

**PROTOCOLO DE PRUEBAS PARA EL SISTEMA DE APOYO A LA MAESTRÍA  
EN INFORMÁTICA PARA LA EDUCACIÓN**

**RUBÉN DARÍO GUERRERO RIVERA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS  
BUCARAMANGA  
2018**

**PROTOCOLO DE PRUEBAS PARA EL SISTEMA DE APOYO A LA MAESTRÍA  
EN INFORMÁTICA PARA LA EDUCACIÓN**

**RUBÉN DARÍO GUERRERO RIVERA**

**Trabajo de Grado para Optar el Título de Ingeniero de Sistemas**

**DIRECTOR**

**GABRIEL RODRIGO PEDRAZA FERREIRA  
DOCTOR EN CIENCIAS DE LA COMPUTACIÓN**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICO MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS  
BUCARAMANGA**

**2018**

## **DEDICATORIA**

Culminar este proyecto de grado para convertirme en profesional es el resultado del apoyo de tantas personas bondadosas que sería imposible mencionarlas a todas ellas en esta página.

Quisiera poder dedicar este triunfo a cada una de esas personas que saben bien cuánto aportaron y lo importante que fueron para que hoy yo pueda conseguir este sueño, aunque mencionarlas a todas ellas aquí resulte imposible.

Sin embargo, no puedo dejar de mencionar a mi madre, Gladys Rivera, a quien le debo el haber soñado antes de mí, con tener un hijo profesional egresado de una institución de alta calidad como la UIS.

Gracias especiales a ti mamá por enseñarme a soñar mucho más allá de mis posibilidades.

Este triunfo también es tuyo como todos los que se han derivado de tu decisión de ser cabeza de nuestro hogar. Gracias por guiarnos y luchar toda tu vida por nosotros.

**Rubén Darío Guerrero Rivera**

## **AGRADECIMIENTOS**

Debo agradecer a muchos compañeros, profesores y personas que aportaron de manera significativa de una u otra forma en mi formación como profesional y para que este proyecto se pudiera llevar a cabo.

Especialmente al Dr. Gabriel Rodrigo Pedraza, a quien tuve la fortuna de tener como director de este proyecto y profesor. Muchas gracias por su paciencia y por esa disposición a compartir su conocimiento siempre con sus estudiantes. Es una fortuna encontrar profesores que saben enseñar tan bien y exigir en la justa proporción, estoy seguro que no soy el único estudiante que agradece haber sido formado por una persona como usted. Muchas gracias.

Al grupo SIMON, especialmente a Eduardo Guerra por su tiempo y paciencia durante la realización del proyecto, su apoyo fue invaluable.

Muchas gracias a todos los profesores que con paciencia y vocación influyeron en mi formación profesional, saben bien que son muy pocos pero son los responsables de que la Universidad Industrial de Santander siga siendo gloriosa.  
¡Millones de gracias a ustedes!

## CONTENIDO

	<b>Pág.</b>
INTRODUCCIÓN .....	17
1. OBJETIVOS.....	18
1.1 OBJETIVO GENERAL .....	18
1.2 OBJETIVOS ESPECÍFICOS.....	18
2. FUNDAMENTACIÓN TEÓRICA .....	19
2.1 FUNDAMENTOS BÁSICOS .....	19
2.1.1 Aplicaciones web y calidad. ....	19
2.1.2 Pruebas de Software. ....	20
2.1.3 Documentación de un proceso de pruebas de software. ....	20
2.1.4 Protocolo de pruebas. ....	21
2.1.5 Informe de resultados y errores. ....	23
2.2 ESTRATEGIAS DE PRUEBAS SOFTWARE.....	23
2.2.1 Pruebas Unitarias. ....	24
2.2.2 Pruebas de Integración.....	27
2.2.3 Pruebas de Regresión. ....	28
2.2.4 Pruebas de Sistema.....	30
2.2.5 Pruebas no funcionales.....	31
2.2.5.1 Pruebas de seguridad. ....	31
2.2.5.2 Pruebas de rendimiento. ....	32
2.2.5.3 Pruebas de carga.....	32
2.2.5.4 Pruebas de usabilidad. ....	33
2.3 HERRAMIENTAS PARA DOCUMENTACIÓN Y AUTOMATIZACIÓN DE PRUEBAS SOFTWARE .....	34
2.3.1 Apache JMeter.....	34

2.3.2 BugZilla.....	35
2.3.3 Selenium.....	35
2.3.4 Testlink.....	36
2.3.5 Gt Metrix.....	36
2.3.6 MantisBT.....	37
3. GENERALIDADES DEL SOPORTE INFORMÁTICO MIE.....	38
3.1 CONTEXTUALIZACIÓN DEL PROYECTO .....	38
3.2 PROCESO PLANTEADO DE DESARROLLO DEL SOPORTE INFORMÁTICO MIE-UIS .....	38
3.2.1 Equipo requerido.....	38
3.2.2 Proceso de desarrollo .....	39
3.3 DISEÑO PLANTEADO DEL SISTEMA.....	39
3.3.1 Usuarios del sistema.....	39
3.3.2 Requerimientos funcionales.....	40
3.3.3 Requerimientos no funcionales.....	42
3.4 DISEÑO DESARROLLADO DEL SISTEMA .....	42
3.4.1 Asignatura.....	43
3.4.2 Modulo.....	44
3.4.3 Actividad.....	44
3.4.4 Recursos.....	45
3.4.5 Usuarios.....	45
3.4.6 Menú.....	46
3.4.7 Salir.....	46
4. DISEÑO PROTOCOLO DE PRUEBAS .....	47
4.1 GENERALIDADES DEL PROTOCOLO.....	47
4.1.1 Roles del equipo de calidad.....	47
4.1.2 Tipos de pruebas y ciclos.....	48
4.1.3 Formato para el diseño de casos de prueba funcionales.....	49

4.1.4 Formato para el diseño de casos de prueba no funcionales (rendimiento y carga).....	50
4.1.5 Tipos de errores.....	51
4.1.6 Prioridad.....	52
4.1.7 Formato para el reporte de incidencias.....	52
4.1.8 Formato para el reporte de resultados.....	53
4.2 PROTOCOLO DE PRUEBAS REQUERIMIENTOS FUNCIONALES .....	54
4.2.1 Pruebas Unitarias o de Módulo.....	54
4.2.2 Pruebas de Regresión.....	59
4.2.3 Pruebas de Integración.....	60
4.2.4 Pruebas de Sistema.....	64
4.3 PROTOCOLO DE PRUEBAS REQUERIMIENTOS NO FUNCIONALES.....	65
4.3.1 Métricas para pruebas de carga y rendimiento.....	67
4.3.2 Pruebas de Rendimiento.....	68
4.3.3 Pruebas de Carga.....	71
4.3.4 Pruebas de Usabilidad.....	74
4.3.4.1 Imagen corporativa.....	75
4.3.4.2 Contenido y organización.....	75
4.3.4.3 Pruebas de tiempo.....	76
4.3.5 Pruebas de compatibilidad.....	78
5. VALIDACION DEL PROTOCOLO DE PRUEBAS .....	80
5.1 GENERALIDADES DE LA VALIDACION.....	80
5.1.1 Criterios de selección de casos de prueba y ciclos.....	80
5.1.1.1 Primer ciclo: .....	80
5.1.1.2 Segundo ciclo.....	81
5.2 PRUEBAS REQUERIMIENTOS FUNCIONALES.....	81
5.2.1 Primer ciclo .....	81
5.2.1.1 Casos de prueba y ejecución.....	82
5.2.1.2 Reporte de incidencias.....	83

5.2.1.3 Reporte de resultados.....	84
5.2.2 Segundo ciclo. ....	85
5.2.2.1 Casos de prueba y ejecución.....	85
5.2.2.2 Reporte de incidencias.....	87
5.2.2.3 Reporte de resultados.....	88
5.3 PROTOCOLO DE PRUEBAS REQUERIMIENTOS NO FUNCIONALES.....	88
5.3.1 Métricas para pruebas de carga y rendimiento. ....	88
5.3.2 Pruebas de Rendimiento.....	89
5.3.3 Pruebas de Carga.....	93
5.3.4 Pruebas de Usabilidad.....	95
5.3.5 Pruebas de compatibilidad.....	98
6. CONCLUSIONES .....	99
7. RECOMENDACIONES.....	100
BIBLIOGRAFÍA.....	103

## LISTA DE TABLAS

Tabla 1 Requerimientos funcionales del sistema.....	40
Tabla 2 Formato para el diseño de casos de prueba funcionales.....	50
Tabla 3 Formato para el diseño de casos de pruebas no funcionales. ....	51
Tabla 4 Tipos de errores.....	51
Tabla 5 Prioridad de un error .....	52
Tabla 6 Formato para el reporte de incidencias.....	53
Tabla 7 Formato para el reporte de resultados.....	54
Tabla 8 Relación entre paquetes, módulos y funcionalidades .....	55
Tabla 9 Ejemplo caso de prueba unitaria 1 .....	57
Tabla 10 Ejemplo caso de prueba unitaria 2.....	58
Tabla 11 Ejemplo caso de prueba unitaria 3.....	59
Tabla 12 Ejemplo caso de uso de integración MIE-TCI121 .....	63
Tabla 13 Métricas de carga y rendimiento .....	68
Tabla 14 Ejemplo 1 Caso de prueba de rendimiento. ....	69
Tabla 15 Ejemplo 2 Caso de prueba de rendimiento. ....	70
Tabla 16 Ejemplo 1 Caso de prueba de carga.....	72
Tabla 17 Ejemplo 2 Caso de prueba de carga.....	72
Tabla 18 Ejemplo 3 Caso de prueba de carga.....	73
Tabla 20 Casos de prueba funcionales primer ciclo de validación.....	81
Tabla 21 Caso de prueba MIE-TCU011.....	82
Tabla 22 Reporte de incidencia en MIE-TCU0111.....	83
Tabla 23 Tabla de resultados ciclo 1 .....	84
Tabla 24 Diseño de segundo ciclo.....	85
Tabla 25 Caso de prueba MIE-TCU012.....	86
Tabla 26 Reporte de incidencias MIE-TCU142.....	87
Tabla 27 Reporte de incidencias MIE-TCU514.....	87

Tabla 28 Reporte de resultados segundo ciclo .....	88
Tabla 29 Ejemplo 1 Caso de prueba de rendimiento. ....	89
Tabla 30 Ejemplo 2 Caso de prueba de rendimiento. ....	91
Tabla 31 Encuesta prueba de usabilidad .....	95
Tabla 27 Resultados prueba de compatibilidad .....	98

## LISTA DE FIGURAS

Figura 1 Controladores y representantes en pruebas unitarias .....	26
Figura 2 Tipos de prueba y ciclos .....	49
Figura 3 Diagrama de dependencia paquete: Asignatura .....	60
Figura 4 Diagrama de dependencia paquete: Módulo .....	61
Figura 5 Diagrama de dependencia paquete: Actividad .....	61
Figura 6 Diagrama de dependencia paquete: Recurso.....	61
Figura 7 Diagrama de dependencia paquete: Usuarios.....	62
Figura 8 Dependencia entre dos modulos para prueba de integración.....	62
Figura 9 Diagrama de interdependencia entre paquetes .....	65
Figura 10 Script y resultado para MIE-TCU011 .....	83
Figura 11 Reporte de incidencia en MIE-TCU011 en la herramienta MantisBT ....	84
Figura 12 Script y resultado para MIE-TCU012 .....	86
Figura 13 Script de prueba en Jmeter para MIE-TCR120.1.....	90
Figura 14 Resultado gráfico de prueba en Jmeter para MIE-TCR120.1 .....	91
Figura 15 Script de prueba en Jmeter para MIE-TCR120.2.....	92
Figura 16 Resultado gráfico de prueba en Jmeter para MIE-TCR120.2.....	92
Figura 17 Resultados en Jmeter para MIE-TCC80.1 .....	93
Figura 18 Resultados prueba de carga MIE-TCC120.1 .....	94
Figura 19 Resutados MIE-TCC250.1 .....	94

## **LISTA DE ANEXOS**

(Los anexos están adjuntos en el CD y puede visualizarlos en base de datos de la biblioteca UIS)

ANEXO A. Validación Ciclo 1

ANEXO B. Validación Ciclo 2

## RESUMEN

**TITULO:** PROTOCOLO DE PRUEBAS PARA EL SISTEMA DE APOYO A LA MAESTRÍA EN INFORMÁTICA PARA LA EDUCACIÓN\*

**AUTOR:** RUBÉN DARÍO GUERRERO RIVERA\*\*

**PALABRAS CLAVE:** Protocolo, pruebas, metodologías, calidad, automatización, funcionales, estrategias, validación.

**DESCRIPCIÓN:** El presente trabajo de grado implementa un protocolo de pruebas para la herramienta de apoyo al programa de Maestría en Informática para la Educación de la escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander denominada “Sistema de apoyo a la Maestría en Informática para la Educación” que de forma metódica y organizada sirve de guía para que el equipo de calidad lleve a cabo la validación del sistema en sus requerimientos funcionales y no funcionales. Para llevar a cabo esta tarea presenta una sección de fundamentación teórica que sirve como base para comprender el documento y otros aspectos relacionados con la calidad de software, más específicamente pruebas funcionales, pruebas no funcionales, metodologías y estrategias a tener en cuenta durante el diseño del protocolo al igual que durante la ejecución del mismo. Enseña algunas herramientas muy útiles que pueden ser usadas de apoyo durante la ejecución del protocolo y que pueden encontrarse fácilmente en el mercado. Se presenta un resumen del diseño inicialmente planteado del sistema al igual que una aproximación al sistema desarrollado y posteriormente se presenta el diseño del protocolo con cada uno de los formatos necesarios sin hacerlo dependiente a ninguna herramienta de automatización en específico definiendo su forma de uso y aplicación en cada posible escenario. Se finaliza el documento con la realización de una validación del protocolo diseñado que a su vez sirve de ejemplo de uso y aplicación.

\* Trabajo de grado.

\*\* Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Ph.D Gabriel Rodrigo Pedraza Ferreira.

## ABSTRACT

**TITLE:** TEST PROTOCOL FOR THE “SISTEMA DE APOYO A LA MAESTRÍA EN INFORMÁTICA PARA LA EDUCACIÓN” \*

**AUTHORS:** RUBÉN DARÍO GUERRERO RIVERA\*\*

**KEYWORDS:** Protocol, tests, methodologies, quality, automation, functional, strategies, validation.

**ABSTRACT:** This bachelor thesis implements a test protocol for the tool of support to the Master's program of the Escuela de Ingeniería de Sistemas e Informática of the Universidad Industrial de Santander: "Maestría en Informática para la Educación". The tool is called “Sistema de apoyo a la Maestría en Informática para la Educación” and the present test protocol is a methodical and organized way to guide the quality team to carry validation of the system mentioned in its functional and non-functional requirements. To carry out this task, it presents a theoretical section that serves as a basis for understanding the document and other aspects related to software quality, more specifically functional tests, non-functional tests, methodologies and strategies to be taken into account during the design of the protocol as well as during the execution of it. This protocol also teaches some very useful tools that can be used as support during the execution of the protocol and that can easily be found in the market. In order to know the tool to be evaluated, the protocol presents a summary of the initial tool's design, as well as an approach to the system developed. The design of the protocol is presented with each of the necessary formats without making it dependent on any specific automation tool, defining its use and application in each possible scenario. The document is finalized with a validation of the designed protocol that in turn serves as an example of use and application.

\* Bachelor Thesis.

\*\* Faculty of Physical-Mechanical Engineering. School of Engineering and Computer Science.  
Director: Ph.D Gabriel Rodrigo Pedraza Ferreira.

## INTRODUCCIÓN

El desarrollo de un producto software tiene como fin principal satisfacer una necesidad existente, por ello, debe ser cuidadosamente estudiado con el fin de que se convierta en una herramienta que permita realizar una tarea de manera más eficiente y que no termine siendo una carga adicional al trabajo que ya se realizaba antes de su existencia.

El sistema de apoyo a la Maestría en Informática para la Educación tiene como objetivo principal, como sugiere su nombre, convertirse en una herramienta de apoyo al programa de Maestría en Informática para la Educación de la escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander. Para que este propósito se cumpla, es lógico pensar que el principal mecanismo consistirá en realizar un conjunto de pruebas de diferente tipo que garanticen el funcionamiento del sistema con la calidad adecuada durante su fase de producción. Sin embargo, hay muchos tipos de pruebas que se pueden realizar con una amplia cantidad y diversidad de herramientas y métodos por lo cual se hace necesario planificar estratégicamente las pruebas de acuerdo a las necesidades, de tal forma que todo el equipo hable un mismo idioma y tenga claro la metodología de cómo se deben diseñar, desarrollar, ejecutar, interpretar y reportar las diferentes pruebas.

Este proyecto busca ofrecer un protocolo de pruebas que de forma ágil y organizada permita a un equipo de calidad realizar la validación necesaria para garantizar el cumplimiento de los requerimientos funcionales y no funcionales establecidos para la plataforma MIE.

## **1. OBJETIVOS**

### **1.1 OBJETIVO GENERAL**

Definir un protocolo de pruebas para el sistema de apoyo a la maestría en informática para la educación que valide propiedades funcionales y no funcionales del sistema.

### **1.2 OBJETIVOS ESPECÍFICOS**

**Proponer una metodología para probar funcionalmente el sistema “Soporte Informático MIE-UIS”.**

**Proponer una metodología que permita probar los principales atributos no funcionales del sistema “Soporte Informático MIE-UIS”.**

**Validar el protocolo propuesto mediante la realización de un subconjunto de las pruebas propuestas.**

## 2. FUNDAMENTACIÓN TEÓRICA

### 2.1 FUNDAMENTOS BÁSICOS

**2.1.1 Aplicaciones web y calidad.** Dado que el sistema que se pretende evaluar consiste en una aplicación basada en web, el primer paso es conocer las principales características que las diferencian de otros programas. Partimos definiendo que las aplicaciones web ya no sólo se limitan a mostrar contenidos planos con textos y gráficos limitados sino que por el contrario han evolucionado en sistemas que en ocasiones resultan extremadamente complejos. Pressman<sup>1</sup> define que los atributos más importantes para este tipo de aplicaciones son:

**Uso intensivo de redes**

**Concurrencia**

**Rendimiento**

**Disponibilidad**

**Orientada a datos**

**Estética**

**Inmediatez**

**Seguridad**

Inicialmente se creía que lo importante a la hora de diseñar una aplicación web era tener una vaga idea inicial sobre lo que se pretendía e ir improvisando sobre la marcha, sin embargo, este enfoque ha ido cambiando en la medida que las aplicaciones web han dejado de ser sencillos sitios web netamente informativos y han comenzado a tener más funcionalidades porque lo anterior necesariamente implica un diseño detallado y todo un equipo especializado encargado de cuidar el

---

<sup>1</sup> PRESSMAN, Roger. Ingeniería del Software, Un enfoque práctico. 7ª Edición, México: McGraw-Hill. 2002 p.9 (La naturaleza única de las webapps). ISBN 978-607-15-0314-5

cumplimiento de diversos criterios funcionales y no funcionales. Este equipo se conoce como el equipo de calidad, quien es el responsable de verificar y validar que el programa cumpla con todos los criterios definidos para él.

**2.1.2 Pruebas de Software.** Lastimosamente, todos los productos software presentan algún tipo de falla de menor o mayor importancia, es algo inherente al proceso de creación de éstos productos. Sin embargo, con el fin de disminuir la cantidad de fallas y en búsqueda de productos cada día mejores, se realizan pruebas y análisis rigurosos a diario.

Entonces, se definen las pruebas de software como aquel proceso que busca evaluar integralmente un programa determinado con el fin de garantizar su correcto funcionamiento. Constituye un proceso muy importante al punto que muchas organizaciones desarrolladoras de software destinan entre el 30% y 40% del esfuerzo en un proyecto a la realización de pruebas.

El objetivo de las pruebas no es otro que el de demostrar que el software hace lo que se supone que debe hacer, como lo debe hacer, garantizando la calidad del producto, un tema de gran importancia actualmente.

Las pruebas de software pueden dividirse de acuerdo a su propósito y método de aplicación. Dado que todo programa debe cumplir con unos requerimientos funcionales y no funcionales, entonces existen pruebas para cada uno de estos tipos de requerimientos.

**2.1.3 Documentación de un proceso de pruebas de software.** De la misma forma que el proceso de desarrollado debe ser planeado y documentado, el proceso de pruebas de software debe contar con su respectiva planeación y documentación. No es descabellado pensar que un equipo de pruebas de software documente o cree más entregables que el equipo de programación mismo.

Algunos de los documentos entregables resultantes de un proceso de pruebas de software según Ron Patton<sup>2</sup>, son:

Protocolo de pruebas: describe el método general que se utilizará para verificar que el software cumple con la especificación del producto y las necesidades del cliente. Incluye los objetivos de calidad, las necesidades de recursos, los horarios, las asignaciones, los métodos, etc.

Casos de prueba: Enumeran los elementos específicos que se probarán y describen los pasos detallados que se seguirán para verificar el software.

Los informes de errores: Describen los problemas encontrados a medida que se siguen los casos de prueba. Esto podría hacerse en papel pero es recomendable hacerlo en una base de datos.

Las métricas, estadísticas y resúmenes: Transmiten el progreso que se está haciendo a medida que avanza el trabajo de prueba. Toman la forma de gráficos, cuadros e informes escritos.

**2.1.4 Protocolo de pruebas.** Un proceso de desarrollo de pruebas llevado a cabo correctamente siempre deberá incluir:

- Planificación de la prueba
- **El diseño de los casos de prueba.**
- **La ejecución de la prueba.**
- **Recolección y evaluación de resultados.**

En muchas ocasiones, el proceso de desarrollo de pruebas se reduce a ejecutar pruebas que, se supone, evalúan adecuadamente los casos de uso que se

---

<sup>2</sup> PATTON, Ron. Software Testing, First Edition. USA: Sams Publishing. 2001. p. 28(Test Documents). ISBN 0-672-31983-7

plantearon durante el diseño, sin ningún estándar que permita a todos los miembros del equipo de trabajo tener claridad acerca del proceso a seguir para llevar un control, detectar y reportar una eventual falla. Surge entonces la necesidad de establecer un protocolo que describa los pasos a seguir como parte de la prueba, cuándo planearlos y llevarlos a cabo, los recursos, tiempo y esfuerzo que van requerir. Como va a ser usado por todos los miembros del equipo, debe ser lo suficientemente flexible de tal forma que permita ser personalizado, sin que ello signifique dejar de lado una planificación y seguimiento durante el desarrollo de todo el proyecto.

*El IEEE Standard 829-1998 for Software Test Documentation<sup>3</sup> establece que el propósito de un protocolo de pruebas es: “Prescribir el alcance, el enfoque, los recursos y el cronograma de las actividades de prueba. Identificar los elementos que se están probando, las características que se probarán, las tareas de prueba que se realizarán, el personal responsable de cada tarea y los riesgos asociados con el plan.”*

Es necesario definir entonces que la primera etapa debe ser un estudio minucioso del producto a poner en prueba: El modelo que se realizó o se pretende realizar, desarrollar e implementar. Infelizmente el desarrollo de programas actualmente es un proceso que implica un número importante de modelos, métodos y fundamentos que se pasan por alto en la práctica en muchas ocasiones, o que apenas si se aplican o se entrelazan con otros. En la práctica es muy extraño encontrar un proyecto que siga fielmente un método o modelo de desarrollo. Es por eso que establecer un protocolo de pruebas inicia con la comprensión del método ideal más adecuado para cada caso y por ende, comprender a fondo el caso específico.

