Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

TESTING REPORT - D04



**C1.04.02**

**Repositorio**: [https://github.com/josrojrom1/C1.04.02](https://github.com/josrojrom1/C1.04.02%C3%A7)

**Autor:**

Parra Méndez, Pablo, pabparmen[@alum.us.es](mailto:josrojrom1@alum.us.es)

**Fecha:** 23/05/2023

**Tabla de contenidos**

[**Resumen**](#_4hmpm5uc85lj) **2**

[**Tabla de revisión**](#_90otg4vxm3ka) **2**

[**Introducción**](#_58vtmc9pw6vo) **2**

[**Contenido**](#_tvy5edcq3b5t) **3**

[**Bibliografía**](#_ysgwrb1dc6k3)

**11**

**Resumen**

En este documento se va a llevar a cabo un desglose detallado a cerca del

*testing suite* completo obtenido por el *coverage* de las pruebas unitarias *JUnit* realizadas sobre el proyecto en su versión final, en concreto sobre las funcionalidades relacionadas con *Enrolment y Activity*

**Tabla de revisión**

| **Versión** | **Fecha** | **Descripción** |
| --- | --- | --- |
| V1 | 25/05/2023 | Creación del testing report |

**Introducción**

El requisito a evaluar por los tests eran el #14 y #15 que dicen lo siguiente:

***#14 [Mandatory] Operations by students on enrolments:***

***- List their own enrolments.***

***- Show the details of their enrolments.***

***- Register an enrolment.***

***- Update or delete an enrolment as long as it is not finalised.***

***- Finalise an enrolment. This requires providing a valid credit card; the system will only store its holder and its lower nibble.***

***#15 [Mandatory] Operations by students on workbooks:***

***- List the activities in their workbooks.***

***- Show the details of their workbooks.***

***- Create a new activity in their workbooks, as long as the corresponding enrolment is finalised.***

***- Update or delete the activities in their workbooks, as long as the corresponding enrolment is finalised.***

**Testing funcional**

**Funcionalidad:** *ENROLMENT*

**Rol:** *STUDENT*

**Test:** List Tutorial with Rol Student

**Ruta:** *“acme>testing>student>tutorial>StudentListTest.java”*

Para el **list** de *Enrolments* se desarrollaron dos casos distintos:

* En el primer caso era uno positivo, donde una vez iniciado sesión como *student1 accedes a los enrolments de ese student*.
* En el segundo caso eran los tests de hacking, donde comprueba que un usuario no autenticado o un usuario que no tenga el rol de student pueda meterse..

Con estos casos de testing para el listado no se han detectado ningún tipo de bugs.

**Test:** Show Enrolment with Rol Student

**Ruta:** *“acme>testing>student>tutorial>StudentEnrolmentShowTest.java”*

Para el test ***show***de *tutorial* se desarrolló los siguientes casos:

* Para comprobar que la información del *Enrolment* se mostrase adecuadamente se creó el fichero *.csv* con un caso de *show* positivo. Para llevarlo a cabo de manera correcta, una vez que se accede al listado, este se ordena con orden ascendente en el campo *code*, así nos aseguramos de que el primer elemento de la lista sea el que intentamos comprobar. Una vez ordenado se accede y se comprueba que los campos tienen los datos correctos.
* Para asegurarnos que la información de un *enrolment* no sea visible para cualquier persona creamos el test de hacking. Para ellos hago una query a la base de datos donde extraigo todos los enrolments de student*1,* después con un bucle for itero la colección y creo un parámetro donde le paso el id de los enrolments y hago la request con distintos principales. Se comprueba con usuarios sin permiso*, mostrando una pantalla de pánico*

Con estos casos de testing para el listado no se han detectado ningún tipo de bugs

**Test:** Create Enrolment with Rol Student

**Ruta:** *“acme>testing>student>enrolemnt>StudentEnrolmentCreateTest.java”*

Para el **create** de *enrolments* se desarrollaron los siguientes casos:

* El primer caso es el test positivo de *create enrolment* donde se siguen los siguientes pasos logueado como *student1*: Click en *“Student > Enrolment”* y de nuevo click en *Register*, después se introducen todos los campos del formulario con las entradas creadas en el fichero *.csv*. Una vez rellenos se hace click en el botón de create y al final se comprueba que no existen errores al publicar el enrolment y que se ha creado correctamente y aparecen todos los campos.
* El segundo caso es el test negativo. En este test se repiten los mismos pasos que en el test anterior pero esta vez en el fichero *.csv* se introducen entradas erróneas comprobando que efectivamente se activan las validaciones correspondientes y que en ningún caso podemos crear un *Enrolment*. Se hacen comprobaciones de entradas en blanco, datos erróneos, o spam. para detectar que ocurran los problemas de manera esperada, en cada entrada del csv se pone un solo error.
* El tercer caso es el test de hacking. Donde se comprueba que ningún principal que no tenga el rol de student tenga la autorización de acceder al formulario ni crear un enrolemnt

