

DP2 2021-2022

D05 – Temas Avanzados

Informe de Testeo

URL Github: <https://github.com/migueegomez7/Acme-Toolkits>

Miembros:

- Miguel Ángel Gómez Gómez (miggomgom1@alum.us.es)
- Dámaris Gómez Serrano (damgomser@alum.us.es)
- Mariano Martín Avecilla (marmarave@alum.us.es)
- Iván Moreno Granado (ivamorgra@alum.us.es)
- Miguel Ángel Rivas Rosado (migrivos@alum.us.es)
- Rafael Sanabria Espárrago (rafasana9@gmail.com)

GRUPO E3.04

Versión 1.0

02-06-2022

Tabla de contenidos

| | |
|--|---|
| Tabla de contenidos | 2 |
| Historial de versiones..... | 3 |
| Introducción | 4 |
| Resumen ejecutivo..... | 4 |
| Contenido..... | 4 |
| Conocimiento posterior sobre el diseño de pruebas..... | 4 |
| Estructura de una prueba | 4 |
| Buenas prácticas para las pruebas | 5 |
| Testeo sobre una arquitectura WIS por capas | 5 |
| Tests de listado | 5 |
| Tests de mostrar | 6 |
| Tests de crear..... | 6 |
| Tests de actualizar | 7 |
| Performance test | 7 |
| Conclusiones | 8 |
| Bibliografía | 8 |

Historial de versiones

| Fecha | Versión | Descripción de los cambios | Sprint |
|------------|---------|--|--------|
| 02/06/2022 | V1.0 | <ul style="list-style-type: none">• Creación del documento• Introducción• Resumen Ejecutivo• Contenido• Conclusiones | 5 |

Introducción

Este documento presenta los conocimientos del equipo de trabajo sobre el diseño y pruebas de un sistema software, una vez cursada la asignatura de Diseño y Pruebas II. Ha sido redactado por el equipo de trabajo pensando en los conocimientos adquiridos a lo largo del curso, tanto en la asignatura de Diseño y Pruebas I como en la asignatura de Diseño y Pruebas II.

Tras un debate previo se han decidido incluir solo los conocimientos que verdaderamente el equipo de trabajo ha aprendido e interiorizado a lo largo del desarrollo del proyecto de la asignatura.

A continuación, se muestra un resumen ejecutivo del informe, el contenido del mismo en el que se especifica todo lo aprendido anteriormente, las conclusiones y la bibliografía.

Resumen ejecutivo

El objetivo de este documento es detallar lo aprendido a lo largo de la asignatura en el proceso de diseño y creación de pruebas sobre un sistema software.

Para ello se proporciona a continuación los conocimientos teóricos, desde un punto de vista de mayor a menor abstracción, de todos los recursos en los que nos apoyamos para realizar desde un punto de vista práctico la realización de las pruebas.

Contenido

Conocimiento posterior sobre el diseño de pruebas

Estructura de una prueba

La estructura de una prueba viene dada en varios ficheros:

- Fichero csv que contiene los datos con los que se trabajará en dicha prueba
- Fichero java donde se alojará la implementación de dicha prueba

Para rellenar el fichero csv se debe tener en cuenta primero el orden en el que aparecen los elementos en la vista. Además, se le debe añadir un atributo nuevo, llamado recordIndex, que es el encargado de numerar las filas de dicho csv.

En cuanto al fichero java, en el siguiente apartado podremos ver algunos ejemplos de cómo se han realizado algunas de las pruebas.

Buenas prácticas para las pruebas

Para lograr un buen testeo de nuestro sistema, se recomienda realizar las siguientes prácticas:

- Parametrización, siempre que sea posible, las pruebas unitarias.
- Mantener las pruebas unitarias enfocadas e independientes
- Mantener claro la causa y el efecto de cada test
- Probar casos extremos aceptados en los test positivos y casos extremos rechazados en los test negativos.

Testeo sobre una arquitectura WIS por capas

En la asignatura DP-2 aprendimos a hacer uso de la arquitectura por capas, utilizando el patrón Modelo-Vista-Controlador. Para ello, nos enseñaron a hacer test para comprobar las siguientes acciones: listar, mostrar, crear, actualizar y borrar.