---

<sup>3</sup> IEE Standard Association [En línea], The IEEE Standard 829-1998 (Recuperado el 26 de febrero de 2018), Disponible en: <https://standards.ieee.org/findstds/standard/829-1998.html>

No se recomienda bajo ninguna circunstancia que la tarea de construir un protocolo de pruebas se delegue únicamente al equipo de desarrollo por la imposibilidad de que un equipo de desarrollo realice un diseño objetivo de las pruebas encaminadas a demostrar que el producto que ellos mismos realizaron, está mal. No significa entonces que el equipo de desarrollo no pueda realizar pruebas o ser partícipe de ellas, es más, siempre que se desarrolla se incluyen muchas pruebas informales y durante la ejecución del protocolo de pruebas, el equipo de desarrollo es fundamental.

**2.1.5 Informe de resultados y errores.** No es suficiente con ejecutar las pruebas y tomar nota de los resultados si ello no va a tener una repercusión en el mejoramiento del programa. Es por eso que no se puede dejar de lado el informe y análisis tanto de los casos exitosos como de los casos fallidos. Si se omite esta parte, todo el anterior trabajo no habrá tenido sentido.

En la actualidad este trabajo se facilita de sobremanera gracias a diversos métodos y herramientas que facilitan la presentación de los resultados, al igual que el reporte y seguimiento a los errores encontrados en cada una de las fases. Es fundamental que los diferentes equipos implicados en el desarrollo e implementación del programa tengan conocimiento de éstos informes, por lo cual, su presentación debe ser tan sencilla como sea posible sin dejar de lado información importante y su objetividad.

## **2.2 ESTRATEGIAS DE PRUEBAS SOFTWARE**

Por tratarse de una serie de actividades previamente planeadas para realizarse de forma sistemática, se podría decir que un protocolo de pruebas es una plantilla para las pruebas que describe los pasos con sus respectivos métodos de prueba y técnicas de diseño de casos de prueba específicos.

Para llevar a cabo estas pruebas es necesario seguir algunas estrategias que se han ido desarrollando a lo largo del tiempo para lograr un proceso más eficiente, Roger Pressman<sup>4</sup> destaca las siguientes:

- Las pruebas pueden ser realizadas por el mismo desarrollador, pero en general y especialmente en grandes proyectos se recomienda incluir un grupo de prueba independiente.
- Aunque probar sea diferente a depurar (reparar), la depuración no puede quedar por fuera de la estrategia de pruebas.
- Debido a que el plan de pruebas se ejecuta con más frecuencia sobre la fecha límite de entrega del proyecto, debe poderse medir el avance y se deben vislumbrar los problemas con la mayor brevedad posible.
- Una estrategia de pruebas debe incluir pruebas de bajo nivel para verificar que un pequeño segmento de código fuente se implementó correctamente, al igual que pruebas de alto nivel para validar las principales funciones del sistema a partir de los requerimientos del cliente.
- La prueba comienza desde los componentes o módulos más simples (pruebas unitarias) y continúa integrando cada componente hasta lograr integrar todo el sistema.

A partir de lo anterior, se definen las siguientes estrategias a la hora de diseñar el protocolo de pruebas.

**2.2.1 Pruebas Unitarias.** Se llaman de esta forma, porque durante ellas se pone a prueba cada unidad, módulo o clases de objetos individuales del programa, por considerarse, los tipos más simples del componente del software. Son pruebas en las que se debe tener especial cuidado en cubrir todas las características del objeto, como por ejemplo, poner a prueba todas las operaciones relacionadas al

---

<sup>4</sup> PRESSMAN, Roger. Ingeniería del Software, Un enfoque práctico. 7ª Edición, México: McGraw-Hill. 2002. p.384 (Enfoque estratégico para las prueba de software). ISBN 978-607-15-0314-5

objeto, verificar el valor de todos los atributos relacionados con el objeto y/o establecerlos, en caso de ser necesario a su vez se debe colocar el objeto en todos los estados posibles de tal manera que se simule todos los eventos que causen un cambio de estado.

Aquí es importante resaltar que cuando una operación es heredada, no es suficiente con probarla en la clase donde se definió y asumir que funcionará de igual forma en las subclases que la heredaron. Se debe probar en cada contexto donde se utilice de manera particular a fin de asegurarse que no hace conjeturas en otras operaciones o atributos que puedan conllevar a fallos en el sistema.

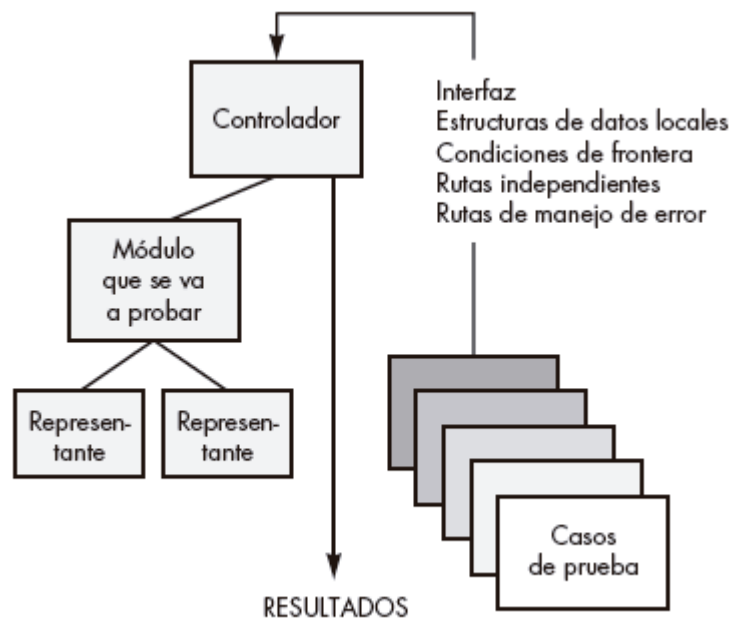
También se debe tener en cuenta que si el objeto que pretendemos probar tiene dependencia de otro objeto que aún no está terminado o que su uso pueda frenar o ralentizar el proceso de pruebas, es conveniente usar objetos simulados con el fin de agilizar las pruebas o incluso, de simular también, situaciones anormales o eventos extraños.

Las pruebas son costosas, por tanto, es importante elegir adecuadamente los casos de pruebas de unidad, aquellos que demuestren que el componente hace, lo que se supone que debe hacer, cuando se usa como se supone que se debe usar y que su uso inadecuado también se procesa adecuadamente sin colapsar el sistema. El primer componente crucial a evaluar se trata del flujo de datos por medio de la interfaz de componente, si los datos no son almacenados y recuperados adecuadamente, las demás pruebas son irrelevantes.

Dado que los componentes no son en realidad partes independientes de un programa, durante una prueba de unidad en muchas ocasiones es necesario diseñar dos tipos de programas que, aunque no se entregan en la versión final del software y no siguen lineamientos formales de desarrollo, son muy importantes durante el proceso de pruebas. Coloquialmente se puede decir que se trata de los

programas que están “antes” y “después” del módulo a probar. Entonces, se crea una interfaz que acepta datos de casos de prueba como entrada, los introduce al módulo en prueba y muestra los datos relevantes arrojados por el mismo. Este programa usualmente es conocido como “Controlador”, (ver Figura 1 Controladores y representantes en pruebas unitarias).

Figura 1 Controladores y representantes en pruebas unitarias



Fuente: PRESSMAN, Roger. *Ingeniería del Software, Un enfoque práctico*. 7ª Edición, México: McGraw-Hill. 2002. p.391 (Entorno de prueba de unidad). ISBN 978-607-15-0314-5

También se debe tener en cuenta que muchas veces los módulos dependen o necesitan invocar otros módulos durante su procesamiento, estos programas que se encargan de sustituir esos módulos se conocen como “Representantes”. Su función es la de tomar los datos que ingresa la interfaz de módulo subordinado, manipularlos mínimamente e introducirlos nuevamente al módulo (algunas veces es recomendable mostrarlos con el fin de verificar esos datos también).

Puede ocurrir que el diseño de estos controladores y representantes resulte muy dispendioso o sobrecargue el sistema. En tales casos es recomendable probar este módulo durante las pruebas de integración. La cantidad y tipo de pruebas de unidad a diseñar no pueden ser tantas que resulten muy dispendioso hacerlas todas, ni tan pocas que dejen pasar por alto algún problema.

**2.2.2 Pruebas de Integración.** También conocidas como pruebas de componentes, se les conoce de este modo porque consiste en probar el componente o conjunto de objetos que interactúan entre sí en un programa y que su funcionalidad se accede por medio de una interfaz propia de ese conjunto de componentes. Así, en estas pruebas no se evalúan los componentes como tal, sino la unión de ellos, aquellos aspectos relacionados justamente a la interacción de esos componentes. Aunque se pueda tener la noción de que el correcto funcionamiento de un componente, necesariamente implica que su unión también funcionará correctamente, esto es completamente falso. La unión de componentes presenta múltiples errores y es un factor con el cual se debe tener especial cuidado, una mala interpretación de interfaz de cada componente, puede arrojar errores fatales. El procedimiento correcto consiste en acoplar componentes poco a poco de acuerdo al diseño e ir probando el funcionamiento de este acople o integración.

En estas pruebas es común ayudarse de los casos de uso, pues ellos usan un enfoque de entrada y salida muy útil para diseñar casos de pruebas para una interfaz. Por supuesto, existen casos en los cuales se presenta interacciones entre pequeñas fallas de objetos que no son fáciles de detectar, por ejemplo, cuando hay una falla en un cálculo, pero la comunicación entre componentes aunque es aceptada, no es la correcta. Para lo anterior, se procede a simular con cálculos de los cuales tengamos resultados preconcebidos por otros métodos con el fin de comparar y hallar posibles errores. Para probar la integración de

componentes se pueden seguir algunos sencillos lineamientos que veremos a continuación:

- Hacer un listado de cada llamado a un componente externo que se encuentre en el código y a partir de este listado se diseña un conjunto de pruebas donde los parámetros dados a los componentes externos estén en los valores extremos de sus respectivos rangos porque son los que tienen mayor probabilidad de mostrar errores. De igual forma, si se trata de un puntero, se debe probar con el valor nulo.
- Diseñar pruebas que hagan fallar a los componentes que sean llamados mediante una interfaz de procedimiento.
- Para las interfaces que pasan mensajes, generar pruebas que generen más mensajes de los que se espera, se generen en la práctica con el fin de comprobar si existen problemas de temporización.

Con frecuencia es difícil establecer la forma correcta de integrar componentes y para evitar construir demasiados programas representantes y controladores (ver Figura 1 Controladores y representantes en pruebas unitarias), se integran muchos componentes haciendo difícil la detección del error cuando se presente y al realizar las pruebas de regresión aparecen nuevos errores convirtiendo la prueba en un bucle frustrante del que no siempre se logra salir fácilmente. El correcto enfoque entonces, consiste en una integración incremental en tantos pasos como sea posible. De ser posible, se debe comenzar con los módulos de menor jerarquía (aquellos que sólo van a depender de programas “controladores” y no tienen ningún programa “representante”) e ir integrando módulo a módulo. Idealmente, con cada nuevo módulo, se deben correr nuevas pruebas cuidando de que este hecho no desencadene una enorme cantidad de pruebas.

**2.2.3 Pruebas de Regresión.** Estas pruebas deben su nombre al simple hecho de que son pruebas que ya se realizaron, así que, en algún momento el equipo de

prueba se “regresa” a volver a realizar una prueba que ya se había realizado. Ahora bien, la razón de ser de este hecho es que en algunas ocasiones cuando un código es sometido a depuración o se agregan nuevos módulos, es necesario verificar que el nuevo módulo o modificación no introdujo algún tipo de error. Esencialmente, se trata de verificar que todo sigue funcionando como antes de hacer el arreglo o agregación.

Buscan descubrir cualquier tipo de anomalía como bugs, carencias de funcionalidad, divergencias funcionales respecto al funcionamiento esperado, que hayan sido causados por alguna modificación funcional o evolución.

Durante las mismas se busca comprobar que los casos de prueba exitosos lo sigan siendo, a la vez que se evalúan nuevas funcionalidades. En general, este tipo de pruebas se suelen automatizar para reducir esfuerzos y tiempo de ejecución, por lo cual se considera una buena práctica que cuando se localiza y corrige un bug, se automatice una prueba que ponga al descubierto el error para correrla posteriormente una vez se corrija el error o se haga una nueva modificación.

Roger Pressman<sup>5</sup> considera que un subconjunto de pruebas de regresión deben estar compuestas por:

- Una muestra representativa de pruebas que ejercitará todas las funciones de software.
- Pruebas adicionales que se enfocan en las funciones del software que probablemente resulten afectadas por el cambio.
- Pruebas que se enfocan en los componentes del software que cambiaron.

---

<sup>5</sup> PRESSMAN, Roger. Ingeniería del Software, Un enfoque práctico. 7ª Edición. México: McGraw-Hill. 2002. p.398 (Pruebas de regresión). ISBN 978-607-15-0314-5

Visto de este modo, parece que en realidad la cantidad de pruebas de regresión van a terminar siendo tediosas e imprácticas. Para evitar esto, se debe contar con el suficiente criterio que permita identificar aquellas pruebas que evalúen la mayor cantidad posible de errores en el programa principal, al igual que el momento adecuado para realizar nuevamente cada conjunto de pruebas. No es necesario correr pruebas de regresión para cada cambio, pero tampoco se pueden realizar muchos cambios antes de volver a correrlas, ya que sería más difícil identificar la modificación que introdujo el error.

**2.2.4 Pruebas de Sistema.** Esta etapa se considera la parte final de las pruebas funcionales y da lugar a las pruebas no funcionales. En ella se integran todos los paquetes de módulos que se venían ya probando en las dos anteriores fases y se prueba la interacción entre paquetes con el sistema completamente integrado. Algunos autores consideran que, en las pruebas de integración ya debió haberse realizado la prueba de integración de todo el sistema de tal forma que en estas pruebas no solamente se prueba si la integración de todos los componentes desarrollados es compatible y funciona de acuerdo a lo esperado, sino que también tiene en cuenta la integración del sistema con otros elementos o sistemas, personas, hardware... etc. Es importante resaltar que estas pruebas, a diferencia de las pruebas de componentes, pueden integrar componentes reutilizables desarrollados por separado y sistemas comerciales, con componentes desarrollados recientemente. Es una etapa más colectiva donde se integran módulos desarrollados por diferentes miembros del equipo o inclusive grupos, en ocasiones se excluyen programadores y diseñadores, siendo desarrollada por un equipo de prueba independiente que estudia y evalúa el comportamiento emergente de la funcionalidad del sistema que se hace presente únicamente durante ésta integración. Este comportamiento muchas veces es indeseado o no había sido planeado en etapas anteriores, pues solo se hace presente durante esta integración porque en muchos casos el desarrollador o equipo de pruebas nunca consideró el enfoque de uso que uno o varios usuarios dieron al sistema.

Es difícil establecer cuantas pruebas de sistemas se deben realizar, las pruebas exhaustivas, además de imposibles, pueden llevar a un sobre costo innecesario. De tal forma que cada empresa suele tener una política al respecto dependiendo de las necesidades y prioridades del software desarrollado. También es importante basarse en la experiencia de uso del sistema con el fin de predecir aquellas interacciones que se consideren más críticas a las cuales tendría que prestársele especial atención.

Las pruebas automatizadas de sistemas son posibles, aunque son mucho más complejas que las de unidad o de componentes ya que en este caso no es tan sencillo predecir una salida y en ocasiones puede decirse que es imposible, por lo cual la salida sencillamente se debe examinar y demostrar que está de acuerdo a lo esperado sin contar con un referente ya obtenido por otro método.

**2.2.5 Pruebas no funcionales.** Se denominan de esta forma al conjunto de pruebas que evalúan los atributos no funcionales de un aplicativo como la seguridad, recuperación, despliegue, esfuerzo y rendimiento, entre otras.

2.2.5.1 Pruebas de seguridad. Este tipo de pruebas requieren de diversas actividades, todas ellas encaminadas y pensadas en función de asegurar tanto la disponibilidad como el funcionamiento de los recursos y servicios prestados por el sistema. Son cuidadosamente diseñadas de acuerdo al tipo de sistema diseñado, la información que maneje, la importancia de ciertos servicios, el nivel de confidencialidad e impacto de los recursos y servicios... etc. Aunque un sistema solamente posea información de dominio público, esto es, no incluya información de seguridad institucional o estratégica, va a requerir de unos mínimos de cuidados.

En las pruebas de seguridad, todo se vale a la hora de intentar penetrar en el sistema, de tal forma que durante el diseño de éstas se pueden intentar tantos

métodos como se consideren pertinentes sin que ello implique excesivos sobrecostos. Con el desarrollo de la ingeniería social, en ocasiones realizar pruebas de seguridad implica también probar la parte humana, ya que en muchos sistemas con excelentes diseños de seguridad, son vulnerados aprovechando la credulidad o ignorancia de los usuarios. Garantizar la seguridad de un sistema es virtualmente imposible, una buena prueba de seguridad penetrará el sistema. Entonces la tarea del diseñador y del equipo de pruebas es garantizar que el costo de penetrar en el sistema sea más alto que la información que se obtendrá.

2.2.5.2 Pruebas de rendimiento. Este tipo de pruebas permiten medir la respuesta del sistema ante un volumen de carga determinado que se considera dentro del uso típico esperado del sistema. Como su nombre lo indica, en este tipo de pruebas se busca medir el rendimiento del sistema en condiciones normales de operación.

Para probar los requerimientos es usual crear un perfil operativo, que consiste esencialmente en crear un perfil que refleje el tipo de carga que manejará mi sistema y en qué porcentajes, es decir, determinar cuáles son aquellos tipos de transacciones que manejará con mayor frecuencia mi sistema y cuales con menor, de tal suerte que el perfil operativo debe considerar más pruebas para las transacciones de mayor porcentaje y consecuentemente, menores pruebas para aquellas transacciones que tendrán poca demanda.

Por lo anterior, en algunas ocasiones es necesario definir una métrica diferente para aquellas transacciones que requieren de mayor concurrencia o un mayor volumen de trabajo en comparación con tareas sencillas.

2.2.5.3 Pruebas de carga. Este tipo de pruebas permiten medir la respuesta del sistema ante distintos volúmenes de carga, definidos como bajo, normal y alto según las métricas de los requerimientos o las que el equipo de desarrollo defina

como garantía y el equipo de calidad hayan considerado aceptables. Por ejemplo, se medirá la respuesta del sistema ante 100, 500 y 1000 transacciones y/o usuarios de forma concurrente. Los resultados se deberán comparar con las métricas establecidas las cuales pueden variar de acuerdo a la versión del sistema en la medida que el mismo se desarrolla, por tanto, se establecen generalmente antes de iniciar las pruebas.

Esta parte muchas veces es omitida o evaluada de forma muy ligera haciendo que el sistema, aunque cumpla con todos los demás requerimientos, no se vea reflejado una vez puesto en producción. Una vez integrado totalmente el sistema, es importante tener certeza acerca de la carga que puede desempeñar mi sistema y si esa carga está de acuerdo a lo que se espera del sistema por parte de los usuarios o clientes.

Muy importante también tener presente que a la hora de ejecutar este tipo de pruebas no basta con cargar el sistema de forma concurrente en un determinado instante, sino mantenerlo cargado por un periodo considerablemente largo de tiempo. Es decir que si mi sistema debe soportar 1.000 transacciones en simultáneo, durante el diseño de las pruebas se deben configurar 60.000 transacciones en 60 segundos, lo que dejará en promedio 1.000 usuarios por segundo.

2.2.5.4 Pruebas de usabilidad. Aunque un sistema técnicamente perfecto, de nada va a servir si es tan poco entendible que su uso se torna muy difícil o casi imposible. Es por eso que este tipo de pruebas toman especial importancia y no se pueden dejar de lado. Las pruebas de usabilidad buscan establecer cuán intuitivo es el uso del sistema creado. Aquí se establece si el sistema comunica lo que se esperaba y en general si la relación con los usuarios es amigable.

Por tratarse de un aspecto un tanto subjetivo, usualmente se realiza mediante la aplicación de encuestas a grupos de posibles usuarios del sistema evaluado. De acuerdo del tipo de sistema pueden llegar a resultar bastante exhaustivas y se debe contar con el suficiente criterio para mantener la objetividad. Es recomendable que estas pruebas se realicen en varias etapas de vida del software con el fin de retroalimentar el funcionamiento o rediseño del mismo.

## **2.3 HERRAMIENTAS PARA DOCUMENTACIÓN Y AUTOMATIZACIÓN DE PRUEBAS SOFTWARE**

**2.3.1 Apache JMeter.** Jmeter es esencialmente una herramienta de prueba de carga que permite medir y analizar el desempeño de diversos servicios o recursos tanto estáticos como dinámicos (bases de datos y consultas, servidores FTP, archivos, servlets...etc).<sup>6</sup>

Puede ser usado para probar la robustez de una red u objeto, simular diferentes tipos de carga pesada en un servidor con el fin de analizar gráficamente o simplemente probar el comportamiento y rendimiento integral del mismo, entre otras funcionalidades.

Por tratarse de un framework multitarea, es posible realizar muestreos simultáneos de muchos procesos y un muestreo de las diversas funciones de grupos de procesos por separado, además de permitir un almacenamiento y análisis de los resultados de las pruebas sin que ellas tengan conexión en sus diferentes repeticiones.

Su descarga e instalación es muy sencilla. Se debe ingresar a la página del autor: [jmeter.apache.org](http://jmeter.apache.org) desde cualquier navegador. Seguir las recomendaciones de uso al igual que los requisitos de instalación y descarga para obtener el archivo

---

<sup>6</sup> Apache Jmeter Application[En línea], The Apache Software Foundation 2018, (Recuperado el 23 de Febrero de 2018) Disponible en: <https://jmeter.apache.org/index.html>

correspondiente al sistema operativo de la máquina que se pretenda usar para realizar las pruebas. En la misma página del autor al igual que en diversos foros se encuentra amplia información sobre su manejo y uso de acuerdo a su versión.

**2.3.2 BugZilla.** Bugzilla “Bug Tracking System” es, como su nombre lo indica, un sistema de seguimiento de errores para un sistema en prueba. Es fácil de usar y tiene una funcionalidad de detección de errores doblemente reportados que lo hace muy popular.

Permite además de registrar los defectos identificados en las pruebas, describirlos y anexar documentación de soporte con opción de ser enviados a todo el equipo de desarrollo. También presenta reportes que permite ver cuántos errores tiene en cada estatus y posee una licencia gratuita<sup>7</sup>.

**2.3.3 Selenium.** Selenium es una suite para realizar pruebas automatizadas gratuita y de código abierto para aplicaciones web en diferentes plataformas y navegadores. Es en esencia, un conjunto de herramientas que permiten desarrollar scripts para pruebas automatizadas en aplicaciones basadas en la web en lenguajes como Java, Ruby, Python, Perl, .Net o PHP.(Selenese).<sup>8</sup>

El desarrollo del presente protocolo, se sugiere el uso del componente “*Selenium IDE*”, un plugin para los navegadores más comunes como Google Chrome o Mozilla Firefox, que provee una herramienta para crear, grabar y reproducir pruebas sin la necesidad de usar un lenguaje de scripting para pruebas. Este plugin provee comandos que ejecutan acciones directamente sobre objetos en el navegador como seleccionar de una lista de opciones, dar click, escribir...etc. Con lo cual se pueden ejecutar casos o conjuntos de casos de prueba.

