings.test.known.csv` contendrá el *rate* indicado en el apartado *Input*.

# Eliminar Ítem

## Objetivo

Garantizar el debido funcionamiento del caso de uso *Delete Item*.

## Elementos integrados en la prueba

- CtrlDomain.java
- CtrlUsers.java
- CtrlDataFactory.java
- User.java
- Item.java

### Capa de datos:

- CtrlData.java
- DataUtils.java
- Utils.java

### Capa de presentación:

- DatasetsView.java
- LoginUserView.java
- MainView.java
- ItemRateView.java

## Ficheros empleados

- *Dataset* de películas.

## Estado del sistema tras la ejecución de la prueba

El sistema eliminará el ítem indicado del sistema.

## Acciones a realizar

Al ejecutar el programa, deberá escogerse el *Dataset* llamado *movies* y pulsar el icono de *Load*. Una vez el usuario se encuentre en la vista *LoginUserView*, es necesario que se introduzca un identificador válido (proporcionado en el apartado *Input*). Tras ser redirigido a la vista *MainView*, pulsará al segundo botón de la parte inferior, lo cual lo conducirá a la vista *ItemRatesView*. Allí deberá clicar el icono de la papelera roja de cualquier ítem de la lista inferior. Al realizar dicha acción, recibirá un Pop Up de advertencia. Finalmente, el sistema eliminará el *rate* del ítem.

## Input

- Identificador de usuario a introducir: 117838.

## Output esperado

- El documento DATA/datasets/movies/temp/ratings.test.known.csv ya no contendrá dicho rate.

## Añadir Ítem

### Objetivo

Garantizar el debido funcionamiento del caso de uso *Añadir Ítem*.

### Elementos integrados en la prueba

- CtrlDomain.java
- CtrlUsers.java
- CtrlDataFactory.java
- User.java
- Item.java

#### **Capa de datos:**

- CtrlData.java
- DataUtils.java
- Utils.java

#### **Capa de presentación:**

- DatasetsView.java
- LoginUserView.java
- MainView.java
- ItemAddView.java

### Ficheros empleados

- *Dataset* de películas.

### Estado del sistema tras la ejecución de la prueba

El sistema añadirá el ítem indicado del sistema.

### Acciones a realizar

Al ejecutar el programa, deberá escogerse el *Dataset* llamado *movies* y pulsar el icono de *Load*. Una vez el usuario se encuentre en la vista *LoginUserView*, es necesario que se introduzca un identificador válido (proporcionado en el apartado

*Input*). Tras ser redirigido a la vista *MainView*, pulsará el primer botón de la parte inferior, lo cual lo conducirá a la vista *ItemAddView*. Allí deberá clicar el icono de suma de cualquier ítem de la lista mostrada. Al realizar dicha acción, recibirá un *Pop Up* en el que deberá indicar una puntuación válida. Finalmente, el sistema añadirá el *rate* del ítem.

## Input

- Identificador de usuario a introducir: 117838.
- Puntuación: 5.0.

## Output esperado.

- El documento `DATA/datasets/movies/temp/ratings.test.known.csv` contendrá el *rate* especificado.