Tests de listado

Solo tiene caso positivo debido a que se esperan una serie determinada de datos a listar. A continuación, nos encontramos con un test del listado y show de Tools.

```
package acme.testing.any.tool;

import org.junit.jupiter.api.Order;

public class AnyToolListAllTest extends TestHarness{

    @ParameterizedTest
    @CsvFileSource(resources = "/any/tool/tool.csv", encoding = "utf-8", numLinesToSkip = 1)
    @Order(10)
    public void positive(final int recordIndex, final String name, final String code, final String technology, final String description,
                        final String retailPrice, final String moreInfo) {

        super.clickOnMenu("Anonymous", "List tools");

        super.checkListingExists();
        super.sortListing(0, "asc");
        super.checkColumnHasValue(recordIndex, 2, technology);
        super.checkColumnHasValue(recordIndex, 1, code);
        super.checkColumnHasValue(recordIndex, 0, name);
        super.checkColumnHasValue(recordIndex, 3, description);

        super.clickOnListingRecord(recordIndex);
        super.checkFormExists();
        super.checkInputBoxHasValue("name", name);
        super.checkInputBoxHasValue("code", code);
        super.checkInputBoxHasValue("technology", technology);
        super.checkInputBoxHasValue("description", description);
        super.checkInputBoxHasValue("retailPrice", retailPrice);
        super.checkInputBoxHasValue("moreInfo", moreInfo);
    }
}
```

| recordIndex | name | code | technology | description | retail-price | more-info |
|-------------|-----------------|-----------|---------------|---------------|--------------|--------------|
| 0 | correa | ZFQ-243-Y | tecnologia nu | description 1 | EUR 0.00 | http://www.g |
| 1 | h | AAA-000-A | java | description-1 | EUR 0.10 | http://www.e |
| 2 | llave | ZAC-999-Z | tecnologia2.3 | description-6 | USD 532.22 | http://www.g |
| 3 | llave de paso | ZAN-124-Z | techonology | d | USD 20,000.2 | http://www.g |
| 4 | martillo | AAA-123-A | tapiz | description-5 | USD 50.00 | http://www.g |
| 5 | memoria ram | WWW-122 | tecnologia m | description 1 | GBP 0.01 | http://www.g |
| 6 | memoria SSD | WWW-123-H | tecnologia m | description 1 | USD 0.10 | http://www.g |
| 7 | tornillotornill | AAA-001-A | h | description-2 | EUR 5.00 | http://www.g |

Tests de mostrar

Pasa lo mismo que en los test de listado, solo tiene sentido realizar casos positivos. Hay un ejemplo en la imagen [anterior](#).

Tests de crear

Crear caso positivo:

```
public class AdministratorAnnouncementCreateTest extends TestHarness {
    // Lifecycle management -----
    // Test cases -----
    @ParameterizedTest
    @CsvFileSource(resources = "/administrator/announcement/create-positive.csv", encoding = "utf-8", numLinesToSkip = 1)
    @Order(10)
    public void positiveTest(final int recordIndex, final String title, final String body, final String email, final String critical, final String moreInfo, final String confirmation) {
        super.signIn("administrator", "administrator");

        super.clickOnMenu("Administrator", "List announcements");
        super.checkListingExists();

        super.clickOnButton("Create");

        super.fillInputBoxIn("title", title);
        super.fillInputBoxIn("body", body);
        super.fillInputBoxIn("email", email);
        super.fillInputBoxIn("critical", critical);
        super.fillInputBoxIn("moreInfo", moreInfo);
        super.fillInputBoxIn("confirmation", confirmation);
        super.clickOnSubmit("Create");

        super.clickOnMenu("Administrator", "List announcements");
        super.checkListingExists();
        super.sortListing(0, "asc");
        super.checkColumnHasValue(recordIndex, 1, title);
        super.checkColumnHasValue(recordIndex, 2, body);
        super.clickOnListingRecord(recordIndex);

        super.checkFormExists();
        super.checkInputBoxHasValue("title", title);
        super.checkInputBoxHasValue("body", body);
        super.checkInputBoxHasValue("email", email);
        super.checkInputBoxHasValue("critical", critical);
        super.checkInputBoxHasValue("moreInfo", moreInfo);

        super.signOut();
    }
}
```

| recordIndex | title | body | email | critical | more-info | confirmation |
|-------------|------------------|-----------------|---------------|----------|---------------|--------------|
| 0 | This is an ann | This is the anr | test@mail.coi | false | | true |
| 1 | Be careful wit | Please take in | administrator | true | https://www.ç | true |
| 2 | You cannot ha | Remember th | administrator | true | https://www.è | true |
| 3 | Title test title | Announcemei | test2@mail.co | false | | true |
| 4 | How to create | You have to c | useful-inform | false | https://www.r | true |
| 5 | How to becoi | Sign in and cl | useful-inform | true | https://www.r | true |
| 6 | This is the las | This is the las | test@mail.coi | false | https://www.r | true |