---

<sup>7</sup> Home Bugzilla [En línea], About Bugzilla 2018, (Recuperado el 23 de marzo de 2018) Disponible en: <https://www.bugzilla.org/about/>

<sup>8</sup> Selenium IDE [En línea]. Herramienta para automatizar pruebas de software, (Recuperado el 23 de Febrero de 2018) Disponible en: <https://www.seleniumhq.org/docs/>

Permite almacenar sus pruebas y resultados, al igual que reproducirlas de forma automática tantas veces como sea necesario. Su instalación es muy sencilla, basta con dirigirse a la página del fabricante: [https://www.seleniumhq.org/download/#selenium\\_ide](https://www.seleniumhq.org/download/#selenium_ide) y seleccionar del listado el navegador con el cual se esté trabajando. Seguir las instrucciones y dar los permisos necesarios para que el plugin se ejecute en el navegador. Existe también amplia información sobre su uso y configuración en la web.

**2.3.4 Testlink.** Testlink es una herramienta de pruebas basado en la web, diseñado para facilitar la garantía del software. Ofrecer soporte para casos de prueba, suites de prueba, planes de prueba, proyectos de prueba y administración de usuarios, así como varios informes y estadísticas.<sup>9</sup>

De esta manera, se asignan pruebas al personal y se registran los resultados que al final son presentados en una gran variedad de informes mostrando lo que se ha realizado hasta el momento. De gran utilidad por permitir la creación de diferentes roles con distintos permisos, algo bastante conveniente en nuestro caso. Además, permite la creación ilimitada de testsuites para organizar mejor los casos de prueba. Es de resaltar que permite realizar un versionado de los casos de prueba.

**2.3.5 Gt Metrix.** Se trata de una herramienta en línea muy intuitiva y sencilla que evalúa en unos segundos diferentes características del frontend, muestra un informe detallado y sus respectivas sugerencias encaminadas a mejorar el rendimiento del sitio web. Fue desarrollado por GT.net como una herramienta destinada a mejorar el rendimiento de los sitios web de sus clientes alrededor del mundo.<sup>10</sup>

---

<sup>9</sup> Testlink home [En línea], TestLink Open Source Test Management (recuperado el 23 de Febrero de 2018) Disponible en: <https://github.com/TestLinkOpenSourceTRMS/testlink-documentation>

<sup>10</sup> GT Metrix [En línea], Website Speed and Performance Optimization, (Recuperado el 20 de Marzo de 2018) Disponible en: <https://gtmetrix.com/about>

Su uso es extremadamente simple, basta con ingresar a la página <https://gtmetrix.com/> e ingresar la dirección del sitio web a analizar en el campo destinado para tal fin. El programa tardará unos segundos en generar un informe a partir del análisis realizado donde se muestran datos básicos de performance inicialmente y debajo de ellos un informe con estadísticas y sugerencias para optimizar el rendimiento.

**2.3.6 MantisBT.** Es una herramienta de código abierto diseñada para hacer seguimiento de problemas e incidencias, destacándose por ser muy simple sin dejar de lado su capacidad funcional de acuerdo a sus creadores<sup>11</sup>, lo que permite aprender a usarla en sólo segundos por parte de cualquier miembro del equipo de colaboradores y clientes. Puede ser usada en cualquier sistema que soporte PHP, además, puede trabajar con los más populares motores de bases de datos y proporciona servicios web que le permiten interacción con otras aplicaciones.

También destaca la posibilidad que tiene de incluir la descripción, archivos adjuntos y configurar envíos de e-mail automáticos para cada incidencia, además de diversos plugins y la posibilidad de integración con otras herramientas como testlink. Para descargarlo se debe ir a la página del autor: <https://www.mantisbt.org/> y buscar la sección de descargas desde donde se busca la última versión estable liberada.

---

<sup>11</sup> MantisBT [En línea], Mantis Bug Tracker, 2018 (Recuperado el 23 de Marzo de 2018) Disponible en: <https://www.mantisbt.org/>

### **3. GENERALIDADES DEL SOPORTE INFORMÁTICO MIE**

#### **3.1 CONTEXTUALIZACIÓN DEL PROYECTO**

“La Maestría en Informática para la Educación” en adelante “MIE-UIS”, es un programa de profundización en la modalidad a distancia (trabajo presencial y virtual soportado en plataformas tecnológicas). “Es un posgrado orientado por el enfoque pedagógico problémico, dirigido a docentes de educación pre-escolar, básica y media y otros profesionales interesados en formarse como docentes investigadores con capacidad para indagar, reflexionar y actuar en su contexto educativo, incorporando de forma creativa los recursos informáticos disponibles.”

12

Con el fin de brindar apoyo al anterior propósito, surge la necesidad de crear una herramienta informática desarrollada a la medida para responder a las necesidades específicas de la ejecución del programa de postgrado denominado “Soporte Informático para la Maestría en Informática para la Educación” en adelante “Soporte Informático MIE-UIS”. Para garantizar la calidad de este producto software, se plantea la creación del presente protocolo de pruebas que busca validar las propiedades funcionales y no funcionales del sistema.

#### **3.2 PROCESO PLANTEADO DE DESARROLLO DEL SOPORTE INFORMÁTICO MIE-UIS**

##### **3.2.1 Equipo requerido.**

- Analista de Software, de preferencia con formación de posgrado y experiencia en levantamiento de requerimientos.

---

<sup>12</sup> Escuela de Ingeniería de Sistemas e informática de la Universidad Industrial de Santander [En línea], Programas de maestría. (Recuperado el 20 de febrero de 2018), disponible en <http://cormoran.uis.edu.co/eisi/eisi.jsp?IdServicio=S78>

- Desarrollador de software Senior, de preferencia con formación de posgrado y experiencia en desarrollo de software.

Diseñador gráfico, con conocimientos y experiencia en diseño de interfaces de usuario de sistemas software web

### **3.2.2 Proceso de desarrollo**

- Análisis y diseño: Consiste en la obtención y validación de requerimientos (Desde el perfil de administrador, coordinador del programa, tutor y del estudiante), modelado de la lógica del negocio-proceso, diseño por módulos y selección de tecnologías de desarrollo e implementación.
- Desarrollo: Consiste en la configuración del ambiente de desarrollo, configuración del ambiente de ejecución (servidor), construcción de la base de datos y desarrollo por módulos.
- Implementación del sistema: Consiste en la instalación en ambiente de ejecución (servidor), instalación por módulos desarrollados, pruebas de funcionalidad, pruebas con usuarios, reporte de errores y desarrollo de manuales (técnico y de usuario)
- Mantenimiento: Consiste en la corrección de errores, pruebas finales y entrega a usuario final.

## **3.3 DISEÑO PLANTEADO DEL SISTEMA**

### **3.3.1 Usuarios del sistema**

- Tutores: Son los usuarios a cargo de guiar las diferentes actividades de los cursos programados. Para guiar las actividades los tutores cuentan con el soporte del sistema que les permite interactuar con los estudiantes y gestionar los recursos usados en las actividades.

- **Estudiantes:** Son los usuarios que están en el proceso de aprendizaje con los tutores y se valen del soporte del sistema para realizar las actividades asignadas y presentar los entregables solicitados por los tutores.
- **Auxiliares:** Los auxiliares son usuarios que ayudan a tutores y administradores a realizar algunas tareas que pueden en el sistema.
- **Coordinador del programa:** El coordinador del programa es un usuario con determinados permisos de acceso y publicación en el sistema.
- **Administrador del sistema:** El administrador del sistema es el usuario que tiene acceso a todas las funciones del sistema y es el encargado de la creación y gestión de los usuarios del sistema, las asignaturas, los grupos de asignaturas, los módulos, actividades, recursos y las demás opciones que tiene el sistema. Es el único usuario con permisos para crear, modificar, eliminar, buscar, activar o desactivar un rol, perfil o curso. Los demás usuarios sólo podrán actualizar sus datos.

**3.3.2 Requerimientos funcionales.** Los requisitos funcionales planteados en el diseño del sistema son los enunciados en la Tabla 1 Requerimientos funcionales del sistema:

Tabla 1 Requerimientos funcionales del sistema

<b>Código</b>	<b>Nombre</b>
RQ01	<b>Difusión información del programa y registro de candidatos.</b> Se debe permitir la publicación de información del programa y el registro de las personas interesadas en el programa con el objetivo de brindarles información del programa y apoyo en el proceso de ingreso y matrícula
RQ02	<b>Gestión de perfil (usuarios):</b> Se debe permitir a los usuarios creados previamente el sistema registrar la información del perfil requerida (formación profesional, experiencia, etc.), adicionalmente se debe permitir opciones de recuperación de nombre de usuario y contraseña
RQ03	<b>Crear estructura de asignaturas.</b> Se debe permitir la creación de la estructura de las asignaturas del programa.

RQ04	<b>Modificar estructura de asignatura.</b> Se debe permitir la modificación de la estructura conductual de las asignaturas del programa creadas.
RQ05	<b>Creación de asignatura.</b> Se debe permitir la creación de una asignatura, ingresando todos los datos pertinentes.
RQ06	<b>Modificación de asignatura.</b> Se debe permitir la modificación de los datos de una asignatura creada previamente
RQ07	<b>Eliminación de asignatura.</b> Se debe permitir la eliminación de una asignatura, siempre y cuando no existan restricciones para su eliminación
RQ08	<b>Creación de módulo.</b> Se debe permitir la creación de un módulo asociado a una asignatura existente, registrando los datos requeridos.
RQ09	<b>Modificación de módulo.</b> Se debe permitir la modificación de los datos de un módulo asociado a una asignatura
RQ10	<b>Eliminación de módulo.</b> Se debe permitir la eliminación de un módulo registrado a una asignatura, siempre y cuando no existan restricciones para su eliminación.
RQ11	<b>Creación de actividades.</b> Se debe permitir la definición de actividades asociadas a un módulo, ingresando los datos requeridos. (Nombres, fechas, etc).
RQ12	<b>Modificación de actividades.</b> Se debe permitir la modificación de las actividades asociadas a un módulo, creadas previamente.
RQ13	<b>Eliminación de actividades.</b> Se debe permitir la eliminación de las actividades asociadas a un módulo, siempre y cuando no existan restricciones para su eliminación
RQ14	<b>Creación de recursos.</b> Se debe permitir la creación de recursos de acuerdos a tipos establecidos
RQ15	<b>Modificación de recursos.</b> Se debe permitir la modificación de recursos creados previamente
RQ16	<b>Eliminación de recursos.</b> Se debe permitir la eliminación de recursos siempre y cuando no existan restricciones para su eliminación
RQ17	<b>Creación, modificación o eliminación de recursos producidos por estudiantes.</b> Se debe permitir la creación, modificación o eliminación de recursos producidos por estudiantes.
RQ18	<b>Consultar recursos producidos por estudiantes.</b> Se debe permitir la consulta de los recursos creados por estudiantes por parte de todos los estudiantes del programa.
RQ19	<b>Generar informe de ejecución de la asignatura.</b> El sistema debe generar un informe de las novedades presentadas durante la ejecución de una asignatura.

RQ20	<b>Creación y modificación de calendarios del programa.</b> Se debe permitir la creación y modificación de calendarios académicos del programa académico para consulta de los estudiantes, docentes y administrativos
RQ21	<b>Creación de usuarios del sistema (Administrador).</b> Se debe permitir la creación de usuarios en el sistema con toda la información pertinente de acuerdo al rol (Estudiantes, profesores)
RQ22	<b>Gestión de los roles de usuario (Administrador).</b> Se debe permitir la asignación y modificación de los roles de los usuarios registrados en el sistema
RQ23	<b>Administración de cursos (Usuarios).</b> Se debe permitir la asignación de usuarios a cursos de acuerdo a la matrícula
RQ24	<b>Administración de cursos (Contenidos).</b> Se debe permitir la creación, modificación y eliminación de los contenidos de apoyo de los cursos creados. (Herramientas, contenidos, información, etc.)
RQ25	<b>Administración de publicación.</b> Se debe permitir la creación, modificación y eliminación de las publicaciones del programa (Información del programa, contacto, marketing, etc.)
RQ26	<b>Administración de la seguridad.</b> Se debe permitir realizar el respaldo manual y automático de la base de datos y los contenidos de los usuarios.

**3.3.3 Requerimientos no funcionales.** Entre los requisitos no funcionales que plantea el diseño del sistema están:

- Documentación (ayuda dentro de la aplicación).
- Eficiencia.
- Rendimiento.
- Usabilidad.

### **3.4 DISEÑO DESARROLLADO DEL SISTEMA**

Aunque la presentación de la página varía de acuerdo al tipo de usuario, a continuación se presenta el diseño realizado del sistema con cada uno de los menús posibles:

**3.4.1 Asignatura.** En este menú se pretenden agrupar todas las posibles acciones a realizar sobre las asignaturas. Éstos son los submenús que se despliegan y su respectiva descripción:

- Propósitos de asignatura: Desde este submenú se puede ingresar, editar o eliminar propósitos de asignatura. Sencillamente se debe enunciar el propósito en menos de 1500 caracteres y darle guardar.
- Competencia de asignatura: Desde este submenú se puede ingresar, editar o eliminar competencias de asignatura. Para poder crear una nueva competencia, es necesario haber creado previamente un tipo de competencia donde se catalogará la nueva competencia creada. Se introduce la descripción y el tipo de competencia la que pertenece para poderse guardar.
- Tipos de competencia de asignatura: Desde este submenú se puede ingresar, editar o eliminar tipos de competencia de asignatura (Actitudinales, procedimentales, cognitivos...etc.). Sencillamente solicita el nombre en menos de 300 caracteres.
- Asignatura (Curso Matriz): Desde este submenú se puede crear, editar o eliminar una asignatura (Curso Matriz). Hay que tener en cuenta que cada asignatura se subdivide en módulos y a su vez, cada módulo puede ser dictado por un tutor diferente. Para crear una nueva asignatura, es necesario haber creado el propósito antes. También cuenta con campos que se deben diligenciar como el nombre y la justificación. (Existen otros campos, pero no se definen como obligatorios).
- Asignar competencias a una asignatura. Es aquí donde se crea el enlace entre las asignaturas y competencias previamente creadas las dos.
- Grupo de asignatura: Desde este submenú se puede crear, editar o eliminar los grupos de asignatura. Para crear un nuevo grupo es necesario haber creado anteriormente la asignatura a la cual se va a asignar ese grupo.

**3.4.2 Modulo.** En este menú se pretenden agrupar todas las posibles acciones a realizar sobre los módulos. Éstos son los submenús que se despliegan y su respectiva descripción:

- Tipo de temáticas para módulo: Desde este submenú se puede ingresar, editar o eliminar el tipo de temáticas para un módulo.
- Temáticas de módulo: Desde este submenú se puede ingresar, editar o eliminar las temáticas de un módulo. Para poder crear una nueva temática, es necesario haber creado anteriormente el tipo de temática a la que corresponde la temática a crear.
- Estrategias de enseñanza y aprendizaje: Desde este submenú se puede ingresar, editar o eliminar estrategias de enseñanza y aprendizaje (estructura).
- Módulos de asignatura creada: Desde este submenú se puede crear, editar o eliminar módulos de una asignatura existente. Para poder crear un nuevo módulo, es necesario que ya haya sido creada la asignatura y temática correspondiente a ese módulo.
- Asignar tutor a módulo: Esta opción enlaza cada uno de los módulos con el respectivo tutor encargado, teniendo en cuenta que las asignaturas se dividen en módulos, cada uno de los cuales puede ser atendido por un tutor diferente, con lo cual, una misma asignatura puede tener varios módulos al igual que varios tutores.

**3.4.3 Actividad.** En este menú se pretenden agrupar todas las posibles acciones a realizar sobre las actividades, los submenús que se despliegan y su respectiva descripción son:

- Tipo de actividad para módulo: Desde este submenú se puede crear, editar o eliminar los tipos de actividades para un módulo.
- Desde este submenú se puede crear, editar o eliminar actividades de estrategia de enseñanza y aprendizaje: Para poder crear una nueva actividad,

es necesario haber creado anteriormente el tipo de actividad, módulo y estrategia de enseñanza-aprendizaje respectiva.

- Asignar recurso a actividad. Aquí se enlazan los recursos con su respectiva actividad, por lo cual es necesario que exista ya tanto el recurso como su respectiva actividad.

**3.4.4 Recursos.** En este menú se pretenden agrupar todas las posibles acciones a realizar sobre los recursos. Éstos son los submenús que se despliegan y su respectiva descripción:

- Tipo de recurso: Desde este submenú se puede administrar los tipos de recursos con los que se van a trabajar durante las actividades.
- Recursos: Desde este submenú se puede crear, editar o eliminar los recursos. Para poder crear un recurso nuevo, es necesario haber creado el tipo de recurso donde se catalogará en el momento de su creación.
- Foros: Desde este submenú se puede crear, editar o eliminar los foros que servirán de canal de retroalimentación en el proceso. Así mismo, se pueden vincular o desvincular participantes a cada foro al igual que visualizar los ya participes.

**3.4.5 Usuarios.** En este menú se pretenden agrupar todas las posibles acciones a realizar sobre los usuarios, éstos son los submenús que se despliegan y su respectiva descripción:

- Usuarios: Desde este submenú se puede crear, editar o eliminar usuarios. Aquí se ingresa toda la información de cada usuario sin importar el rol que vaya a desempeñar.
- Rol de usuario: Desde este submenú se puede crear, editar o eliminar rol de usuario del sistema: En esta opción se crea el nombre de cada futuro rol de usuario.

- **Asignar rol de usuario:** Desde este submenú se puede asignar o editar rol de usuario a un usuario. Aquí se enlaza el usuario creado en el primer submenú con el rol creado en el segundo submenú.
- **Datos de tutores:** Desde este submenú se puede ingresar o editar los datos de los tutores. A aquellos usuarios que se les haya definido anteriormente con el rol de “tutor”, aquí se debe agregar su correspondiente información adicional como su formación, horario y sede a la que pertenece.
- **Instituciones:** Desde este submenú se puede ingresar, editar o eliminar instituciones donde los estudiantes desarrollarán sus actividades.
- **Datos de estudiantes:** Desde este submenú se puede ingresar o editar datos de los estudiantes. A aquellos usuarios que se les haya definido anteriormente con el rol de “estudiantes”, aquí se debe agregar su correspondiente información adicional como su formación, experiencia, institución donde labora e institución donde ha propuesto trabajar su proyecto.
- **Asignar estudiante a grupo:** Desde este submenú se puede asignar estudiante a grupo correspondiente de materia.

**3.4.6 Menú.** En este menú se pretenden agrupar todas las posibles acciones a realizar sobre los menús. Éstos son los submenús que se despliegan y su respectiva descripción:

- **Opciones de menús secundarios:** Desde este submenú se puede crear, editar o eliminar las opciones de menús secundarios.
- **Asignar opciones de menú a roles de usuario:** Desde este submenú se pueden definir los menús y submenús a los cuales un tipo de usuario puede acceder.

**3.4.7 Salir.** En esta opción de menú es posible cerrar la sesión de forma segura, dando clic en el botón del submenú: “Salir”.

## 4. DISEÑO PROTOCOLO DE PRUEBAS

### 4.1 GENERALIDADES DEL PROTOCOLO

El presente protocolo está diseñado para que, con su aplicación, se evalúen atributos funcionales y no funcionales del sistema por parte de un equipo de calidad, es decir, aunque un equipo de desarrollo puede hacer uso del presente protocolo, está orientado al equipo de calidad. Se realizará una validación (¿Estamos construyendo el producto correcto?) de caja negra y mediante una estrategia convencional de pruebas de software para determinar si las principales características de la plataforma MIE-UIS cumple con los requisitos de calidad necesarios.

Se deberá contar en todo momento con un ambiente de pruebas (ambiente aislado) que funcione exactamente igual al ambiente de producción (ambiente en servicio), pero que no tenga relación que pueda entorpecer este último. Como entregables del proceso de pruebas, se generan por parte de los usuarios que prueban y de acuerdo a lo contemplado en el presente documento, los documentos con los resultados de las pruebas, sugerencias, recomendaciones, etc.

**4.1.1 Roles del equipo de calidad.** El equipo de calidad deberá contar con los siguientes roles con sus respectivas funciones:

- **LIDER:** Como todo de equipo, se requiere de un líder que será el encargado de verificar y guiar el proceso de validación de acuerdo al protocolo y documentación.

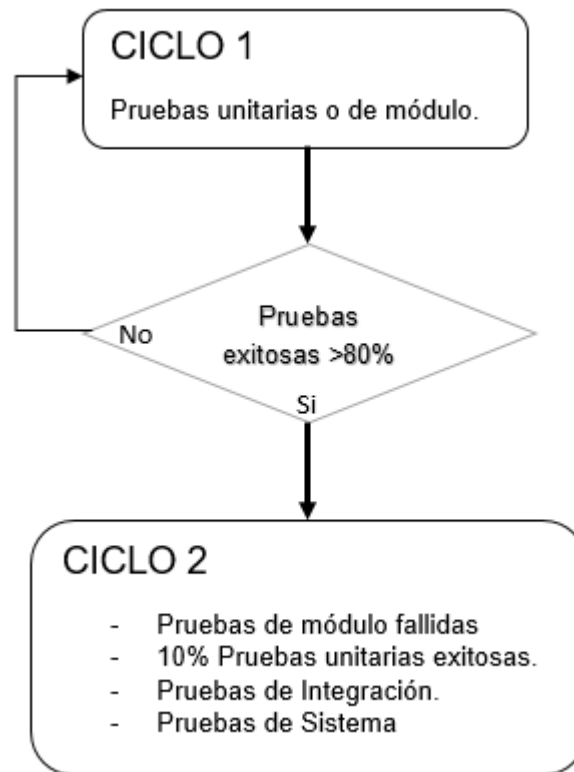
- **DISEÑADOR DE CASOS DE PRUEBA:** Se encarga de diseñar los casos de prueba siguiendo las pautas del presente protocolo y aquellas que el líder considere.
- **EJECUTOR DE CASOS DE PRUEBA:** Es el encargado de llevar a cabo los casos de prueba diseñados siguiendo fielmente las indicaciones en ellos descritas. También es la persona que registrar las incidencias en caso de presentarse.

En caso de que la prueba que se pretenda llevar a cabo no sea exhaustiva, los tres anteriores roles podrán ser desempeñados por una o dos personas que cuenten con los conocimientos y capacidades para ellos. De forma análoga, en caso de que se requiera, podrán contar con varios diseñadores y/o ejecutores de casos de prueba según considere el líder del equipo de calidad de acuerdo al presupuesto y tiempo destinados.

**4.1.2 Tipos de pruebas y ciclos.** Las pruebas serán divididas en dos tipos: las funcionales y las no funcionales. Dentro de las pruebas funcionales se definen pruebas unitarias, de integración, de sistema y de regresión. Para su realización se establecen dos ciclos de ejecución. El primer ciclo constituye únicamente las pruebas unitarias. Un segundo ciclo se encarga de evaluar las pruebas de integración, de sistema y de regresión.

Las pruebas de regresión constituyen todas aquellas pruebas que se consideren fallidas durante la primera fase más un diez por ciento de las que se hayan considerado exitosas con el fin de comprobar que los ajustes realizados no hayan inducido nuevos errores, a la vez que identifica aquellos errores que ya se corrigieron desde el primer ciclo.