**Test:** Update Enrolment with Rol Student

**Ruta:** *“acme>testing>student>enrolment>StudentEnrolmentUpdateTest.java”*

Para el **update** de *enrolment* se desarrollaron los siguientes casos:

* El primer caso es el test positivo de *update* donde una vez logueado como *student1* se siguen los siguientes pasos, se hace click en el menú Student y después a Enrolment, ordena la lista de manera ascendente por el código, para asegurarnos que escogemos el correcto. Una vez ordenado entramos al primer enrolment y rellenamos los campos con los datos que aparecen en el csv, se clicka el botón de update y se comprueba que los campos han sido modificados correctamente.
* El segundo caso es el test negativo. En este test se repiten los mismos pasos que en el test anterior pero esta vez en el fichero *.csv* se introducen entradas erróneas comprobando que efectivamente se activan las validaciones correspondientes y que en ningún caso podemos modificar un *enrolment*. Se hacen comprobaciones de entradas en blanco, datos erróneos, o spam. para detectar que ocurran los problemas de manera esperada, en cada entrada del csv se pone un solo error.
* El tercer caso es el test de hacking. Donde se comprueba que ningún principal que no tenga el rol de student tenga la autorización de acceder al formulario ni crear un tutorial.

Con estos test no encontré ningún bug significativo, simplemente encontré incongruencias en los archivos csv iniciales.

**Test:** Finalise Enrolment with Rol Student

**Ruta:** *“acme>testing>student>enrolment>StudentEnrolmentFinaliseTest.java”*

Para el **Finalise** de *Enrolments* se desarrollaron los siguientes casos:

* El primer caso es el test positivo de *finalise* donde una vez logueado como *student1* se siguen los siguientes pasos, se hace click en el menú Student y después a Enrolment, ordena la lista por el código, para asegurarnos que escogemos el correcto. Una vez ordenado entramos al enrolment deseado y clickamos en el botón publish. Después se comprueba que verdaderamente se haya finalizadp el enrolment, para ello se comprueba que no existan los botones de update, delete y finalise.
* El segundo caso.: es el test negativo para comprobar que los datos metidos en el form extra para finalizar el enrolment son erróneos
* El tercer caso es el test de hacking, donde se comprueba que los usuarios que no sean el creador del tutorial puedan publicarlo. Para ello se prueba sacando todos los enrolments de student1 y con un bucle for los itero y compruebo individualmente, con diferentes usuarios.

Con estos tests no se encontró ningún bug.

**Funcionalidad:** *Activity*

**Rol:** *Student*

**Test:** Create Activity with Rol Student

**Ruta:** *“acme>testing>student>activity>StudentActivityCreateTest.java”*

Para el **create** de *Activity* se desarrollaron los siguientes casos:

* El primer caso es el test positivo de *create Activity* donde se siguen los siguientes pasos logueado como *student2*: Click en *“Student > Activity”* y de nuevo click en la primer activity, ahora se hace click en *create. En el formulario que aparece* se introducen todos los campos del formulario con las entradas creadas en el fichero *.csv*. Una vez rellenos se hace click en el botón submit de create y al final se comprueba que no existen errores al crear el Activity y que se ha creado correctamente y aparecen todos los campos.
* El segundo caso es el test negativo. En este test se repiten los mismos pasos que en el test anterior pero esta vez en el fichero *.csv* se introducen entradas erróneas comprobando que efectivamente se activan las validaciones correspondientes y que en ningún caso podemos crear un *Activity*. Se hacen comprobaciones de entradas en blanco, datos erróneos, o spam. Para detectar que ocurran los problemas de manera esperada, en cada entrada del csv se pone un solo error.
* El tercer caso es el test de hacking. Donde se comprueba que ningún principal que no tenga el rol de student tenga la autorización de acceder al formulario ni crear una activity.

Con estos tests no se encontró ningún bug.