Crear caso negativo:

```
@ParameterizedTest
@CsvFileSource(resources = "/administrator/announcement/create-negative.csv", encoding = "utf-8", numLinesToSkip = 1)
@Order(20)
public void negativeTest(final int recordIndex, final String title, final String body, final String email, final String critical, final String moreInfo, final String confirmation) {

    super.signIn("administrator", "administrator");

    super.clickOnMenu("Administrator", "List announcements");
    super.clickOnButton("create");
    super.checkFormExists();

    super.fillInputBoxIn("title", title);
    super.fillInputBoxIn("body", body);
    super.fillInputBoxIn("email", email);
    super.fillInputBoxIn("critical", critical);
    super.fillInputBoxIn("moreInfo", moreInfo);
    super.fillInputBoxIn("confirmation", confirmation);

    super.clickOnSubmit("create");

    super.checkErrorsExist();

    super.signOut();
}

@Test
@Order(30)
public void hackingTest() {
    super.checkMottLinkExists("Account");
    super.navigate("/administrator/announcement/create");
    super.checkPanicExists();

    super.signIn("patron1", "patron1");
    super.navigate("/administrator/announcement/create");
    super.checkPanicExists();
    super.signOut();

    super.signIn("inventor1", "inventor1");
    super.navigate("/administrator/announcement/create");
    super.checkPanicExists();
    super.signOut();
}

// Auxiliary methods -----
```

| recordIndex | title | body | email | critical | more-info | confirmation |
|-------------|-------------------|----------------|---------------|----------|---------------|--------------|
| 0 | This is a title t | excededlengt | damgomser@ | false | | true |
| 1 | excededlengt | body | negative-test | true | http://more-i | true |
| 2 | Negative test | Negative test | example@m | false | | false |
| 3 | It will show a | Of course it w | alum | true | | true |
| 4 | | | | | | true |
| 5 | | This will show | showing-erro | false | https://exam | true |
| 6 | Error in the b | | someone@gr | true | | true |

Tests de actualizar

Los test de actualizar se realizan de manera similar a los test de crear, la única diferencia es que se realizan a la hora de actualizar un elemento. En ambos casos lo ideal es que se comprobasen casos extremos para comprobar las validaciones de los campos del sistema y detectar posibles fallos.

Performance test

Para realizar el test de rendimiento es necesario ejecutar todas las pruebas de nuestro sistema. Una vez ejecutadas y aseguradas de que todas y cada una de ellas han resultado ser un éxito, debemos extraer el log ubicado en la carpeta de logs del directorio del proyecto.

Este archivo .log se debe exportar a un Excel de manera en la que podamos tratar los datos obtenidos en la ejecución de las pruebas. Una vez hecho esto, debemos extraer los parámetros necesarios para valorar cómo funciona el sistema y sacar las conclusiones. Este test se puede hacer siempre que se haga como mínimo junto a otra persona, para comparar las velocidades conseguidas por cada ordenador.

Conclusiones

En general, en el documento se detalla el conocimiento obtenido en la asignatura de Diseño y Pruebas II sobre el proceso de diseño e implementación de las pruebas funcionales sobre un sistema de información web.

En general, se ha trabajado con la ayuda del Framework GeckoDriver, encargado de, junto con Mozilla, hacer de usuario en modo “marioneta”. Desde nuestro punto de vista, es algo que ha cambiado por completo nuestra percepción en el diseño e implementación de estas pruebas, ya que hemos presenciado en primera persona cómo el Framework actúa.

Por otro lado, hemos aprendido a hacer test de rendimiento de nuestra aplicación. Gracias a ello, tenemos el conocimiento suficiente para poder sacar conclusiones sobre los datos extraídos de este test.

Se ha enfocado profundamente en el aprendizaje de pruebas paramétricas que, junto con los ficheros csv, nos ha permitido controlar las acciones realizadas en las simulaciones de las pruebas.

Como aspecto negativo, consideramos que los tests soportados por ficheros csv a medio o largo plazo supone mucho tiempo perdido, ya que hay elementos en el sistema que tienen atributos que pueden anticuarse y, por tanto, no aparecer en vistas, como el caso de las chirps en nuestro caso. Esto representa un punto negativo en cuanto a la mantenibilidad de los tests.

Bibliografía

Intencionadamente en blanco.