Figura 2 Tipos de prueba y ciclos



Es importante tener en cuenta que para que llegar a un segundo ciclo es necesario que el 80% de las pruebas ejecutadas en el primer ciclo se consideren aprobadas. En caso contrario, se debe repetir el primer ciclo hasta que se cumpla la condición.

Las pruebas no funcionales, por su parte, se ejecutarán en un único ciclo.

**4.1.3 Formato para el diseño de casos de prueba funcionales.** Durante el diseño de casos de prueba o test cases funcionales, se define el siguiente formato:

Tabla 2 Formato para el diseño de casos de prueba funcionales.

<b>Código</b>	<b>Los casos de prueba se organizan de acuerdo al consecutivo: MIE-TCx.A.B.C, donde x será reemplazado por U, I o S de acuerdo al tipo de prueba a desarrollarse (Unitarias, Integración o Sistema). A corresponde al número de paquete, B corresponde a la posición del módulo dentro del paquete y C es el número de caso de prueba.</b>	
<b>Nombre</b>	Nombre del caso de prueba a ejecutar	
<b>Descripción</b>	Se enuncia una breve descripción que amplíe la información sobre el caso de prueba a realizarse.	
<b>Paquete-Módulo</b>	Aquí se relaciona el paquete y módulo al que corresponde la funcionalidad bajo prueba.	
<b>Dependencias</b>	Aquí se enuncian aquellos módulos de los cuales dependa el módulo en prueba.	
<b>Actores</b>	Se establece los actores que podrán probar el módulo. Esto, a partir de los requerimientos iniciales. Idealmente se debe comprobar que cada uno de los actores enunciados aquí, en efecto, tengan los permisos suficientes para llevar a cabo la prueba. Por tanto, los actores que no estén aquí definidos, no deberían tener acceso al módulo en la plataforma. Es un aspecto muy importante que se debe ir comprobando en la medida que se avanza en cada módulo.	
<b>Pre condiciones</b>	En esta casilla se deben establecer las condiciones iniciales que deben darse para que la prueba pueda ser llevada a cabo en el módulo en cuestión. No es necesario definir aspectos triviales o comunes a todos los módulos como el navegador, la conexión a internet, el acceso a un computador... etc. Pero si se debe definir aquellos aspectos de los que dependa la prueba, por ejemplo, para crear una asignatura es necesario haber definido antes un propósito de asignatura, haber ingresado como administrador o tutor... etc.	
<b>Post condiciones</b>	Aquí se debe establecer las condiciones de salida de la prueba, es decir, el estado esperado del sistema una vez se realiza la prueba.	
<b>Escenario principal</b>	<b>Actores</b>	<b>Sistema</b>
	En estas casillas se define el paso a paso de lo que debe realizar el actor.	En estas casillas se define el paso a paso de lo que debe realizar el sistema.
	También se define el comportamiento esperado por parte de los actores o actor implicado en el caso de prueba.	También se define el comportamiento esperado por parte del sistema durante el caso de prueba.
	Se puede agregar o eliminar casillas según requiera cada uno de los casos de prueba específicos.	
<b>Observaciones</b>	En este espacio se pueden realizar las observaciones a las que hubiere lugar y que se consideren de importancia para el desarrollo del caso de prueba.	

**4.1.4 Formato para el diseño de casos de prueba no funcionales (rendimiento y carga).** Durante el diseño de casos de prueba o test cases no funcionales, se define el siguiente formato:

Tabla 3 Formato para el diseño de casos de pruebas no funcionales.

<b>Código</b>	<b>Los casos de prueba se organizan de acuerdo al consecutivo: MIE-TC.A.B.C, donde A puede variar entre "C" para carga y "R" para rendimiento, B corresponde al número de usuarios concurrentes y C es el número de caso de prueba.</b>
<b>Nombre</b>	Nombre del caso de prueba a ejecutar
<b>Descripción</b>	Se enuncia una breve descripción que amplíe la información sobre el caso de prueba a realizarse.
<b>Paquete-Módulo</b>	Aquí se relaciona el paquete y módulo al que corresponde la funcionalidad bajo prueba.
<b>Pre condiciones</b>	En esta casilla se deben establecer las condiciones iniciales que deben darse para que la prueba pueda ser llevada a cabo en el módulo en cuestión. No es necesario definir aspectos triviales o comunes a todos los módulos como el navegador, la conexión a internet, el acceso a un computador... etc. Pero si se debe definir aquellos aspectos de los que dependa la prueba, por ejemplo, para crear una asignatura es necesario haber definido antes un propósito de asignatura, haber ingresado como administrador o tutor... etc.
<b>Post condiciones</b>	Aquí se debe establecer las condiciones de salida de la prueba, es decir, el estado esperado del sistema una vez se realiza la prueba.
<b>Script de prueba</b>	<ul style="list-style-type: none"> <li>- Lanzar el servidor en el ambiente de pruebas</li> <li>- General 100 solicitudes simultaneas en un espacio de 10 segundos</li> <li>- Medir cuantas solicitudes fueron exitosas (unidades)</li> <li>- Medir el tiempo promedio de las solictudes (seg)</li> <li>- Medir el tiempo total de ejecución (min)</li> <li>- Medir el throuput de la funcionalidad (solicitudes/min)</li> <li>- Medir el tamaño promedio de respuesta (kb)</li> </ul>
<b>Observaciones</b>	En este espacio se pueden realizar las observaciones a las que hubiere lugar y que se consideren de importancia para el desarrollo del caso de prueba.

**4.1.5 Tipos de errores.** Durante la ejecución de las pruebas se deberán tipificar los errores de acuerdo a la siguiente tabla.

Tabla 4 Tipos de errores

<b>Tipo de error</b>	<b>Descripción</b>	<b>% Fallo</b>
<b>Crítico</b>	Error que produce el fallo general de la plataforma e impide que la prueba continúe. Su aparición afecta gravemente la operación de la plataforma.	1
<b>Grave</b>	Este tipo de error produce el fallo del módulo en proceso de prueba, ya sea porque no funciona o el resultado arrojado no es el esperado.	0,8
<b>Intermedio</b>	El error provoca un comportamiento inesperado de la plataforma, pero el módulo continúa funcionando y el resultado es el esperado.	0,6
<b>Menor</b>	Comportamiento inesperado del módulo en prueba que no afecta su funcionamiento ni el resultado esperado. Problemas de usabilidad, de	0,4

	pantallas... etc.	
<b>Simple</b>	Errores insignificantes que se pueden solucionar de manera sencilla, por ejemplo, un error cosmético, un texto desalineado, una falla de ortografía... etc.	0,2

El porcentaje de fallo permite determinar el porcentaje de éxito de una suite de pruebas: por ejemplo, si se realizan 5 casos de prueba: 3 fueron exitosos (% fallo= 0) y 2 con errores intermedios (0,6 cada uno), se dice entonces que la suite de pruebas falló 1,2(0,6+0,6) casos de prueba, es decir tuvo un porcentaje de error del 24% (1,2 casos de 5 totales), o bien, fue aprobada en un 76% (3,8 de 5 totales).

**4.1.6 Prioridad.** Es importante aclarar que la prioridad no está directamente relacionada con el tipo de error. Puede existir un error grave, pero si es causado por una acción poco usual o que, por ejemplo, sólo realiza el administrador, no tiene una prioridad tan alta como otro error menor, pero que corresponda a un módulo de alta concurrencia.

De esa forma, la prioridad se determina según la inmediatez con la que sea necesario atender el caso según la siguiente tabla:

Tabla 5 Prioridad de un error

<b>Prioridad</b>	<b>Descripción</b>
<b>Altísima</b>	Cuando el error produce el fallo general de la plataforma e impide que la prueba continúe. Su aparición afecta gravemente la operación de la plataforma.
<b>Alta</b>	Cuando el error se presenta en un módulo con alta concurrencia y que su presencia afecta considerablemente la normal operación.
<b>Normal</b>	Cuando el error se presenta en un módulo de baja concurrencia y su aparición no entorpece seriamente el funcionamiento de los demás módulos.
<b>Baja</b>	Cuando el error se presenta en un módulo con poca concurrencia y no representa mayor problema al funcionamiento de la plataforma.

**4.1.7 Formato para el reporte de incidencias.** Para realizar el reporte de una incidencia se recomienda usar el software mencionado en el numeral **MantisBT**. Para el reporte se deberá usar el siguiente formato o uno similar que contenga

mínimamente los mismos campos o sus análogos. El equipo de calidad podrá definir otros en caso de considerarlo necesario.

Tabla 6 Formato para el reporte de incidencias

<b>Código</b>		<b>MIE-TCxx es el código del caso de prueba donde se presentó el error.</b>	
<b>Reproducibilidad</b>	Aquí se define cuán reproducible es el error detectado durante el caso de prueba. Las opciones disponibles son: <ul style="list-style-type: none"> <li>- Siempre: (cuando el error se presenta siempre que se ejecuta la prueba).</li> <li>- A veces (Cuando el error se presenta sólo algunas veces).</li> <li>- No reproducible (Cuando el error se presenta sólo una vez, pero al repetir la prueba no se logra reproducir).</li> </ul>		
<b>Tipo de error</b>	Se define de acuerdo a los tipos de errores anteriormente definidos en la sección 4.1.3., con su respectiva prioridad.		
<b>Prioridad</b>	Definida de acuerdo al tipo de error clasificado.		
<b>Actores</b>	Define el actor que participó en la prueba		
<b>Pre condiciones</b>	Define las condiciones iniciales con las que se realizó la prueba. Aquí también se definen el tipo de equipo o terminal usado, navegador... etc.		
<b>Resultado obtenido.</b>	Aquí se debe establecer las condiciones de salida obtenidas durante la prueba.		
<b>Pasos para reproducir</b>	<b>Actores</b>	<b>Sistema</b>	
	En estas casillas se define el paso a paso de lo que realizó el actor para reproducir el error.	En estas casillas se define el paso a paso de lo que realizó el sistema.	
<b>Observaciones</b>	En este espacio se pueden realizar las observaciones a las que hubiere lugar y que se consideren de importancia.		
<b>Adjuntar archivos</b>	El software de gestión de incidencias permite adjuntar archivos, de tal manera que aquí se recomienda adjuntar todos los posibles archivos de configuración, pantallazos, etc. Que ayuden con la reproducibilidad del error.		

**4.1.8 Formato para el reporte de resultados.** Pese a que cada incidencia debe reportarse siguiendo el formato establecido para tal fin y que cada herramienta presenta un informe detallado de resultados, en general, se define el siguiente formato como el que se deberá diligenciar mínimamente después de la ejecución de cada suite de pruebas.

Tabla 7 Formato para el reporte de resultados

N° Código MIE		Resultado	Tipo de error
#	MIE - TCxxx	Aquí se describe brevemente el resultado obtenido.	En caso de tener un error, se enuncia según el tipo de error.
#	MIE-TCxxx	Se agregarán tantas filas como casos de prueba se realicen	(Crítico, grave, intermedio, menor, simple).
<b>Conclusión:</b> Aquí se finaliza enunciando las conclusiones a partir de los resultados obtenidos como el porcentaje de acierto y desacierto en el ciclo de pruebas.			

## 4.2 PROTOCOLO DE PRUEBAS REQUERIMIENTOS FUNCIONALES

**4.2.1 Pruebas Unitarias o de Módulo.** Para esta fase se consideran aquellas funcionalidades unitarias de cada menú o paquete por separado y se definen como módulos unitarios, y son los primeros en ser puestos a prueba. Con el fin de automatizar las pruebas se recomienda el uso de herramientas como “Selenium IDE” de la suite de Selenium. Para realizar el reporte de incidencias se recomienda “Mantis Bug Tracker. Finalmente, para el diseño de casos de uso se puede usar el software “testlink. Sin embargo, si la organización cuenta con otro tipo de herramientas para los fines anteriores, se podrán usar siempre que se puedan adaptar al presente protocolo.

Los módulos unitarios conocidos hasta el momento de desarrollo del presente protocolo, se organizan a continuación de acuerdo a su respectivo menú, que en adelante llamaremos, paquete. Los test cases correspondientes se enumeran uno a uno de acuerdo al consecutivo: MIE-TCx.A.B.C, según se define en el Formato para el diseño de casos de prueba funcionales. El consecutivo cero (0) se usa para empaquetar aquellos módulos relacionados con la sesión del usuario, uno (1) para asignatura, dos (2) para módulo, tres (3) para actividad, cuatro (4) para recursos y cinco (5) para usuarios.

Los módulos unitarios organizados por paquetes y con su correspondiente funcionalidad se organizan en la siguiente tabla:

Tabla 8 Relación entre paquetes, módulos y funcionalidades

Paquete	Modulo	Funcionalidad
<b>0. Sesión</b>	1. Inicio de sesión	Iniciar sesión
		Recuperar Contraseña
	2. Cerrar sesión	Cerrar Sesión
<b>1. Asignatura</b>	1. Propósito Asignatura	Crear propósito de asignatura
		Editar propósito de asignatura
		Eliminar propósito de asignatura
	2. Competencias	Crear competencia
		Editar competencia
		Eliminar competencia
	3. Tipos de competencias	Ingresar tipo de competencia
		Editar tipo de competencia
		Eliminar tipo de competencia
	4. Asignatura	Crear asignatura
		Editar asignatura
		Eliminar asignatura
	5. Asignar competencia a asignatura	Asignar competencia a asignatura
		Editar asignación
Eliminar asignación		
6. Grupos de asignatura	Crear grupo de asignatura	
	Editar grupo de asignatura	
	Eliminar grupo de asignatura	
<b>2. Módulo</b>	1. Tipos de temáticas	Crear tipo de temática
		Editar tipo de temática
		Eliminar tipo de temática
	2. Temáticas	Crear temática
		Editar temática
		Eliminar temática
	3. Estrategias de enseñanza y aprendizaje	Crear estrategias de enseñanza y aprendizaje
		Editar estrategias de enseñanza y aprendizaje
		Eliminar estrategias de enseñanza y aprendizaje
	4. Módulos de asignatura	Crear módulo de asignatura
		Editar módulo de asignatura
		Eliminar módulo de asignatura
	5. Asignar tutor a módulo	Asignar tutor a módulo
		Editar asignación
		Eliminar asignación
<b>3. Actividad</b>	1. Tipo de actividad	Crear tipo de actividad
		Editar tipo de actividad
		Eliminar tipo de actividad
	2. Actividades de estrategia de enseñanza y aprendizaje	Crear actividad
		Editar actividad
		Eliminar actividad
	3. Asignar recursos a actividad	Asignar recursos a actividad
		Editar asignación
		Eliminar asignación
	1. Tipos de recursos	Crear tipo de recurso

<b>4. Recursos</b>		Editar tipo de recurso
		Eliminar tipo de recurso
	2. Recursos	Crear recurso
		Editar recurso
		Eliminar recurso
	3. Foros	Crear foro
		Editar foro
	Eliminar foro	
	Adicionar participante	
	Eliminar participante	
<b>5. Usuarios</b>	1. Usuario	Crear nuevo usuario
		Editar información de usuario existente
		Eliminar usuario existente
	2. Rol	Crear nuevo rol
		Editar rol existente
		Eliminar rol existente
	3. Asignar rol a usuario	Asignar rol a usuario
		Editar asignación
		Eliminar asignación
	4. Datos de tutores	Crear registro de tutor
		Editar datos
		Eliminar registro
	5. Instituciones	Crear institución
		Editar información de institución
		Eliminar registro de institución.
	6. Datos de estudiantes	Crear registro de estudiante
		Editar datos
		Eliminar registro
	7. Asignar estudiante a grupo	Asignar estudiante a grupo
		Editar asignación
		Eliminar asignación.

En el evento que se deban agregar nuevos módulos y funcionalidades, al igual que eliminar alguna ya existente, se agregaran u omiten siguiendo el mismo procedimiento y de acuerdo al requerimiento que motivó su desarrollo o eliminación.

Es importante también tener en cuenta que cada caso de uso puede motivar el desarrollo de varios casos de prueba. Por ejemplo, la funcionalidad: “crear usuario” del paquete “Usuarios” puede motivar casos de prueba como:

- Crear usuario “Roberto Baez” (caso nominal): El caso de prueba va a introducir todos los datos del usuario “Roberto Baez”.

- Crear usuario (manejo de excepción: sin nombre): En este caso se va a tratar de probar el funcionamiento de la plataforma en caso de que no se introduzca el campo “nombre” de la tabla para crear el nuevo usuario. De acuerdo a los requerimientos, ningún usuario podría ser creado sin nombre, de manera que los resultados esperados serían, que la plataforma no dejara guardar el nuevo usuario hasta que se introduzca su nombre.
- Crear usuario sin datos: De manera similar a la anterior, se busca probar si es posible crear un usuario sin ningún dato y estudiar si el comportamiento de la plataforma es el esperado de acuerdo al requerimiento de la funcionalidad.

Como se entenderá, no es posible estudiar todas las posibles variaciones de la funcionalidad, de tal manera que se deben seleccionar estratégicamente según considere la persona encargada de diseñar los casos de prueba. Al final del presente documento se presentan algunas recomendaciones de utilidad, pero el encargado podrá decidir si implementa otras estrategias a la hora de diseñar los casos de prueba.

De acuerdo a lo anteriormente establecido, el diseño de un caso de prueba o test case en el formato anteriormente definido quedaría así:

### **MIE-TCU111: Crear Propósito de Asignatura: “Analizar diseño curricular”**

Tabla 9 Ejemplo caso de prueba unitaria 1

<b>Código</b>		<b>MIE-TCU111</b>
<b>Nombre</b>	Crear Propósito de Asignatura: “Analizar diseño curricular” - caso nominal.	
<b>Descripción</b>	Este caso de prueba busca probar la creación de un propósito de una asignatura con el nombre “Analizar diseño curricular”	
<b>Paquete-Modulo</b>	Asignatura - Propósitos Asignatura	
<b>Dependencias</b>		
<b>Actores</b>	Administrador, Tutor, Coordinador del programa.	
<b>Pre condiciones</b>	El actor debe haber iniciado sesión en el sistema.	
<b>Post condiciones</b>	El sistema almacena el propósito de asignatura “Analizar diseño curricular”.	

<b>Escenario principal</b>	Actores	Sistema
	Ingresar al menú de Asignatura-Propósito de asignatura.	Despliega la ventana de propósitos de asignatura donde lista los propósitos ya existentes.
	Da clic en la opción de "nuevo propósito".	El sistema despliega una ventana donde solicita el nuevo propósito de asignatura.
	El actor digita el siguiente texto: "Analizar el diseño curricular" y da clic en "Guardar".	El sistema regresa a la ventana de propósitos de asignatura donde muestra el nuevo propósito creado junto a los anteriormente existentes.
<b>Observaciones</b>		

## MIE-TCU112: Editar Propósito de Asignatura

Tabla 10 Ejemplo caso de prueba unitaria 2

<b>Código</b>		<b>MIE-TCU112</b>
<b>Nombre</b>	Editar Propósito de Asignatura-Caso nominal	
<b>Descripción</b>	Este caso de prueba busca probar la edición de un propósito de una asignatura.	
<b>Paquete-Modulo</b>	Asignatura - Propósitos Asignatura	
<b>Dependencias</b>		
<b>Actores</b>	Administrador, Tutor, Coordinador del programa.	
<b>Pre condiciones</b>	El actor debe haber iniciado sesión en el sistema. El propósito "Analizar el diseño curricular" debe existir	
<b>Post condiciones</b>	El sistema almacena el valor "Idear un nuevo diseño" en lugar del propósito "Analizar el diseño curricular".	
<b>Escenario principal</b>	Actores	Sistema
	Ingresar al menú de Asignatura-Propósito de asignatura.	Despliega la ventana de propósitos de asignatura donde lista los propósitos ya existentes.
	Identifica el propósito "Analizar el diseño curricular" a editar y da clic en la opción "editar" de ese propósito.	El sistema despliega una ventana donde muestra la información existente del propósito.
	El actor elimina el texto existente, digita el siguiente texto: "Idear un nuevo diseño" y da clic en "Guardar".	El sistema regresa a la ventana de propósitos de asignatura donde muestra el nuevo propósito creado junto a los ya existentes.
<b>Observaciones</b>		

## MIE-TCU533: Editar el rol ya asignado a un usuario

Tabla 11 Ejemplo caso de prueba unitaria 3

Código		MIE-TCU533
<b>Nombre</b>	Editar el rol ya asignado a un usuario – caso nominal	
<b>Descripción</b>	Este caso de prueba busca probar la edición del rol de un usuario a quien ya se le ha asignado uno previamente.	
<b>Paquete-Modulo</b>	Usuarios- Asignar rol a usuario	
<b>Dependencias</b>	Depende de los módulos “Rol” y “Usuario” del mismo paquete.	
<b>Actores</b>	Administrador, Tutor, Coordinador del programa.	
<b>Pre condiciones</b>	El actor debe haber iniciado sesión en el sistema. Los roles “tutor” y “estudiante”, al igual que el usuario “gpedraza” ya debieron crearse. El usuario “gpedraza” se le debe haber definido el rol de estudiante.	
<b>Post condiciones</b>	El sistema quita los privilegios de estudiante al usuario “gpedraza” y le asigna los permisos de tutor.	
<b>Escenario principal</b>	Actores	Sistema
	Ingresar al menú de Usuario-Asignar rol a usuario.	Despliega la ventana donde se muestra la tabla relacional de cada usuario con su respectivo rol.
	Identifica el usuario “gpedraza” y da clic en la opción de “editar”.	El sistema despliega una ventana donde muestra la relación existente entre el usuario y su respectivo rol.
	El actor selecciona el rol “tutor” para el usuario “gpedraza” y da clic en guardar.	El sistema regresa a la ventana donde se muestra la tabla relacional de cada usuario con su respectivo rol, donde se aprecia el usuario “gpedraza” con el rol de “tutor”.
<b>Observaciones</b>	Como verificación final, se deberá ingresar con las credenciales del usuario “gpedraza” y verificar que, en efecto, el sistema le ha dado los privilegios de tutor y le ha quitado los permisos de estudiante.	

**4.2.2 Pruebas de Regresión.** Como ya se definió en el marco teórico y en las generalidades del protocolo, estas pruebas se aplican una vez se han realizado el primer ciclo correspondiente a las pruebas unitarias o de módulo especificadas anteriormente, dando como resultado un 80% o más éxito durante este ciclo. Una vez superado este umbral, se procede al segundo ciclo que se inicia con las pruebas de regresión.

Durante estas pruebas se repiten todas aquellas pruebas que se hayan reportado como fallidas durante el primer ciclo. Adicionalmente se prueba un 10% de las

pruebas consideradas como exitosas para comprobar que los nuevos ajustes no hayan inducido nuevos errores. Este 10% debe ser seleccionado estratégicamente y en caso de ser necesario, puede ampliarse a criterio del equipo de calidad sin extenderse demasiado. Se deben buscar aquellos casos de prueba de módulos que tengan mayor dependencia de otros módulos como primera prioridad. También se deben tratar de incluir aquellos que dependan de módulos que durante el primer ciclo hayan tenido un comportamiento inesperado. Si con los dos anteriores criterios no se cumple la cuota establecida, los demás casos de prueba faltantes serán elegidos a criterio propio del equipo de calidad.

**4.2.3 Pruebas de Integración.** En esta fase se desarrolla durante el segundo ciclo y en ella se procederán a probar la forma como se comportan los módulos anteriormente probados cuando se integran entre sí. Se inicia probando el módulo que tiene menores dependencias y se van agregando módulos progresivamente de acuerdo al diagrama de interdependencia del paquete correspondiente que se presenta a continuación.