**Test:** Update Activity with Rol Student

**Ruta:** *“acme>testing>student>tactivity>StudentActivityUpdateTest.java”*

Para el **Update** de *Activity* se desarrollaron los siguientes casos:

* El primer caso es el test positivo de *update activity* donde se siguen los siguientes pasos logueado como *student2*: Click en *“Student > Activity”*, ahora se hace click en la primera activity*. En el formulario que aparece* se introducen todos los campos del formulario con las entradas creadas en el fichero *.csv*. Una vez rellenos se hace click en el botón submit de update y al final se comprueba que no existen errores al modificar la Activity y que se ha modificado correctamente y aparecen todos los campos.
* El segundo caso es el test negativo. En este test se repiten los mismos pasos que en el test anterior pero esta vez en el fichero *.csv* se introducen entradas erróneas comprobando que efectivamente se activan las validaciones correspondientes y que en ningún caso podemos modificar un *Activity*. Se hacen comprobaciones de entradas en blanco, datos erróneos, o spam. Para detectar que ocurran los problemas de manera esperada, en cada entrada del csv se pone un solo error.
* El tercer caso es el test de hacking. Donde se comprueba que ningún principal que no tenga el rol de student tenga la autorización de acceder al formulario ni modificar una activity.

Con estos tests no se encontró ningún bug.

**Test:** List Enrolment with Rol Student

**Ruta:** *“acme>testing>student>activity>StudentActivityListTest.java”*

Para el **list** de *Activity* se desarrollaron tres casos distintos:

* En el primer caso era uno positivo, donde una vez iniciado sesión como *student2,* se hace click en el menú Student y se le da al botón Activity, una vez dentro se comprueba que los datos sean correctos.
* En el segundo caso eran los tests de hacking, donde comprueba que un usuario no autenticado o un usuario que no tenga el rol de student no pueda acceder a la información.

Con estos casos de testing para el listado no se han detectado ningún tipo de bugs.

**Test:** Show Activity with Rol Student

**Ruta:** *“acme>testing>student>activity>StudentActivityShowTest.java”*

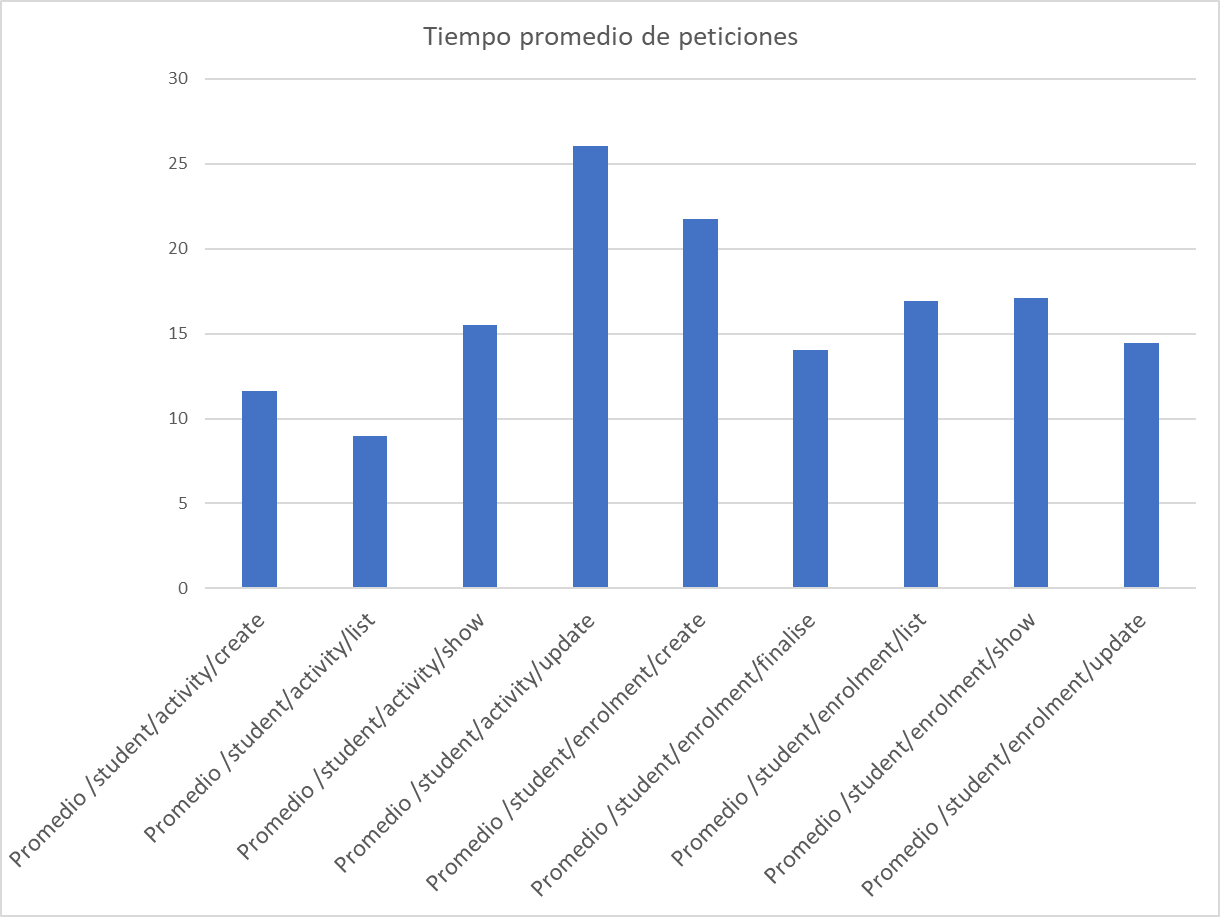
Para el test ***show***de *activity* se desarrolló los siguientes casos:

* En el primer caso, inicia sesión como student2 y hace click en el botón del menú Student y seguidamente al de Activity, en esta lista entra al primero y comprueba que la información se muestra correctamente.
* En el tercer caso eran los tests de hacking, donde comprueba que un usuario no autenticado o un usuario que no tenga el rol de student no pueda acceder a la información.

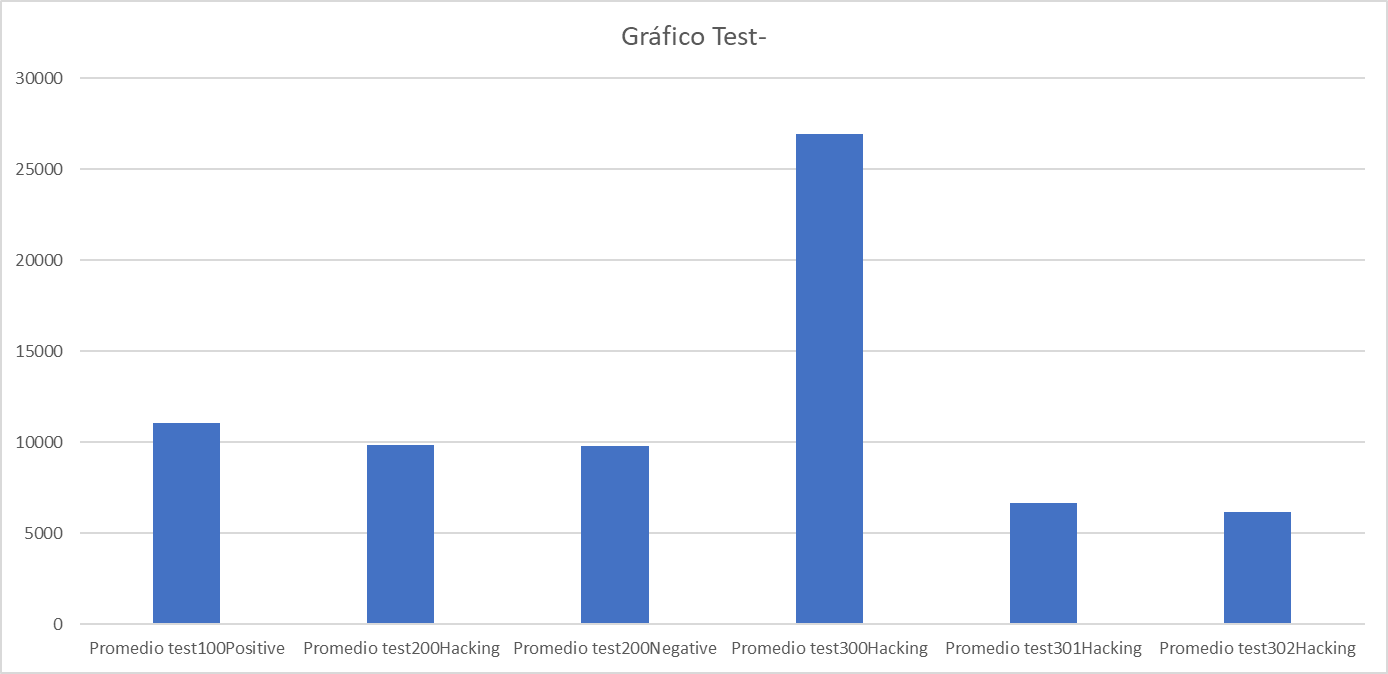
Para estos test ocurrían lo que pensaba que era un bug, porque no me detectaba correctamente los valores de dos select. Una vez preguntado en un follow up, la profesora me indicó que para hacer la búsqueda en la vista debía poner “\_proxy” después del nombre del atributo que buscaba para que se hiciese correctamente. A parte de este apunte, no encontré ningún bug.

**Testing de rendimiento**

**REQUEST**



**TEST**

****

# Bibliografía

* Documento 08 Annexes, de los contenidos de la plataforma virtual de esta asignatura.
* Documentos del lesson L04 - formal testing