

# Recomendador

Decisiones tomadas sobre los tests

**Gerard Madrid Miró**  
**Guillem Gràcia Andreu**  
**Ismael Quiñones Gama**  
**Pol Ken Galceran Kimura**

# Descripción general

En este proyecto hemos decidido hacer todos los drivers y tests con JUnit, ya que esta herramienta nos permite hacer tests de forma semiautomática.

Después de pensar sobre cómo hacer los tests, hemos llegado a la conclusión de que no hacía falta hacer tests para todas las clases. Por eso hemos eliminado algunos tests respecto a la primera entrega, ya que tenemos clases que solo usan funciones muy básicas o son clases privadas y por ende no son testeables o no necesitan dichos tests.

Es por eso que hemos hecho drivers de las clases que hemos pensado que no serían tan simples o que podrían tener algo de dificultad, sin ir aún a los casos extremos.

Más tarde hemos decidido hacer tests, entrando ahora sí en los casos extremos para las clases que más problemas pueden causar o que mejor están preparadas para estos casos: ParserCSV y Data. Estas clases tratan con funciones de lectura del CSV, y por eso deben adaptarse a los errores que pueda tener este.

## ParserCSV

ParserCSV es la clase de testing que más ha cambiado desde la primera entrega. Mientras que en la primera entrega teníamos un txt con información inventada y poco real, para esta entrega le hemos metido líneas de texto de la base de datos usada en la ejecución real.

No solo eso, sino que para los casos extremos, hemos desordenado los valores, cambiado los nombres de los atributos e incluso hemos escrito algunos nombres de distintas maneras o incluso mal, para enseñar lo que pasaría en esos casos. También hemos aprovechado para meter floats en el itemID o userID, donde hemos visto que claramente salta una excepción. Cada test tiene un pequeño comentario explicativo para aclarar cualquier duda al respecto.

## Data

Data es otra de las clases que tiene un caso extremo. Para este caso le hemos metido varios saltos de línea en medio de la descripción del ítem, para comprobar que parsea adecuadamente incluso en estas situaciones.