Figura 3 Diagrama de dependencia paquete: Asignatura

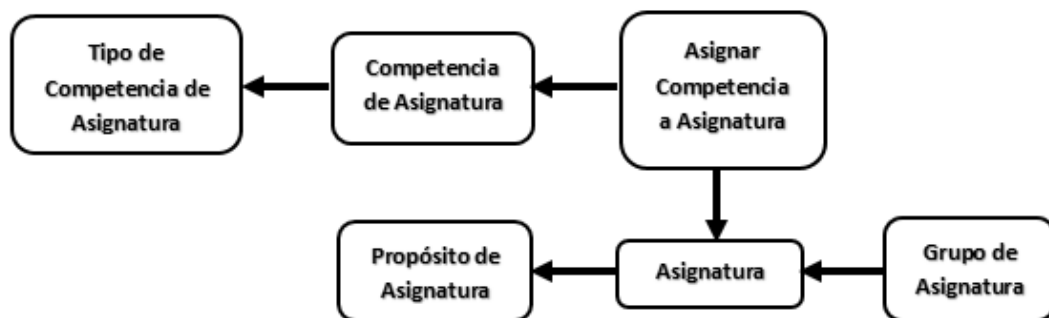


Figura 4 Diagrama de dependencia paquete: Módulo

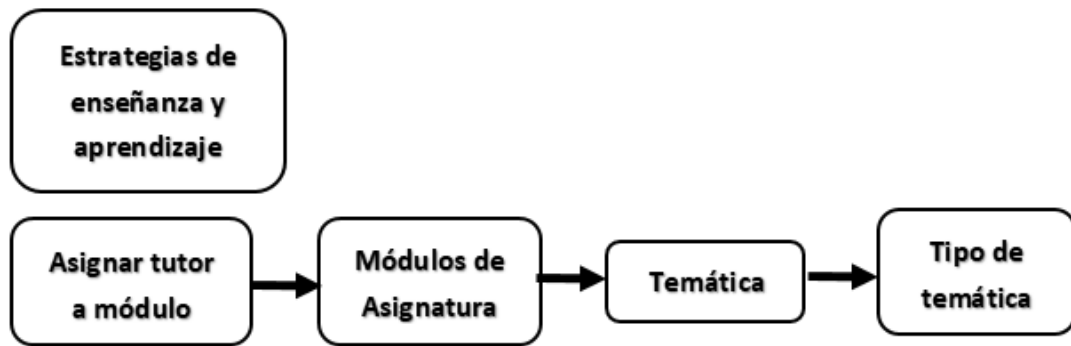


Figura 5 Diagrama de dependencia paquete: Actividad



Figura 6 Diagrama de dependencia paquete: Recurso

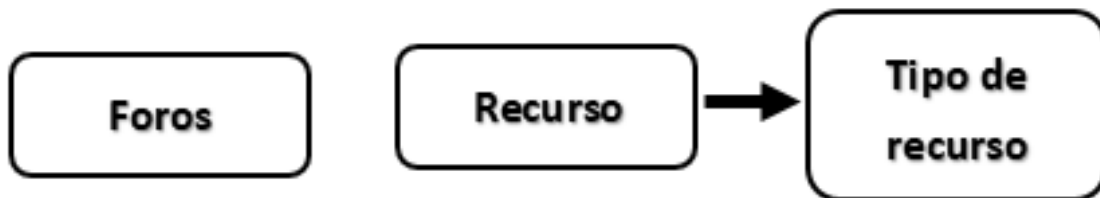
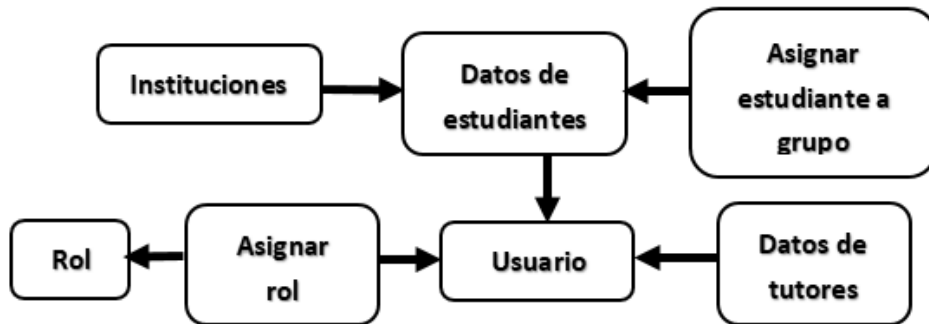


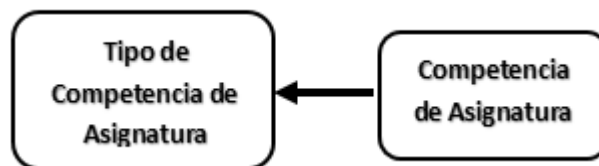
Figura 7 Diagrama de dependencia paquete: Usuarios



Aquellas relaciones existentes entre paquetes se ignoran pues se estudian en la siguiente fase. En caso de agregarse o eliminarse algún módulo, el diagrama deberá ser actualizado al igual que la prueba de integración.

Un ejemplo de prueba de integración del módulo “Competencia de Asignatura” implicaría de acuerdo a la Figura 3 Diagrama de dependencia paquete: Asignatura, **¡Error! No se encuentra el origen de la referencia.** la integración de los módulos de “Tipo de competencia de Asignatura” con “Competencia de Asignatura”, ya que ésta última sólo tiene dependencia de esa primera:

Figura 8 Dependencia entre dos modulos para prueba de integración



Si se dijera que se va a realizar la prueba de integración para el módulo “asignar competencia a asignatura”, entonces se observa en la figura 3 que este módulo depende de dos módulos que a su vez dependen de otros dos módulos, todos ellos deberán ser secuencialmente probados.

A continuación se presenta el caso de prueba de integración del módulo “Competencia de Asignatura”.

**MIE-TCI121: Competencia de asignatura “Competencia 1” del tipo de competencia de asignatura “1 Tipo de competencia”.**

Tabla 12 Ejemplo caso de uso de integración MIE-TCI121

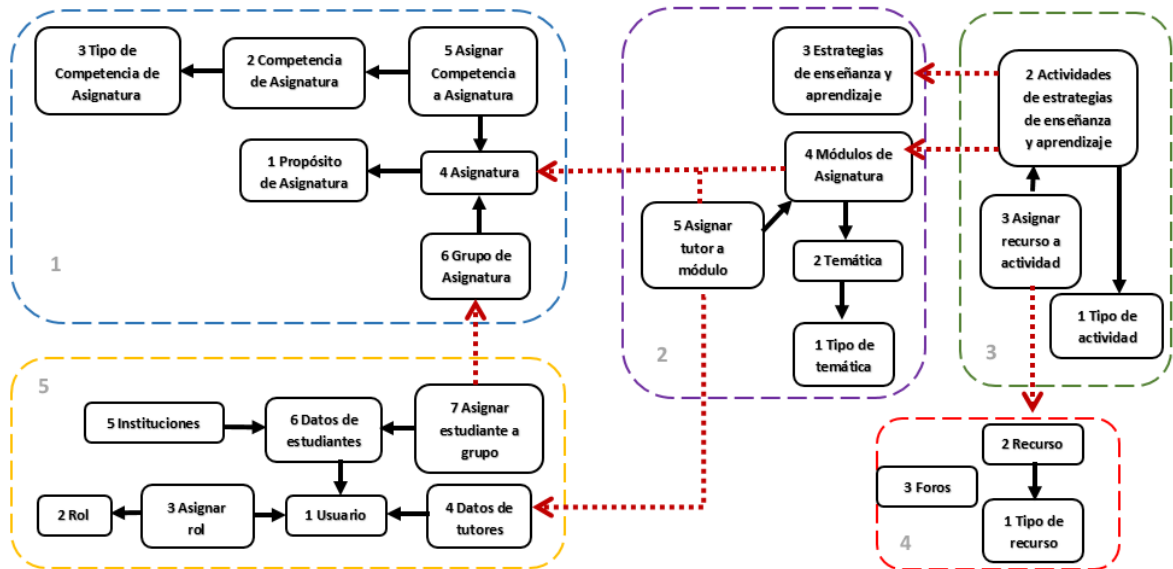
<b>Código</b>		<b>MIE-TCI121</b>
<b>Nombre</b>	Competencia de asignatura “Competencia 1” del tipo de competencia de asignatura “1 Tipo de competencia”.	
<b>Descripción</b>	Este caso de prueba busca probar la creación de una competencia de asignatura “Competencia 1” del tipo de competencia de asignatura “1 Tipo de competencia”.	
<b>Paquete-Modulo</b>	Asignatura – Competencia de Asignatura	
<b>Dependencias</b>	Tipo de Competencia de Asignatura	
<b>Actores</b>	Administrador, Tutor, Coordinador del programa.	
<b>Pre condiciones</b>	El actor debe haber iniciado sesión en el sistema.	
<b>Post condiciones</b>	El sistema almacena la competencia de asignatura “Competencia 1” del tipo de competencia de asignatura “1 Tipo de competencia”.	
<b>Escenario principal</b>	<b>Actores</b>	<b>Sistema</b>
	Ingresar al menú de Asignatura-Tipo de competencia de asignatura.	Despliega la pantalla de tipo de competencia de asignatura donde lista los tipos de competencia de asignatura ya existentes.
	Da clic en la opción de “nuevo tipo de competencia”.	El sistema despliega una ventana donde solicita el nuevo tipo de competencia de asignatura.
	Digita el siguiente texto: “1 Tipo de competencia” y da clic en “Guardar”.	El sistema regresa a la pantalla de tipo de competencia de asignatura donde muestra el nuevo tipo de competencia creado junto a los que ya existían.
	Ingresar al menú de Asignatura-Competencia de asignatura.	Despliega la pantalla de competencia de asignatura donde lista las competencias de asignatura ya existentes.
	Da clic en la opción de “nueva competencia”.	El sistema despliega una ventana donde solicita la nueva competencia y el tipo al que corresponde.
	El actor digita el siguiente texto: “Competencia 1” y selecciona el tipo “ 1 tipo de competencia” y da clic en “Guardar”.	El sistema regresa a la pantalla de competencia de asignatura donde muestra la nueva competencia creada junto a los que ya existían.
<b>Observaciones</b>		

Se sigue un procedimiento similar para cada uno de los diferentes módulos integrándolos con los demás de su mismo paquete sin pasar por alto ninguna interrelación.

**4.2.4 Pruebas de Sistema.** Estas pruebas cierran el segundo ciclo de pruebas funcionales. En ésta fase se definirán casos de prueba de tal manera que implique el uso de todo el sistema. Se busca la interacción de tantos módulos como sea posible, al igual que emular un uso típico esperado del sistema por parte de un usuario en cualquiera de los posibles roles. Por tratarse de una prueba más general, aquí se van a tener en cuenta los roles y sus funciones de acuerdo al análisis de requerimientos en búsqueda de evaluar todas las posibles acciones que un rol de usuario puede o no realizar. De esa forma se evalúa también el control de acceso a contenidos.

El siguiente es el diagrama que relaciona los diferentes módulos existentes hasta el momento. El recuadro azul representa el menú asignatura, el púrpura el menú módulo, el verde el menú actividad, el rojo el menú recursos y el amarillo el menú de usuario.

Figura 9 Diagrama de interdependencia entre paquetes



Como en este caso se busca probar la interrelación de los diferentes paquetes, se deben crear casos de prueba como: “asignar un tutor a un módulo”, “crear actividades de estrategias de enseñanza y aprendizaje” para luego “asignar actividad a un recurso”, “asignar un estudiante a un grupo”... etc. Todas aquellas donde se aprecie que un grupo de módulos pertenecientes a un paquete se relacionen con otros grupos de módulos de otro paquete y que en el gráfico se resalten como líneas rojas punteadas.

Usando el diagrama de dependencias, el equipo de calidad debe definir las rutas de prueba de sistema que evalúen objetivamente la interrelación entre los diferentes paquetes incluyendo la mayor cantidad posible de módulos y sin que ninguno de ellos quede sin pertenecer a al menos una ruta de prueba.

#### 4.3 PROTOCOLO DE PRUEBAS REQUERIMIENTOS NO FUNCIONALES

Dado que el proyecto MIE se trata de un proyecto que se encuentra en su primera fase y que pretende mantenerse como una herramienta a largo plazo, al igual que

por su propósito y aplicación; los atributos de mayor importancia a evaluar son el rendimiento, carga y usabilidad.

Aspectos como la seguridad o disponibilidad son tenidos en cuenta pero de forma muy austera por las razones que se explican a continuación.

En cuanto a seguridad, no se hace especial hincapié por tratarse de una plataforma académica que no cuenta con información privilegiada, financiera o confidencial. Sin embargo, aunque la seguridad no es un aspecto tan crítico, se debe comprobar que el acceso a los recursos, menús, etc. Sólo pueda hacerse por parte de los usuarios registrados según su rol y que un usuario no puede acceder ni modificar los datos de otro (con excepción del administrador, claro). Así mismo, se debe verificar que el sistema, para proteger la manipulación malintencionada de un tercero, cierre automáticamente la sesión después de un tiempo de inactividad definido. Este tiempo lo puede definir el equipo de calidad según su criterio. Lo importante es que el sistema efectivamente cierre la sesión una vez transcurrido ese tiempo.

Para impedir ataques por fuerza bruta también se recomienda el uso de una prueba de turing o captcha que detenga posibles intentos no deseados de acceso o un número importante de solicitudes irrelevantes desde una misma máquina que pueda afectar el rendimiento de la plataforma. Dicho lo anterior, se debe comprobar que el sistema cuenta con una forma de prevenir ataques de fuerza bruta o un número anormal de solicitudes desde una misma máquina.

Respecto a la disponibilidad, no es un aspecto tan esencial, ya que se espera que a la plataforma solo accedan personas con el uso horario colombiano, de manera que en horas de la madrugada podría ocasionalmente el servicio no estar disponible sin que ello afecte gravemente el funcionamiento.

**4.3.1 Métricas para pruebas de carga y rendimiento.** Por tratarse de una aplicación web, se recomienda usar una herramienta como Apache jMeter (ver Apache **JMeter**) que permita realizar múltiples solicitudes http simulando los usuarios concurrentes. Las métricas a evaluar en este caso serán:

- Número de solicitudes atendidas: La cantidad de solicitudes que el sistema resolvió exitosamente. Se compara con la cantidad de solicitudes enviadas.
- Tiempo total de la prueba: Es el tiempo empleado desde que se hace la primera solicitud hasta que el sistema atiende la última.
- Tiempo Promedio: Es la suma del tiempo empleado en resolver cada una de las solicitudes dividido en la cantidad total de solicitudes.
- Tiempo Máximo: El tiempo máximo que el sistema tardó en resolver una determinada solicitud.
- Tiempo Mínimo: El tiempo mínimo que el sistema tardó en resolver una determinada solicitud.
- Throughput: También se conoce como la “tasa de transferencia efectiva” y es el volumen de información o trabajo neto que fluye en un sistema. Para el caso de la plataforma MIE, es la cantidad de solicitudes que es capaz de resolver el sistema en un tiempo determinado [Solicitudes/min]. Es importante ya que, un sistema puede parecer muy eficiente al atender en 1 segundo una solicitud (tiempo promedio), pero si sólo logra resolver una solicitud a la vez, al cabo de un minuto sólo habrá resuelto 60 solicitudes, mientras que un sistema que se demore, 3 segundos por solicitud, pero logra resolver 6 solicitudes de forma simultánea, al cabo de un minuto habrá resuelto (20x6) 120 solicitudes por minuto, es decir, soporta un mayor volumen de información.

Estos valores de métricas esperadas son las que el equipo de desarrollo definió de acuerdo a los requerimientos del aplicativo, que para el momento del desarrollo del presente protocolo es:

Tabla 13 Métricas de carga y rendimiento

Carga	Número de usuarios	Ventana de respuesta
Normal	80	40ms a 80 ms
Media	120	60ms a 200ms
Alta	250	120ms a 400 ms

**4.3.2 Pruebas de Rendimiento.** Evaluaremos la latencia media de los módulos sobre los cuales se espera que tenga mayor concurrencia con parámetros de carga regulares. Es importante aclarar que se espera contar con sólo algunos administradores de la plataforma, así mismo, que acciones como crear cursos, actividades, tipos de recursos, usuarios... etc. (Actividades propias del rol de administrador) no se consideran de alta concurrencia porque se realizarán muy esporádicamente y por muy pocas personas.

Así mismo, por la dinámica misma del sistema educativo, un solo tutor tendrá a en su módulo a varios estudiantes, es decir, habrán siempre más estudiantes que tutores o administradores. Sólo en caso de que la plataforma crezca al punto de tener al menos 100 tutores, no se considerarán acciones de alta concurrencia las realizadas por este rol.

Por tanto, las posibles acciones a realizar por parte del rol de estudiante serán las de alta concurrencia a menos que el equipo de calidad decida incluir otras acciones en cuyo caso se deberá proceder de forma análoga a las acciones que el presente protocolo establece para el rol de estudiante. El rol de estudiante tiene acceso a tres menús:

- Perfil: Desde aquí accede a la opción de editar sus datos de usuario, editar la institución con la que trabaja y agregar una foto de perfil.
- Mis cursos: En este menú el estudiante puede visualizar sus correspondientes cursos junto con las respectivas actividades de módulo. Así mismo, puede

visualizar los recursos correspondientes a las mismas y cargar archivos entregables en caso de ser necesario.

- Evaluación docente: Este enlace lo dirige a un listado de docentes donde podrá evaluar el trabajo de cada uno de ellos.

De acuerdo a lo anterior, se deberán diseñar scripts que simulen el uso concurrente de una carga normal de usuarios para evaluar el rendimiento de la plataforma para las funcionalidades típicas del rol de estudiante por las razones ya expuestas.

Los resultados obtenidos deberán ser comparados con las métricas ya establecidas o las que establezcan el equipo de desarrollo para condiciones normales de operación. En caso de que el tiempo promedio de respuesta supere los 200ms (o el tiempo máximo que se establezca en las métricas de una carga normal), o exista algún comportamiento anómalo durante la prueba o en los resultados de la misma, se deberá realizar el reporte usando el formato establecido al equipo desarrollador adjuntando el caso de prueba de rendimiento debidamente diligenciado en el formato que el presente protocolo define para tal fin.

A continuación se presentan dos ejemplos de casos de prueba de rendimiento:

### **MIE-TCR120.1: Ver recurso asignado a una actividad “plantilla.doc”**

Tabla 14 Ejemplo 1 Caso de prueba de rendimiento.

<b>Código</b>		<b>MIE-TCR120.1</b>
<b>Nombre</b>	Ver recurso asignado: “plantilla.doc” de la actividad “Actividad en prueba” asignados a al módulo “modulo en prueba” de la Asignatura “Asignatura en prueba”.	
<b>Descripción</b>	Se pone a prueba el rendimiento de la plataforma cuando 120 usuarios de forma concurrente consultan el recurso “plantilla.doc” asignado a la “actividad en prueba” del “modulo en prueba”.	
<b>Paquete-Módulo</b>	Estudiante – Mis Cursos	

<b>Pre condiciones</b>	El sistema debe haber sido lanzado en ambiente de pruebas. El estudiante debe haber iniciado sesión satisfactoriamente. Debe existir una asignatura llamada "asignatura en prueba". El estudiante debe tener asignada la "asignatura en prueba". La "asignatura en prueba" debe tener un módulo llamado "módulo en prueba" con una actividad llamada "actividad en prueba". El recurso "plantilla.doc" debe estar asignado a la "actividad en prueba".
<b>Post condiciones</b>	El estudiante debe haber accedido al recurso "plantilla.doc" de la "actividad en prueba".
<b>Script de prueba</b>	<ul style="list-style-type: none"> <li>- El estudiante se dirige al paquete "estudiante" del menú principal.</li> <li>- Selecciona el Módulo "mis cursos" y en la tabla que se despliega ubica la "Asignatura en prueba" y selecciona dentro de su respectiva columna de opciones: "Ver actividades de módulos".</li> <li>- En la tabla que se despliega de Actividades busca la "actividad en prueba" y en su respectiva columna de opciones da clic en "ver detalles".</li> <li>- En la nueva ventana que se genera ubica el link de "recursos asociados" y da clic en él.</li> <li>- Selecciona el recurso "plantilla.doc".</li> </ul>
<b>Configuración del script de prueba.</b>	<ul style="list-style-type: none"> <li>- Generar 120 solicitudes simultaneas en un espacio de 10 segundos</li> <li>- Medir cuantas solicitudes fueron exitosas (unidades)</li> <li>- Medir el tiempo promedio de las solicitudes (seg)</li> <li>- Medir el tiempo total de ejecución (min)</li> <li>- Medir el throuput de la funcionalidad (solicitudes/min)</li> <li>- Medir el tamaño promedio de respuesta (kb)</li> </ul>
<b>Observaciones</b>	

## MIE-TCR120.2: Participar en foro "foro de prueba"

Tabla 15 Ejemplo 2 Caso de prueba de rendimiento.

<b>Código</b>	<b>MIE-TCR120.2</b>
<b>Nombre</b>	Participar en un foro de prueba
<b>Descripción</b>	Se pone a prueba el rendimiento de la plataforma cuando 120 usuarios de forma concurrente participan en un foro "foro de prueba" del cual son participantes.
<b>Paquete-Módulo</b>	Recursos – Foros
<b>Pre condiciones</b>	El sistema debe haber sido lanzado en ambiente de pruebas. El estudiante debe haber iniciado sesión satisfactoriamente. Debe existir un foro activo llamado "foro de prueba". El estudiante debe ser participante del foro "foro en prueba".
<b>Post condiciones</b>	El foro debe contener el comentario "un comentario de prueba".
<b>Script de prueba</b>	<ul style="list-style-type: none"> <li>- El estudiante se dirige al ícono superior derecho de su pantalla inicial y da clic sobre él.</li> <li>- En el menú que se despliega selecciona el foro denominado "foro de prueba"</li> <li>- En la ventana que se despliega ubica la opción "Nuevo comentario principal" y da clic en ella.</li> <li>- En la ventana que se muestra digita: "Este es un comentario principal" y da clic en guardar.</li> </ul>

<b>Configuración del script de prueba.</b>	<ul style="list-style-type: none"> <li>- Generar 120 solicitudes simultaneas en un espacio de 10 segundos</li> <li>- Medir cuantas solicitudes fueron exitosas (unidades)</li> <li>- Medir el tiempo promedio de las solictudes (seg)</li> <li>- Medir el tiempo total de ejecución (min)</li> <li>- Medir el throuput de la funcionalidad (solicitudes/min)</li> <li>- Medir el tamaño promedio de respuesta (kb)</li> </ul>
<b>Observaciones</b>	

**4.3.3 Pruebas de Carga.** Para las pruebas de carga evaluaremos el tiempo de respuesta de una cantidad de solicitudes que la plataforma logra resolver exitosamente sin errores durante un tiempo determinado. A diferencia de las pruebas de rendimiento, aquí no se usa una carga regular. Ésta se va aumentando desde una baja concurrencia de usuarios hasta una alta concurrencia.

El criterio de alta o baja concurrencia de usuarios, varía en la medida que la plataforma deba crecer, sin embargo, en el momento del desarrollo del presente protocolo, se tendrán en cuenta las métricas anteriormente establecidas en el apartado 4.3.1, a menos que el equipo de calidad y desarrollo consideren tener en cuenta unas diferentes.

En los resultados arrojados por la herramienta se debe comparar las métricas obtenidas con las esperadas y definir si el comportamiento está dentro de lo esperado, de no ser así, se debe adjuntar un informe de errores detallado siguiendo el formato definido para tal fin. Aquí, además de los tiempos mínimos, máximos y promedios, se recomienda medir la cantidad de solicitudes no atendidas y la desviación estándar de las mediciones para poder realizar un análisis más objetivo de los resultados.

A continuación se presentan los ejemplos de pruebas de carga el caso de prueba en el que varios estudiantes de forma concurrente deban acceder a un recurso asignado a una actividad para baja, media y alta concurrencia de acuerdo a las métricas establecidas.

## MIE-TCC80.1: Ver recurso asignado a una actividad “plantilla.doc”

Tabla 16 Ejemplo 1 Caso de prueba de carga

<b>Código</b>		<b>MIE-TCR80.1</b>
<b>Nombre</b>	Ver recurso asignado: “plantilla.doc” de la actividad “Actividad en prueba” asignados a al módulo “modulo en prueba” de la Asignatura “Asignatura en prueba”.	
<b>Descripción</b>	Se pone a prueba el rendimiento de la plataforma cuando 120 usuarios de forma concurrente consultan el recurso “plantilla.doc” asignado a la “actividad en prueba” del “modulo en prueba”.	
<b>Paquete-Módulo</b>	Estudiante – Mis Cursos	
<b>Pre condiciones</b>	El sistema debe haber sido lanzado en ambiente de pruebas. El estudiante debe haber iniciado sesión satisfactoriamente. Debe existir una asignatura llamada “asignatura en prueba”. El estudiante debe tener asignada la “asignatura en prueba”. La “asignatura en prueba” debe tener un módulo llamado “módulo en prueba” con una actividad llamada “actividad en prueba”. El recurso “plantilla.doc” debe estar asignado a la “actividad en prueba”.	
<b>Post condiciones</b>	El estudiante debe haber accedido al recurso “plantilla.doc” de la “actividad en prueba”.	
<b>Script de prueba</b>	<ul style="list-style-type: none"> <li>- El estudiante se dirige al paquete “estudiante” del menú principal.</li> <li>- Selecciona el Módulo “mis cursos” y en la tabla que se despliega ubica la “Asignatura en prueba” y selecciona dentro de su respectiva columna de opciones: “Ver actividades de módulos”.</li> <li>- En la tabla que se despliega de Actividades busca la “actividad en prueba” y en su respectiva columna de opciones da clic en “ver detalles”.</li> <li>- En la nueva ventana que se genera ubica el link de “recursos asociados” y da clic en él.</li> <li>- Selecciona el recurso “plantilla.doc”.</li> </ul>	
<b>Configuración del script de prueba.</b>	<ul style="list-style-type: none"> <li>- Generar 80 solicitudes simultaneas en un espacio de 10 segundos</li> <li>- Medir cuantas solicitudes fueron exitosas (unidades)</li> <li>- Medir el tiempo promedio de las solictudes (seg)</li> <li>- Medir el tiempo total de ejecución (min)</li> <li>- Medir el throuput de la funcionalidad (solicitudes/min)</li> </ul> Medir el tamaño promedio de respuesta (kb)	
<b>Observaciones</b>		

## MIE-TCC120.2: Ver recurso asignado a una actividad “plantilla.doc”

Tabla 17 Ejemplo 2 Caso de prueba de carga

<b>Código</b>		<b>MIE-TCC120.2</b>
<b>Nombre</b>	Ver recurso asignado: “plantilla.doc” de la actividad “Actividad en prueba” asignados a al módulo “modulo en prueba” de la Asignatura “Asignatura en prueba”.	
<b>Descripción</b>	Se pone a prueba el rendimiento de la plataforma cuando 120 usuarios de forma concurrente consultan el recurso “plantilla.doc” asignado a la	

	“actividad en prueba” del “modulo en prueba”.
<b>Paquete-Módulo</b>	Estudiante – Mis Cursos
<b>Pre condiciones</b>	El sistema debe haber sido lanzado en ambiente de pruebas. El estudiante debe haber iniciado sesión satisfactoriamente. Debe existir una asignatura llamada “asignatura en prueba”. El estudiante debe tener asignada la “asignatura en prueba”. La “asignatura en prueba” debe tener un módulo llamado “módulo en prueba” con una actividad llamada “actividad en prueba”. El recurso “plantilla.doc” debe estar asignado a la “actividad en prueba”.
<b>Post condiciones</b>	El estudiante debe haber accedido al recurso “plantilla.doc” de la “actividad en prueba”.
<b>Script de prueba</b>	<ul style="list-style-type: none"> <li>- El estudiante se dirige al paquete “estudiante” del menú principal.</li> <li>- Selecciona el Módulo “mis cursos” y en la tabla que se despliega ubica la “Asignatura en prueba” y selecciona dentro de su respectiva columna de opciones: “Ver actividades de módulos”.</li> <li>- En la tabla que se despliega de Actividades busca la “actividad en prueba” y en su respectiva columna de opciones da clic en “ver detalles”.</li> <li>- En la nueva ventana que se genera ubica el link de “recursos asociados” y da clic en él.</li> <li>- Selecciona el recurso “plantilla.doc”.</li> </ul>
<b>Configuración del script de prueba.</b>	<ul style="list-style-type: none"> <li>- Generar 120 solicitudes simultaneas en un espacio de 10 segundos</li> <li>- Medir cuantas solicitudes fueron exitosas (unidades)</li> <li>- Medir el tiempo promedio de las solicitudes (seg)</li> <li>- Medir el tiempo total de ejecución (min)</li> <li>- Medir el thruput de la funcionalidad (solicitudes/min)</li> </ul> <p>Medir el tamaño promedio de respuesta (kb)</p>
<b>Observaciones</b>	

### MIE-TCC250.3: Ver recurso asignado a una actividad “plantilla.doc”

Tabla 18 Ejemplo 3 Caso de prueba de carga

Código		MIE-TCC250.3
<b>Nombre</b>	Ver recurso asignado: “plantilla.doc” de la actividad “Actividad en prueba” asignados a al módulo “modulo en prueba” de la Asignatura “Asignatura en prueba”.	
<b>Descripción</b>	Se pone a prueba el rendimiento de la plataforma cuando 120 usuarios de forma concurrente consultan el recurso “plantilla.doc” asignado a la “actividad en prueba” del “modulo en prueba”.	
<b>Paquete-Módulo</b>	Estudiante – Mis Cursos	
<b>Pre condiciones</b>	El sistema debe haber sido lanzado en ambiente de pruebas. El estudiante debe haber iniciado sesión satisfactoriamente. Debe existir una asignatura llamada “asignatura en prueba”. El estudiante debe tener asignada la “asignatura en prueba”. La “asignatura en prueba” debe tener un módulo llamado “módulo en prueba” con una actividad llamada “actividad en prueba”. El recurso “plantilla.doc” debe estar asignado a la “actividad en prueba”.	
<b>Post condiciones</b>	El estudiante debe haber accedido al recurso “plantilla.doc” de la “actividad en prueba”.	
<b>Script de prueba</b>	<ul style="list-style-type: none"> <li>- El estudiante se dirige al paquete “estudiante” del menú principal.</li> </ul>	

	<ul style="list-style-type: none"> <li>- Selecciona el Módulo “mis cursos” y en la tabla que se despliega ubica la “Asignatura en prueba” y selecciona dentro de su respectiva columna de opciones: “Ver actividades de módulos”.</li> <li>- En la tabla que se despliega de Actividades busca la “actividad en prueba” y en su respectiva columna de opciones da clic en “ver detalles”.</li> <li>- En la nueva ventana que se genera ubica el link de “recursos asociados” y da clic en él.</li> <li>- Selecciona el recurso “plantilla.doc”.</li> </ul>
<b>Configuración del script de prueba.</b>	<ul style="list-style-type: none"> <li>- Generar 250 solicitudes simultaneas en un espacio de 10 segundos</li> <li>- Medir cuantas solicitudes fueron exitosas (unidades)</li> <li>- Medir el tiempo promedio de las solicitudes (seg)</li> <li>- Medir el tiempo total de ejecución (min)</li> <li>- Medir el throuput de la funcionalidad (solicitudes/min)</li> <li>- Medir el tamaño promedio de respuesta (kb)</li> </ul>
<b>Observaciones</b>	

**4.3.4 Pruebas de Usabilidad.** Dado que la plataforma es una ayuda en el desarrollo de la maestría, su uso debe ser tan amigable y sencillo como sea posible, es importante que los usuarios puedan usar intuitivamente el aplicativo, que se entienda con facilidad su uso y que llevar a cabo una tarea no resulte tan dispendioso o complejo.

Por tratarse de un aspecto un tanto subjetivo, se evaluará mediante la aplicación de encuestas a perfiles de usuarios que correspondan con los que usarán la plataforma en sus diferentes tipos a la vez que se solicita realizar tareas sencillas mientras se cronometra el tiempo que le toma al usuario llevarla a cabo para compararlo con el que debería tomar según el equipo de desarrollo. Se le debe permitir al encuestado hacer sugerencias o comentarios acerca de cada tarea.

Durante la fase de producción se podrán aplicar las encuestas que correspondan a algunos usuarios reales que estén de acuerdo en participar para retroalimentar y rediseñar en caso de que así se considere.

Tras analizar las tareas más habituales de una plataforma web educativa, se identifican los siguientes aspectos a evaluar, pudiendo ser modificados por el equipo de calidad en caso de ser necesario con fines prácticos o para darle una

mayor objetividad a la prueba hacia las áreas de interés que se tengan en un momento determinado buscando establecer de forma general la forma como un usuario percibe las funcionalidades y contenidos de la plataforma:

4.3.4.1 Imagen corporativa. Este aspecto será el primero a evaluar por tratarse de la primera impresión que adquiere el usuario al ingresar a la página. Es importante aclarar que se debe responder sin antes navegar por el sitio ni conocerlo. Las preguntas aquí realizadas buscan establecer si el usuario identifica fácilmente a que institución pertenece el sitio web y si existe algún elemento que no esté acorde con la imagen que se busca proyectar. Así mismo, se busca establecer si el usuario logra identificar directamente los elementos esenciales a usar. A continuación se presentan algunas de las preguntas que se podrían hacer con el fin de evaluar este aspecto:

- De acuerdo a lo que logra ver en la pantalla: ¿A cuál institución considera usted que pertenece el sitio web? ¿Por qué?
- ¿Hay algún elemento que considere fuera de lugar o que considere que no corresponde al sitio web?
- ¿Relaciona los colores predominantes con la imagen de la institución?
- ¿Relaciona la dirección del sitio web con la institución?
- ¿Hacia qué tipo de audiencia cree que está dirigido este sitio web? Explique su respuesta.
- En caso de requerir información adicional o necesitar ponerse en contacto con la institución: ¿Puede encontrar fácilmente la información de soporte y/o contacto? ¿Considera que es suficiente la información ofrecida por el sitio web? ¿Pudo encontrarla fácilmente?

4.3.4.2 Contenido y organización. Para realizar estas preguntas se debe permitir al usuario navegar por la página de tal forma que pueda evaluar objetivamente si la forma como está estructurado el contenido le permite obtener una idea concreta y

acorde con la que se pretendía comunicar. Así mismo, se busca evaluar si la forma como se organizó la información es comprensible para un usuario determinado, si es posible navegar entre menús sin perderse. A continuación se presentan algunas de las preguntas que se podrían hacer con el fin de evaluar este aspecto:

- ¿Qué opina sobre la portada del sitio?
- ¿Considera que la información contenida es excesiva, suficiente, escasa o incomprensible?
- ¿Cuál considera que es el contenido más relevante y por qué?
- ¿Los textos usados para nombrar los enlaces brindan información acerca de la página a la que dirigen?
- ¿Logra distinguir fácilmente en cada página la forma como se navega por el sitio?
- ¿Considera sencillo ir de sección en sección de la página? ¿Se siente desubicado en algún **momento**?

4.3.4.3 Pruebas de tiempo. En estas pruebas se busca determinar la facilidad y el tiempo que toma el usuario para realizar una tarea típica de acuerdo a su rol en la plataforma, por lo tanto, es necesario conocer claramente los permisos o atributos de cada rol. A modo de ejemplo, a continuación se enuncian las opciones y permisos de acceso de los principales roles activos dentro de la plataforma para el momento en que se realiza el presente protocolo, sin embargo, antes de realizar estas pruebas es necesario verificar que no existan nuevos roles o que haya dejado de existir alguno de los aquí mencionados. De igual forma, se debe verificar que los permisos de acceso de cada rol sean los mismos y en caso de no ser así, actualizarlos antes de llevar a cabo la prueba.

Todos los roles de usuario en la parte superior derecha pueden acceder a los foros activos o el histórico de los ya cerrados mediante un ícono de burbuja de mensaje.

- **Administrador:**

Este rol de usuario cuenta con acceso a todos los menús ya mencionados del sistema.

- **Estudiante**

Cuenta con tres menús disponibles:

- Perfil: Desde aquí accede a la opción de editar sus datos de usuario, editar la institución en la que trabaja y agregar una foto de perfil.
- Mis cursos: En este menú el estudiante puede visualizar sus correspondientes cursos junto con las respectivas actividades de módulo. Así mismo, puede visualizar los recursos correspondientes a las mismas y cargar archivos entregables en caso de ser necesario.
- Evaluación docente: Este enlace lo dirige a un listado de docentes donde podrá evaluar el trabajo de cada uno de ellos.

- **Tutor**

Cuenta con tres menús disponibles:

- Perfil: Desde aquí accede a la opción de editar sus datos de usuario, editar la unidad en la que trabaja y agregar una foto de perfil.
- Módulos: Desde este menú visualiza los módulos, estudiantes, actividades y entregables.
- Fechas de actividades: Desde este menú accede a las diferentes actividades asignadas a sus correspondientes módulos para visualizarlas y editar sus fechas de inicio y fin.

Se espera que la persona encargada de administrar el sistema cuente con la preparación suficiente para entender y manejar ampliamente la plataforma, por lo tanto, las presentes pruebas no se aplicarán a este rol de usuario.

De esta forma, las pruebas de usabilidad estarán enfocadas principalmente a los tutores y estudiantes en sus respectivas acciones que típicamente, se espera, deban realizar durante el normal uso de la plataforma. A continuación se presentan algunas de las preguntas que se podrían hacer con el fin de evaluar este aspecto:

- ¿Cuánto tiempo toma cambiar la foto de perfil y actualizar sus datos personales?
- ¿Cuánto tiempo le toma participar en un foro activo?
- ¿Cuánto tiempo tarda en visualizar una actividad a realizar y adjuntar un archivo como entregable de la misma? (Para el caso de estudiantes)
- ¿Cuánto tiempo tarda en visualizar una actividad de módulo y descargar un archivo entregable de la misma? (Para el caso de tutores).
- ¿Cuánto tiempo tarde en cambiar la fecha de inicio y fin de una actividad? (Para el caso de tutores).

**4.3.5 Pruebas de compatibilidad.** Cuando un programa debe ser ejecutado en diversas plataformas, bajo diversos sistemas operativos, se realizan estas pruebas. Consisten en probarlos en cada una de los posibles entornos donde se supone que deberá operar. Para el caso del presente protocolo, probar la compatibilidad consistirá en usar diferentes navegadores para navegar en la plataforma y comprobar que todo se visualiza y opera normalmente.

Se recomienda usar el rol de administrador, por tratarse del rol con mayor número de funcionalidades y vistas. Una vez se inicia sesión con un usuario administrador, se procede a probar cada una de las funcionalidades de los diferentes módulos

descritos en la Tabla 8 Relación entre paquetes, módulos y funcionalidades, en cada uno de los navegadores más populares en el momento de la ejecución de la prueba.

En el evento que exista alguna incompatibilidad, ésta deberá reportar el paquete-módulo-funcionalidad en la que se presenta, al igual que el navegador con su respectiva versión y una descripción del problema presentado de acuerdo al formato establecido en el presente protocolo para tal fin.

## **5. VALIDACION DEL PROTOCOLO DE PRUEBAS**

### **5.1 GENERALIDADES DE LA VALIDACION**

Con el fin de validar y mejorar diferentes aspectos del protocolo, a continuación se procederá a aplicar el protocolo de pruebas para requerimientos funcionales y no funcionales.

Para diseñar los casos de prueba a usarse se tendrán en cuenta las diferentes recomendaciones que el protocolo establece. Además se buscará evaluar objetivamente el protocolo sin entrar en pruebas exhaustivas, por lo tanto, las pruebas aquí expuestas se caracterizarán por su sencillez y objetividad con el fin de llevar a cabo una validación completa del mismo.

#### **5.1.1 Criterios de selección de casos de prueba y ciclos**

##### **5.1.1.1 Primer ciclo:**

- 80% casos de pruebas que evalúan funcionalidades en su versión nominal.
- 10% casos de prueba que evalúen condiciones de frontera o manejo de excepciones.
- 10% casos de prueba que evalúan redundancias.

Lo anterior en búsqueda de cumplir el umbral establecido para avanzar a una segunda etapa y sabiendo que las pruebas de los diferentes módulos en su versión nominal tienen una alta probabilidad de éxito. Durante una prueba más exhaustiva, los diferentes porcentajes deberán ser más homogéneos.

5.1.1.2 Segundo ciclo. Se inicia como establece el protocolo:

- Repetir las pruebas de módulo fallidas.
- 10% de las pruebas unitarias exitosas consideradas más críticas: Crear usuarios.
- Pruebas de integración evaluando la interrelación entre dos módulos dentro de los paquetes usuario y asignatura.
- Pruebas de sistema evaluando la interrelación entre los paquetes de módulo, asignatura y usuario con la integración del módulo “Asignar tutor a módulo.
- Se finaliza con la aplicación de las pruebas de requerimientos no funcionales siguiendo las métricas establecidas en la sección Métricas para pruebas de carga y **rendimiento**

## 5.2 PRUEBAS REQUERIMIENTOS FUNCIONALES

**5.2.1 Primer ciclo.** Como sólo se trata de dar una rápida validación, se aplicarán 10 casos de prueba organizados de la siguiente manera acuerdo a lo establecido en las generalidades:

Tabla 19 Casos de prueba funcionales primer ciclo de validación

Tipo	Paquete	N	Código MIE-	Caso de prueba
				Nombre
<b>Func. en su versión nominal</b>	Sesión	1	TC011	Iniciar Sesión como “admin”
	Asignatura	2	TC131	Crear tipo de competencia de asignatura “1 Tipo de competencia de asignatura”.
		3	TC111	Crear propósito de asignatura “1 propósito de asignatura”
	Módulo	4	TC211	Crear tipo de temática “1 Tipo de temática“
	Actividad	5	TC311	Crear tipo de actividad “1 Tipo de actividad“
	Recursos	6	TC411	Crear tipo de recurso “1 Tipo de recurso“
	Usuarios	7	TC511	Crear usuario1 “Nombre1 Apellido1”,
8		TC521	Crear rol “1 rol de prueba”,	
<b>Excepciones</b>	Asignatura	9	TC141	Crear asignatura sin introducir ningún dato.
<b>Homonimias</b>	Usuarios	10	TC512	Crear usuario “Nombre1 Apellido1”

5.2.1.1 Casos de prueba y ejecución. De acuerdo a lo anteriormente establecido, se procede a diseñar los casos de pruebas en la herramienta testlink y a ejecutarse usando Selenium, por organización del presente documento, se usa el formato establecido en lugar de mostrar el diseño en testlink para los casos de uso.

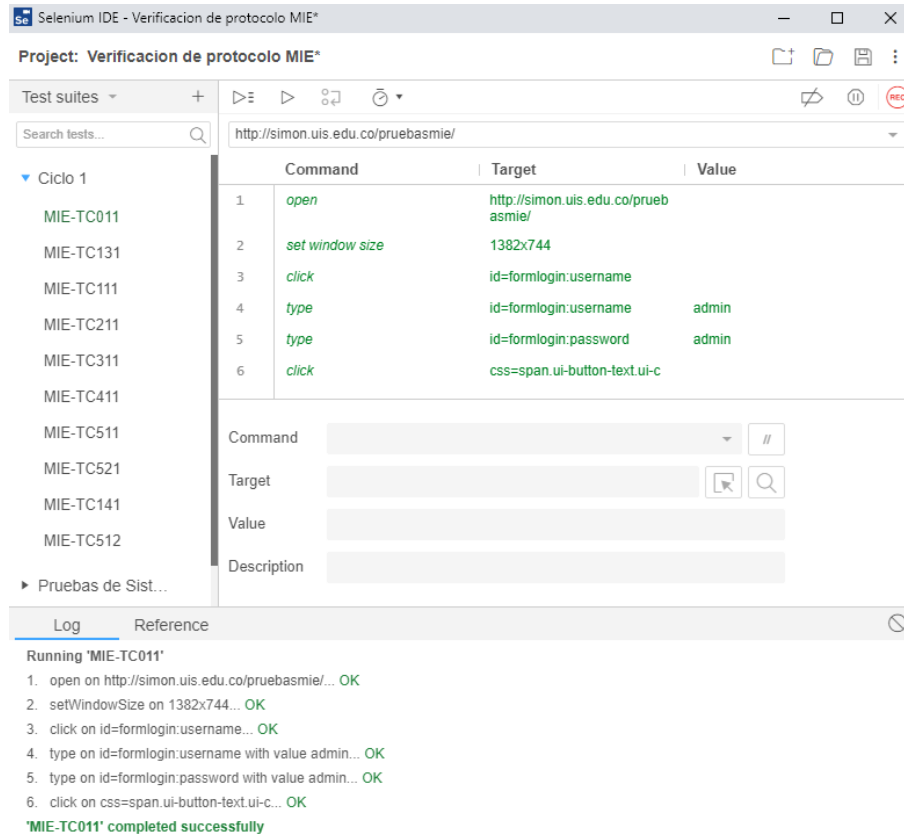
### **MIE-TCU011: Iniciar sesión con el usuario “Admin”**

Tabla 20 Caso de prueba MIE-TCU011

<b>Código</b>		<b>MIE-TCU011</b>
<b>Nombre</b>	Iniciar sesión con el usuario “Admin”	
<b>Descripción</b>	Este caso de prueba busca probar el comportamiento de la plataforma al iniciar sesión para un usuario con nombre “admin” y que tiene el rol de administrador.	
<b>Paquete-Modulo</b>	Sesión-Iniciar sesión	
<b>Dependencias</b>		
<b>Actores</b>	Administrador	
<b>Pre condiciones</b>	El usuario “admin” debe haber sido creado en la plataforma previamente con su respectiva contraseña “admin”.	
<b>Post condiciones</b>	El sistema permite el inicio de sesión.	
<b>Escenario principal</b>	<b>Actores</b>	<b>Sistema</b>
	Ingresar la url del portal web	Despliega la pantalla de inicio de sesión donde solicita las credenciales de usuario.
	Ingresar el nombre de usuario “admin” y la contraseña “admin”	El sistema permite el ingreso del usuario con las credenciales digitadas.
<b>Observaciones</b>		

El script diseñado para el anterior caso de prueba junto con su respectivo resultado en la herramienta Selenium es el que se muestra a continuación:

Figura 10 Script y resultado para MIE-TCU011



Los demás casos de pruebas y ejecuciones se presentan en el anexo para una mejor organización del documento.

5.2.1.2 Reporte de incidencias. A continuación se presenta el reporte de incidencia para el caso de prueba MIE-TCU011.

### MIE-TCU011: Iniciar sesión con el usuario “Admin”

Tabla 21 Reporte de incidencia en MIE-TCU0111

Código	MIE-TCU011
Reproducibilidad	A veces
Tipo de error	Menor
Prioridad	Alta
Actores	Presencia en todos los tipos de perfiles

<b>Pre condiciones</b>	Navegador Google Chrome, con un usuario ya creado en la plataforma.	
<b>Resultado obtenido.</b>	El usuario logra ingresar pero debe dar clic repetidas veces en "ingresar"	
<b>Pasos para reproducir</b>	<b>Actores</b>	<b>Sistema</b>
	El actor ingresa sus credenciales de ingreso y da clic en "ingresar"	El sistema produce un salto en la pantalla como si tratara de ingresar, pero continúa en la pantalla de inicio de sesión.
<b>Observaciones</b>		
<b>Adjuntar archivos</b>	N/A	

Figura 11 Reporte de incidencia en MIE-TCU011 en la herramienta MantisBT

☰ Ver detalles de la incidencia

ID	Proyecto	Categoría	Visibilidad	Fecha de envío	Última actualización
0000002	MIE	[Todos los proyectos] Desarrollo	público	2018-10-01 14:18	2018-10-01 14:19

<b>Informador</b>	administrator	<b>Asignada a</b>	desarrollo
<b>Prioridad</b>	alta	<b>Severidad</b>	menor
<b>Estado</b>	<span style="color: blue;">■</span> asignada	<b>Resolución</b>	abierta
<b>Plataforma</b>	Google Chrome	<b>SO</b>	Windows
<b>Reproducibilidad</b>	a veces		
<b>Versión de SO</b>	10 Pro		

**Resumen** 0000002: MIE-TCU011: Iniciar sesión

**Descripción** El usuario logra ingresar pero debe dar clic repetidas veces en "ingresar"

**Pasos para reproducir** El actor ingresa sus credenciales de ingreso en la pantalla de inicio de sesión y da clic en "ingresar". El sistema produce un salto en la pantalla como si tratara de ingresar, pero continúa en la pantalla de inicio de sesión.

**Información adicional** El error se produce cuando es la primera vez que se ingresa en un buen tiempo.... Cuando se acaba de cerrar sesión para volver a ingresar funciona normalmente

**Etiquetas** Sin etiquetas adjuntas.

**Adjuntar Etiquetas** (Separado por ',')  Etiquetas existentes ▼

Los demás reportes de incidencias se podrán consultar en el Anexo B.

5.2.1.3 Reporte de resultados. A continuación se presentan un reporte de resultados (Este se puede consultar en la herramienta de apoyo, sin embargo, se establece el siguiente reporte como opcional en caso de no contar con el anterior):

Tabla 22 Tabla de resultados ciclo 1

	<b>Código MIE</b>	<b>Resultado</b>	<b>Tipo de error</b>
<b>1</b>	TCU011	Inicio de sesión con observaciones. Es necesario dar clic en repetidas ocasiones a "ingresar".	Menor
<b>2</b>	TCU131	Exitoso.	
<b>3</b>	TCU111	Exitoso.	
<b>4</b>	TCU211	Exitoso.	

5	TCU311	Exitoso.	
6	TCU411	Exitoso.	
7	TCU511	Exitoso.	
8	TCU521	Exitoso.	
9	TCU141	No permite crear asignatura sin llenar los campos obligatorios. Sin embargo, el mensaje que se reporta no se entiende con facilidad.	Menor
10	TCU512	Crear usuario1 "Nombre1 Apellido1" permite la redundancia.	Grave
<p><b>Conclusión: De acuerdo a los resultados anteriormente obtenidos, se calcula el porcentaje de fallo como 0,4+0,4+0,8 debido a las tres incidencias presentadas; dos con errores menores y una con error grave.</b></p> <p><b>Se obtiene entonces que el primer ciclo falló 1,6 pruebas de las 10 realizadas, o bien, tiene un 16% de fallo, con lo cual se considera el primer ciclo terminado por superar el umbral del 80% de éxito establecido como mínimo.</b></p>			

**5.2.2 Segundo ciclo.** Se definen el segundo ciclo de acuerdo a las generalidades inicialmente establecidas (todas las pruebas fallidas y un 10% de las exitosas del primer ciclo, 3 de integración y 3 más de sistemas).

Tabla 23 Diseño de segundo ciclo

Tipo	Paquete	Caso de prueba		
		N	Código MIE-	Nombre
Fallidas	Sesión	1	TCU012	Iniciar Sesión como "admin"
	Asignatura	2	TCU142	Crear asignatura sin introducir ningún dato.
	Usuarios	3	TCU514	Crear usuario1 "Nombre1 Apellido1"
Exitosas	Usuarios	4	TCU515	Crear usuario2 "Nombre2 Apellido2"
Integración	Usuario	5	TCI541	Ingresar datos de tutor "usuario1"
	Asignatura	6	TCI121	Competencia de asignatura "Competencia 1" del tipo de competencia de asignatura "1 Tipo de competencia".
Sistema	Modulo	7	TCS251	Asignar tutor "Tutor1" a módulo "Modulo 1"

**5.2.2.1 Casos de prueba y ejecución.** De acuerdo a lo anteriormente establecido, se procede a diseñar los casos de pruebas en la herramienta testlink y a ejecutarse usando Selenium, por organización del presente documento, se usa el formato establecido en lugar de mostrar el diseño en testlink para los casos de uso.

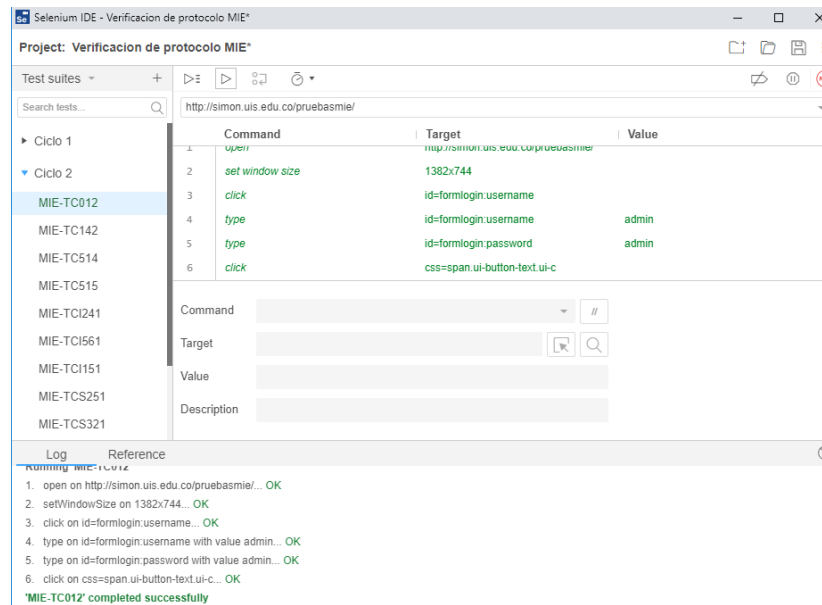
## MIE-TCU012: Iniciar sesión con el usuario “Admin”

Tabla 24 Caso de prueba MIE-TCU012

Código		MIE-TCU012
<b>Nombre</b>	Iniciar sesión con el usuario “Admin”	
<b>Descripción</b>	Este caso de prueba busca probar el comportamiento de la plataforma al iniciar sesión para un usuario con nombre “admin” y que tiene el rol de administrador.	
<b>Paquete-Modulo</b>	Sesión-Iniciar sesión	
<b>Dependencias</b>		
<b>Actores</b>	Administrador	
<b>Pre condiciones</b>	El usuario “admin” debe haber sido creado en la plataforma previamente con su respectiva contraseña “admin”.	
<b>Post condiciones</b>	El sistema permite el inicio de sesión.	
<b>Escenario principal</b>	Actores	Sistema
	Ingresa la url del portal web	Despliega la pantalla de inicio de sesión donde solicita las credenciales de usuario.
	Ingresa el nombre de usuario “admin” y la contraseña “admin”	El sistema permite el ingreso del usuario con las credenciales digitadas.
<b>Observaciones</b>		

El script diseñado para el anterior caso de prueba junto con su respectivo resultado en la herramienta Selenium es el que se muestra a continuación:

Figura 12 Script y resultado para MIE-TCU012



Los demás casos de pruebas y ejecuciones se presentan en el anexo para una mejor organización del documento.

5.2.2.2 Reporte de incidencias. A continuación se presenta el reporte de incidencias

Tabla 25 Reporte de incidencias MIE-TCU142

<b>Código</b>		<b>MIE-TCU142</b>	
<b>Reproducibilidad</b>	Siempre		
<b>Tipo de error</b>	Menor		
<b>Prioridad</b>	Normal		
<b>Actores</b>	Administrador		
<b>Pre condiciones</b>	Navegador Google Chrome, con un usuario ya loogeadado como administrador.		
<b>Resultado obtenido.</b>	La plataforma reporta un error que no es comprensible.		
<b>Pasos para reproducir</b>	<b>Actores</b>	Sistema	
	Ingresar al menú Asignatura-Asignatura curso matriz	Despliega la pantalla donde se muestran las asignaturas ya creadas.	
	Ubica la opción "Nueva asignatura" y da clic sobre ella.	El sistema despliega una ventana donde solicita diligenciar el formulario para la nueva asignatura a crear.	
	Da clic en "guardar" sin diligenciar ningún campo del formulario.	Muestra un mensaje difícil de interpretar e impide la creación de la nueva asignatura manteniendo la venta del formulario.	
<b>Observaciones</b>			
<b>Adjuntar archivos</b>	N/A		

Tabla 26 Reporte de incidencias MIE-TCU514

<b>Código</b>		<b>MIE-TCU514</b>	
<b>Reproducibilidad</b>	Siempre		
<b>Tipo de error</b>	Grave		
<b>Prioridad</b>	Normal		
<b>Actores</b>	Administrador		
<b>Pre condiciones</b>	Navegador Google Chrome, con un usuario ya loogeadado. El usuario con datos: Usuario:"Usuario1" Contraseña: "usuario1" Identificación: "12345" Nombre: "Nombre1" Apellido: "Apellido1" Email: "correo@prueba.co" Ya debió ser previamente creado.		

<b>Resultado obtenido.</b>	El usuario logra crear un nuevo usuario con los mismos datos de otro ya existente.	
<b>Pasos para reproducir</b>	<b>Actores</b>	<b>Sistema</b>
	Ingresar al menú de Usuario - Usuario	Despliega la pantalla donde se muestran los usuarios ya existentes.
	Ubica la opción "Nuevo usuario" y da clic en ella.	Despliega una ventana solicitando los datos del nuevo usuario.
	Diligencia los campos: Usuario: "Usuario1" Contraseña: "usuario1" Identificación: "12345" Nombre: "Nombre1" Apellido: "Apellido1" Email: "correo@prueba.co" Y selecciona guardar.	El sistema permite la creación del nuevo usuario con los datos del anterior usuario.
<b>Observaciones</b>		
<b>Adjuntar archivos</b>	N/A	

5.2.2.3 Reporte de resultados. A continuación se presentan un reporte de resultados del segundo ciclo

Tabla 27 Reporte de resultados segundo ciclo

<b>Código MIE</b>		<b>Resultado</b>	<b>Tipo de error</b>
<b>1</b>	TCU012	Exitoso, pese a no haberse realizado ajuste alguno	
<b>2</b>	TCU142	No permite crear asignatura sin llenar los campos obligatorios. Sin embargo, el mensaje que se reporta no se entiende con facilidad.	Menor
<b>3</b>	TCU514	Crear usuario1 "Nombre1 Apellido1" permite la redundancia de nuevo.	Menor
<b>4</b>	TCU515	Exitoso.	
<b>5</b>	TCI541	Exitoso.	
<b>6</b>	TCI121	Exitoso.	
<b>7</b>	TCS251	Exitoso.	
<b>Conclusión: De acuerdo a la anterior ejecución, se cuenta con un porcentaje de error del 0,4 para dos de las 7 pruebas aplicadas, por tanto, se tiene un porcentaje de error del 0,8 de 7, es decir 11,4%.</b>			

### 5.3 PROTOCOLO DE PRUEBAS REQUERIMIENTOS NO FUNCIONALES

**5.3.1 Métricas para pruebas de carga y rendimiento.** Las métricas a usar durante estas pruebas son las mismas definidas en el apartado de **Métricas para pruebas de carga y rendimiento.**

**5.3.2 Pruebas de Rendimiento.** Para este tipo de pruebas se usará la herramienta Jmeter como se indicó y se configurará inicialmente dos pruebas a una carga normal de 120 usuarios para dos casos de uso que se consideran son de alta concurrencia para el rol de estudiante:

MIE-TCR120.1: Ver recurso asignado a una actividad “Plantilla.doc”

MIE-TCR120.2: Participar en foro “foro de prueba”

Los scripts de prueba diseñados junto con sus casos de prueba y resultados se muestran a continuación:

**MIE-TCR120.1: Ver recurso asignado a una actividad “plantilla.doc”**

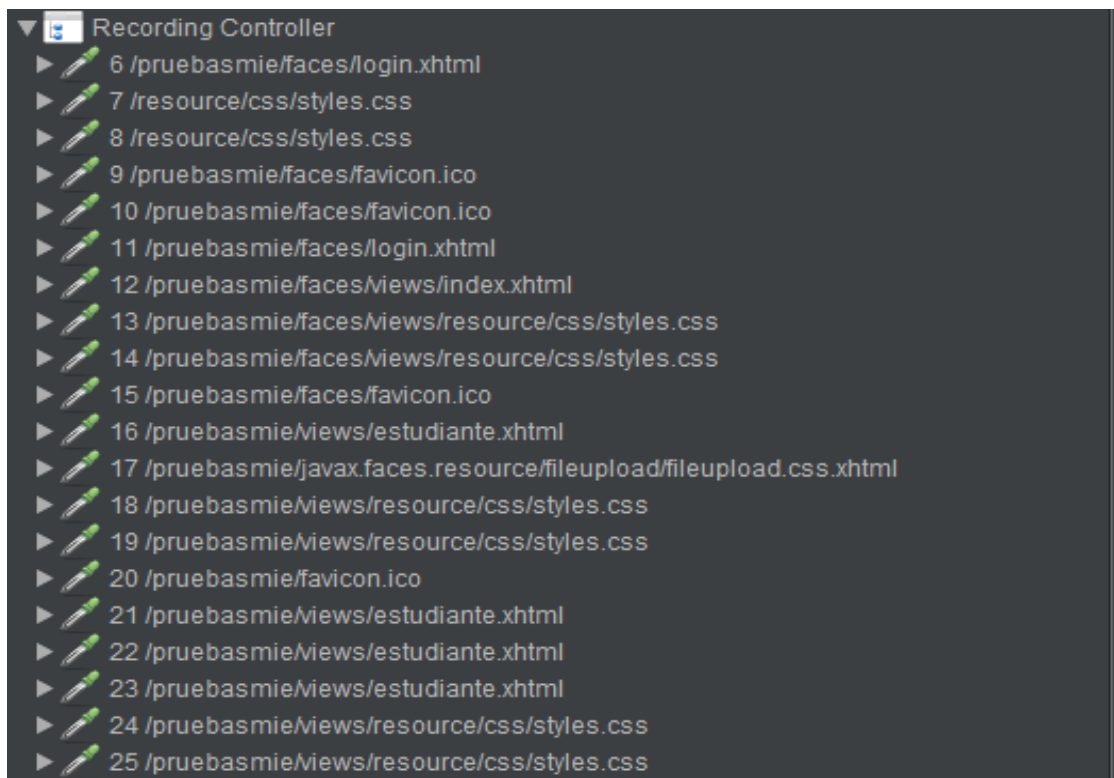
Tabla 28 Ejemplo 1 Caso de prueba de rendimiento.

<b>Código</b>		<b>MIE-TCR120.1</b>
<b>Nombre</b>	Ver recurso asignado: “plantilla.doc” de la actividad “Actividad en prueba” asignados a al módulo “modulo en prueba” de la Asignatura “Asignatura en prueba”.	
<b>Descripción</b>	Se pone a prueba el rendimiento de la plataforma cuando 120 usuarios de forma concurrente consultan el recurso “plantilla.doc” asignado a la “actividad en prueba” del “modulo en prueba”.	
<b>Paquete-Módulo</b>	Estudiante – Mis Cursos	
<b>Pre condiciones</b>	El sistema debe haber sido lanzado en ambiente de pruebas. El estudiante debe haber iniciado sesión satisfactoriamente. Debe existir una asignatura llamada “asignatura en prueba”. El estudiante debe tener asignada la “asignatura en prueba”. La “asignatura en prueba” debe tener un módulo llamado “módulo en prueba” con una actividad llamada “actividad en prueba”. El recurso “plantilla.doc” debe estar asignado a la “actividad en prueba”.	
<b>Post condiciones</b>	El estudiante debe haber accedido al recurso “plantilla.doc” de la “actividad en prueba”.	
<b>Script de prueba</b>	<ul style="list-style-type: none"> <li>- El estudiante se dirige al paquete “estudiante” del menú principal.</li> <li>- Selecciona el Módulo “mis cursos” y en la tabla que se despliega ubica la “Asignatura en prueba” y selecciona dentro de su respectiva columna de opciones: “Ver actividades de módulos”.</li> <li>- En la tabla que se despliega de Actividades busca la “actividad en prueba” y en su respectiva columna de opciones da clic en “ver detalles”.</li> <li>- En la nueva ventana que se genera ubica el link de “recursos asociados” y da clic en él.</li> <li>- Selecciona el recurso “plantilla.doc”.</li> </ul>	

<b>Configuración del script de prueba.</b>	<ul style="list-style-type: none"> <li>- Generar 120 solicitudes simultaneas en un espacio de 10 segundos</li> <li>- Medir cuantas solicitudes fueron exitosas (unidades)</li> <li>- Medir el tiempo promedio de las solicitudes (seg)</li> <li>- Medir el tiempo total de ejecución (min)</li> <li>- Medir el throuput de la funcionalidad (solicitudes/min)</li> </ul>
<b>Observaciones</b>	

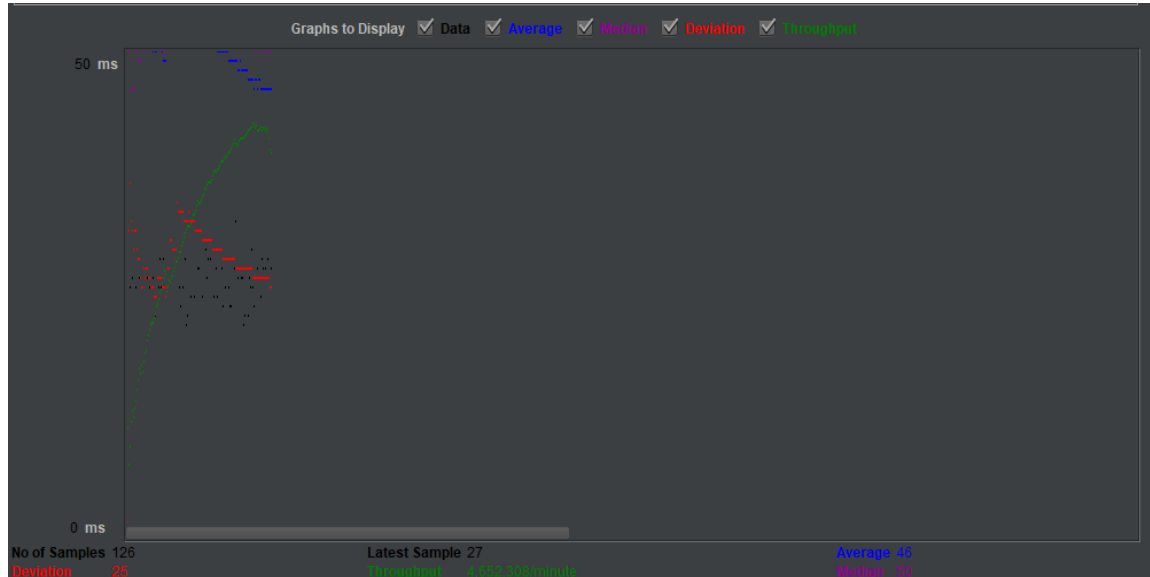
## Script de prueba en Jmeter

Figura 13 Script de prueba en Jmeter para MIE-TCR120.1



## Resultado de la prueba

Figura 14 Resultado gráfico de prueba en Jmeter para MIE-TCR120.1



## MIE-TCR120.5: Participar en foro “foro de prueba”

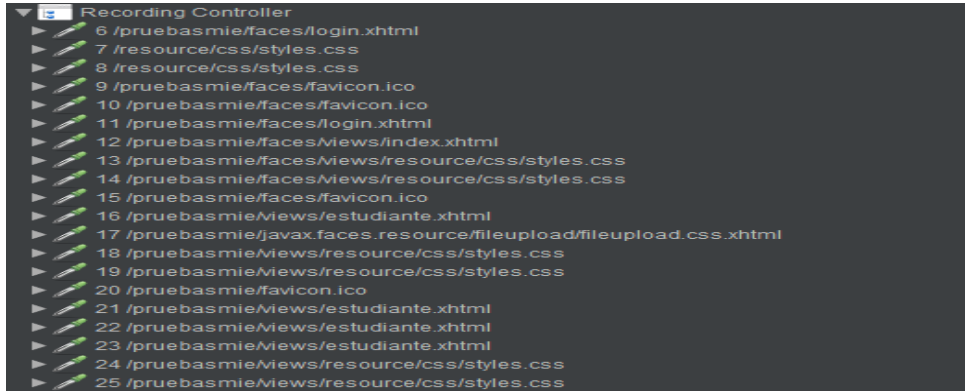
Tabla 29 Ejemplo 2 Caso de prueba de rendimiento.

Código	MIE-TCR120.5
<b>Nombre</b>	Participar en un foro de prueba
<b>Descripción</b>	Se pone a prueba el rendimiento de la plataforma cuando 120 usuarios de forma concurrente participan en un foro “foro de prueba” del cual son participantes.
<b>Paquete-Módulo</b>	Recursos – Foros
<b>Pre condiciones</b>	El sistema debe haber sido lanzado en ambiente de pruebas. El estudiante debe haber iniciado sesión satisfactoriamente. Debe existir un foro activo llamado “foro de prueba”. El estudiante debe ser participante del foro “foro en prueba”.
<b>Post condiciones</b>	El foro debe contener el comentario “un comentario de prueba”.
<b>Script de prueba</b>	<ul style="list-style-type: none"> <li>- El estudiante se dirige al ícono superior derecho de su pantalla inicial y da clic sobre él.</li> <li>- En el menú que se despliega selecciona el foro denominado “foro de prueba”</li> <li>- En la ventana que se despliega ubica la opción “Nuevo comentario principal” y da clic en ella.</li> <li>- En la ventana que se muestra digita: “Este es un comentario principal” y da clic en guardar.</li> </ul>
<b>Configuración del script de prueba.</b>	<ul style="list-style-type: none"> <li>- Generar 120 solicitudes simultaneas en un espacio de 10 segundos</li> <li>- Medir cuantas solicitudes fueron exitosas (unidades)</li> </ul>

	<ul style="list-style-type: none"> <li>- Medir el tiempo promedio de las solicitudes (seg)</li> <li>- Medir el tiempo total de ejecución (seg)</li> <li>- Medir el throuput de la funcionalidad (solicitudes/min)</li> </ul>
<b>Observaciones</b>	

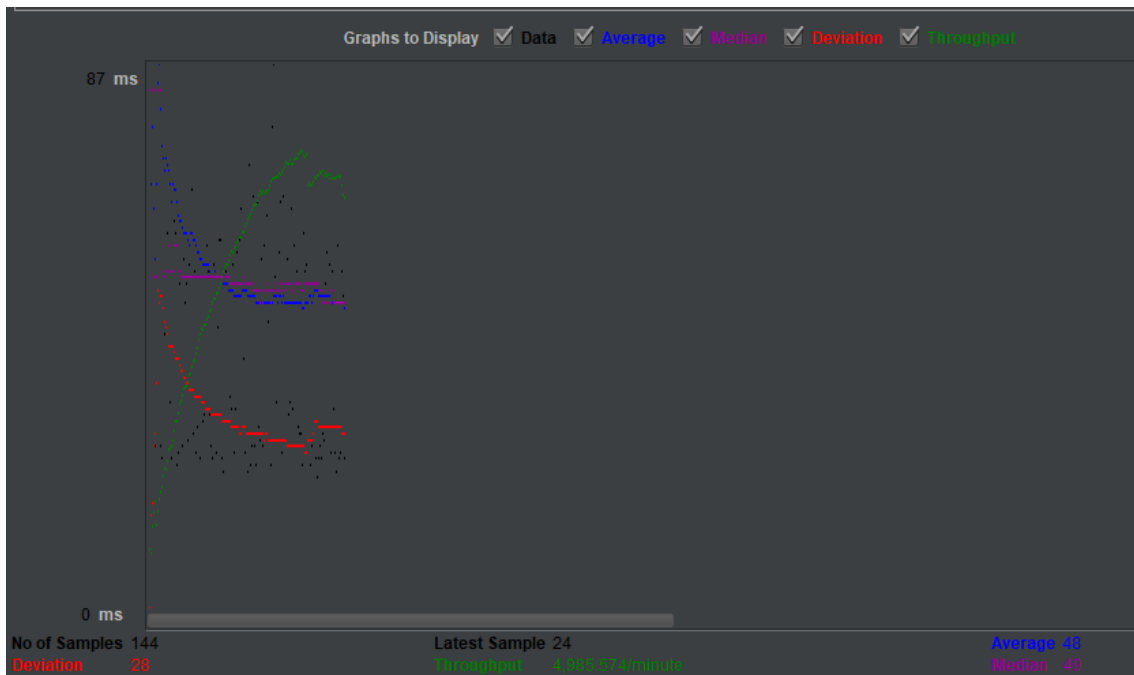
## Script de prueba en Jmeter

Figura 15 Script de prueba en Jmeter para MIE-TCR120.2



## Resultado de la prueba

Figura 16 Resultado gráfico de prueba en Jmeter para MIE-TCR120.2



**5.3.3 Pruebas de Carga.** Para este tipo de pruebas se usará la herramienta Jmeter como se indicó y se configurará tres pruebas a una carga que va desde 80, luego 120 y finalmente 250 usuarios para un caso de uso que se consideran de alta concurrencia para el rol de estudiante:

### **MIE-TCCxxx.1: Ver recurso asignado a una actividad “Plantilla.doc”**

Los scripts de prueba son los mismos diseñados para las pruebas de rendimiento, sin embargo la configuración de la prueba aumenta gradualmente la carga de acuerdo a lo establecido en las métricas, los resultados se muestran a continuación:

Figura 17 Resultados en Jmeter para MIE-TCC80.1

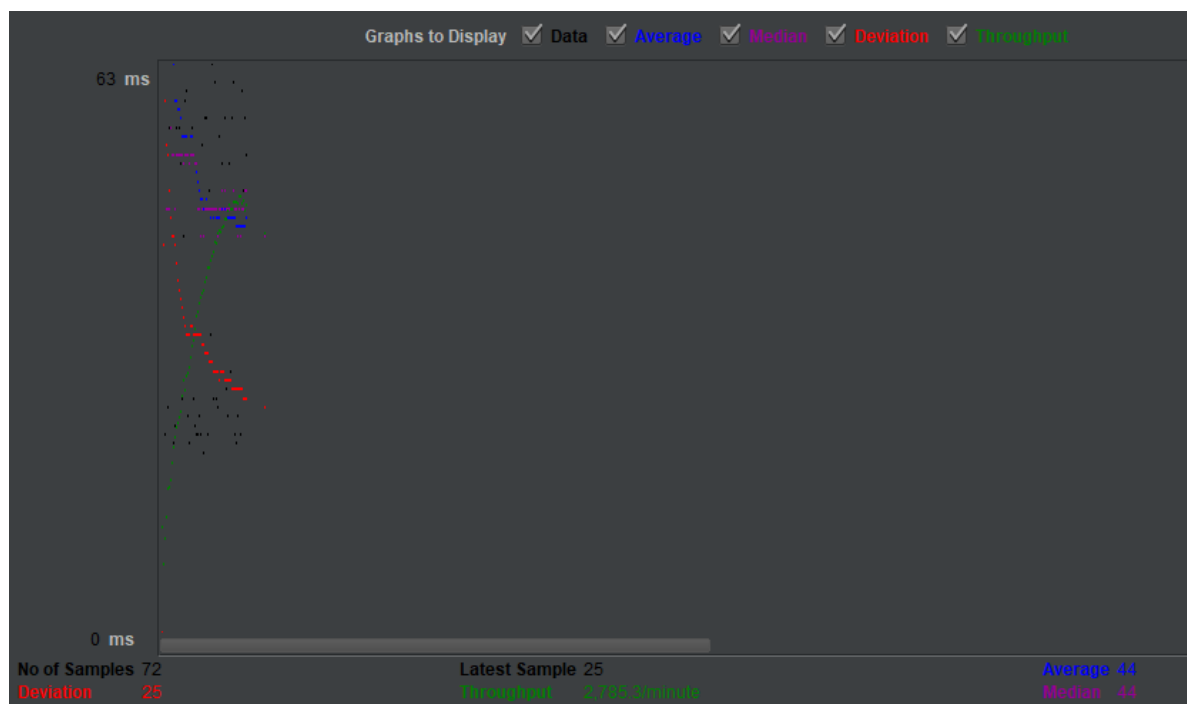


Figura 18 Resultados prueba de carga MIE-TCC120.1

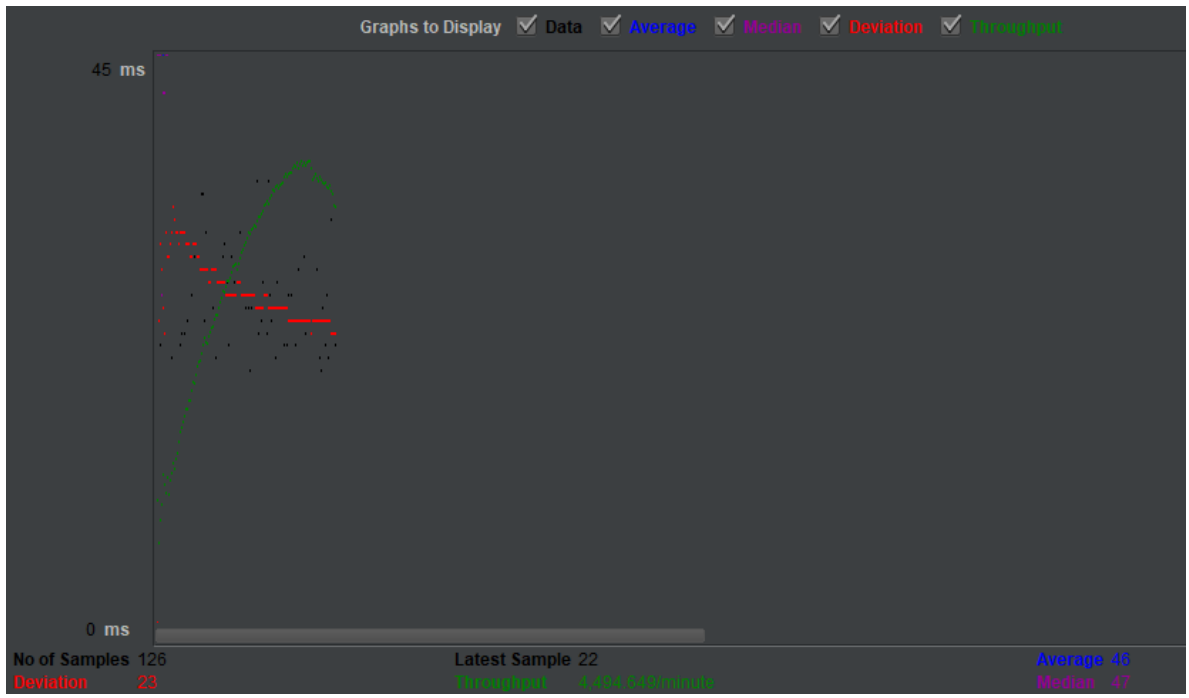
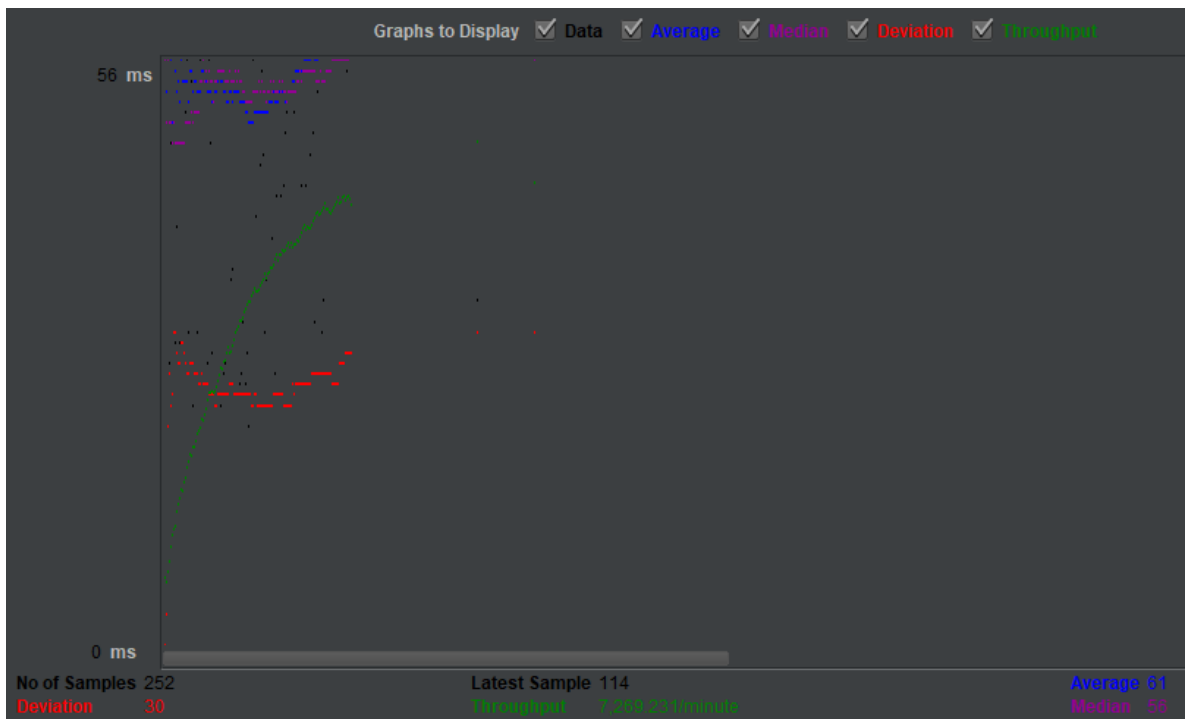



Figura 19 Resultados MIE-TCC250.1



**5.3.4 Pruebas de Usabilidad.** Con el fin de probar la usabilidad de la plataforma MIE se diseñan las siguientes encuestas a aplicarse a 10 estudiantes al azar.

Tabla 30 Encuesta prueba de usabilidad

ENCUESTA PRUEBA DE USABILIDAD	Formulario N° PU__
<p>Agradecemos su participación en la presente encuesta, sus respuestas son anónimas, sólo serán usadas con fines estadísticos y son muy importantes en nuestro proceso de mejoramiento continuo. Por favor, sea tan objetivo, claro y específico como sea posible, en caso de requerir de más espacio del suministrado, escriba en la parte trasera de la hoja iniciando con el número de pregunta a la cual está respondiendo.</p>	
<p><b>Para las siguientes preguntas, por favor, abra el navegador de su preferencia e ingrese al siguiente portal web o escanee el código QR de la derecha:</b>  <a href="http://simon.uis.edu.co/pruebasmie/">http://simon.uis.edu.co/pruebasmie/</a></p>	
<p>Navegador usado: _____</p>	<p>Dispositivo usado: _____</p>
<p><b>A partir de lo que logra visualizar inicialmente en su navegador, sin desplazarse por los menús:</b></p>	
<p>1 - ¿A cuál institución considera usted que pertenece el sitio web? ¿Por qué?</p> <p>_____</p> <p>_____</p> <p>_____</p>	
<p>2 - ¿Hay algún elemento que considere fuera de lugar o que considere que no corresponde al sitio web?</p> <p>_____</p> <p>_____</p> <p>_____</p>	
<p>3 - ¿Relaciona los colores predominantes con la imagen de la institución?</p> <p>_____</p> <p>_____</p> <p>_____</p>	
<p>4 - ¿Relaciona la dirección del sitio web con la institución?</p> <p>_____</p> <p>_____</p> <p>_____</p>	
<p>5 - ¿Hacia qué tipo de audiencia cree que está dirigido este sitio web? Explique su respuesta.</p> <p>_____</p> <p>_____</p> <p>_____</p>	
<p>6 - En caso de requerir información adicional o necesitar ponerse en contacto con la institución: ¿Puede encontrar fácilmente la información de soporte y/o contacto? ¿Considera que es suficiente la información ofrecida por el sitio web? ¿Pudo encontrarla fácilmente?</p> <p>_____</p>	

<p>_____</p> <p>_____</p>
<p><b>A continuación, por favor proceda a navegar por el sitio web y observe atentamente la forma como está diseñado y responda:</b></p>
<p>7 - ¿Qué opina sobre la portada del sitio?</p> <p>_____</p> <p>_____</p> <p>_____</p>
<p>8 - ¿Considera que la información contenida es excesiva, suficiente, escasa o incomprensible?</p> <p>_____</p> <p>_____</p> <p>_____</p>
<p>9 - ¿Cuál considera que es el contenido más relevante y por qué?</p> <p>_____</p> <p>_____</p> <p>_____</p>
<p>10 - ¿Los textos usados para nombrar los enlaces brindan información acerca de la página a la que dirigen?</p> <p>_____</p> <p>_____</p> <p>_____</p>
<p>11 - ¿Logra distinguir fácilmente en cada página la forma como se navega por el sitio?</p> <p>_____</p> <p>_____</p> <p>_____</p>
<p>12 - ¿Considera sencillo ir de sección en sección de la página? ¿Se siente desubicado en algún momento?</p> <p>_____</p> <p>_____</p> <p>_____</p>

Observaciones y sugerencias: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

<b>HALLAZGOS PRUEBA DE USABILIDAD</b>	<b>Informe MIE-RPU01</b>
El presente resumen contiene un compendio de los hallazgos de las encuestas de usabilidad aplicadas.	
Navegadores usados: 1. Google Chrome 2. Mozilla Firefox	Dispositivos usados: 1. Computadores 2. Tablets 3. Celular

<b>Las siguientes preguntas buscaban evaluar la imagen corporativa del sitio web:</b>	
1 - ¿A cuál institución considera usted que pertenece el sitio web? ¿Por qué?	Dado que se cuenta con una fotografía de portada de la UIS y el logotipo de la Universidad, fácilmente se relaciona con la institución.
2 - ¿Hay algún elemento que considere fuera de lugar o que considere que no corresponde al sitio web?	Se evidencia que todos los elementos parecen estar en su lugar en cada uno de los dispositivos cumpliendo con muy buen diseño responsivo.
3 - ¿Relaciona los colores predominantes con la imagen de la institución?	Los colores predominantes si se relación con la institución.
4 - ¿Relaciona la dirección del sitio web con la institución?	Dado que termina en "uis.edu.co" se relaciona de inmediato con la UIS.
5 - ¿Hacia qué tipo de audiencia cree que está dirigido este sitio web? Explique su respuesta.	Dado que el menú inicial aparece con la palabra "estudiante" todos los encuestados coinciden en que se trataba de una página dirigida a estudiantes.
6 - En caso de requerir información adicional o necesitar ponerse en contacto con la institución: ¿Puede encontrar fácilmente la información de soporte y/o contacto? ¿Considera que es suficiente la información ofrecida por el sitio web? ¿Pudo encontrarla fácilmente?	Se pone en evidencia que la página no cuenta con información de contacto, más allá de un hipervínculo que redirige a la página principal de la UIS
<b>Las siguientes preguntas pretendían evaluar la</b>	
7 - ¿Qué opina sobre la portada del sitio?	Se evidencia que es una portada muy sobria y agradable que da una buena primera impresión comunicando el tipo de página en la que se navega (educativa).
8 - ¿Considera que la información contenida es excesiva, suficiente, escasa o incomprensible?	Algunas personas consideraron escasa la información, pero en general, suficiente para comunicar la idea que se pretende.
9 - ¿Cuál considera que es el contenido más relevante y por qué?	Todos los encuestados consideraron que "Mis cursos" era la información más relevante ya que contenía más información y con mayor importancia.
10 - ¿Los textos usados para nombrar los enlaces brindan información acerca de la página a la que dirigen?	Todos los encuestados consideraron que los enlaces contienen palabras claves que dan una idea sobre el mismo.
11 - ¿Logra distinguir fácilmente en cada página la forma como se navega por el sitio?	

Los encuestados consideraron que se trataba de un sitio web sencillo y que el menú principal que aparece en él, permite navegar fácilmente por la página.
12 - ¿Considera sencillo ir de sección en sección de la página? ¿Se siente desubicado en algún momento?  Solo un usuario reportó que las diversas tablas que se despliegan en detalles desde el menú “mis cursos” lo hizo sentir desubicado.

Comentarios: Algunos usuarios reportaron que para iniciar sesión fue necesario dar varias veces clic en “ingresar”

**5.3.5 Pruebas de compatibilidad.** Las pruebas de compatibilidad se aplicaron en los navegadores y dispositivos usados durante las pruebas de usabilidad:

Google Chrome y Mozilla Firefox, la navegación y uso de la página se mantuvieron constantes, sin novedad cuando se navega desde computadores y tablets.

Sin embargo, la visualización de las diferentes vistas como tablas es muy difícil y en ocasiones imposible cuando se realiza en celulares. Los resultados se muestran en la siguiente tabla:

Tabla 31 Resultados prueba de compatibilidad

<b>COMPATIBILIDAD</b>	Google Chrome	Mozilla Firefox	Microsoft Edge
Computador	Compatible	Compatible	Compatible
Tablet	Compatible	Compatible	N/A
Celular	Incompatible	Incompatible	N/A

## 6. CONCLUSIONES

1. Se evidencia la importancia de la organización de las pruebas en un protocolo que claramente establezca una metodología de prueba a aplicarse tanto en las pruebas funcionales como no funcionales. Contar con un protocolo hecho a la medida agiliza el trabajo tanto en ejecución como en el reporte y resolución de errores para los diferentes actores que participan en el proceso de vida del software.
2. Se concluye que la utilización de las diferentes herramientas de automatización son un mecanismo de gran ayuda, esencial para el diseño, ejecución y documentación del presente protocolo de pruebas siendo el soporte del mismo.
3. El diseño de un protocolo de pruebas debe contar con un ciclo de validación que permita realizar ajustes a lo inicialmente diseñado. Pese a que inicialmente pueda considerarse completado, la etapa de validación es un gran aporte que no se puede pasar por alto porque aporta en el aseguramiento de la calidad del producto.
4. El presente protocolo se ha diseñado y validado para la plataforma MIE-UIS, sin embargo, la metodología propuesta puede ser usada en otros aplicativos web con características similares ajustando las métricas y algunos otros parámetros al nuevo sistema.

## 7. RECOMENDACIONES

### **Recomendaciones de diseño para el diseñador de casos de prueba**

- El encargado de diseñar el caso de prueba deberá en todo momento tener en cuenta las condiciones límite o de frontera del módulo que está probando, es decir, diseñar pruebas en las que evalúe el comportamiento del módulo en prueba cuando se introducen datos que están en sus límites, por ejemplo, crear un caso de prueba en el que se cree un grupo con cero estudiantes o una asignatura sin propósito o una competencia con más de un tipo de competencia. Se debe buscar siempre evaluar la forma como el módulo opera las excepciones, homonimias y los datos máximos o mínimos que se espera, acepte o no de acuerdo al análisis de requerimientos y funcionamiento esperado del sistema.
- Para estudiar el manejo de excepciones se recomienda diseñar casos de uso en los que no se introduzca ningún dato y verificar si los campos que la plataforma marca como obligatorios son los deseados.

### **Verificar Manejo de errores**

Un correcto diseño debe predecir los errores y establecer las rutas de manejo para que el proceso termine de ejecutarse limpiamente. A esto se le conoce como manejo de errores. Para probar esta característica es importante buscar la forma de que se den los errores y observar el comportamiento del sistema: si muestra algún mensaje, si el mensaje es legible y fácilmente interpretable, si en verdad corresponde al tipo de error que se presenta, entre otras.

### **Pruebas de recuperación**

Muchos sistemas son concebidos desde un inicio para ser tolerante a fallas, es decir, que el sistema debe recuperarse de fallas con poco o ningún tiempo de inactividad. Si se presenta un fallo en el procesamiento de este tipo de sistemas,

debe tener la capacidad de continuar funcionando a riesgo de daños de diversas índoles permitiendo poner al descubierto posibles problemas o defectos que se presentan durante la falla del sistema. Esto es, evidenciar que no se presente pérdida de datos o servicios, que no cause corrupción de datos...etc.

Dicho lo anterior, una prueba de recuperación se diseña para hacer que el sistema falle y evaluar si el tiempo de recuperación se encuentra dentro de los parámetros permitidos. El alcance del presente documento no contempla este tipo de pruebas ya que los requerimientos del sistema no contemplan el tener que soportar grandes cargas que lo lleven a un punto de estrés, sin embargo, si el equipo de calidad lo considera necesario o el sistema se escala por encima de lo contemplado en sus requerimientos iniciales, se podrán ejecutar pruebas de este tipo usando alguna herramienta que lleve al fallo el sistema como Jmeter rediseñando el caso de prueba para que lleve el sistema por encima de sus límites.

### **Pruebas de Rendimiento y carga a otros roles de usuario**

Aunque las pruebas de carga y rendimiento se centran sobre las funcionalidades que realizan los estudiantes, por las razones ya expuestas; se podrían evaluar el rendimiento de la plataforma en las funcionalidades correspondientes a otros roles de usuario de forma manual, mientras se ejecuta un script de prueba automatizado para medir rendimiento o carga para evaluar el rendimiento de la plataforma en funcionalidades poco concurridas como, por ejemplo, crear un usuario, en eventos en los que el aplicativo esté siendo usado en otras funcionalidades más concurridas como la consulta de actividades de un módulo.

Para lo anterior, se debe establecer una prueba de rendimiento o carga automatizada y medir los resultados de la prueba automatizada sin realizar ninguna otra solicitud. Posteriormente ejecutar nuevamente la prueba automatizada y mientras ella se ejecuta, probar la funcionalidad del otro rol de

forma paralela. Finalmente comparar los resultados de las dos pruebas con el fin de establecer el comportamiento de la plataforma durante la ejecución de la funcionalidad no concurrente mientras hay un alto flujo de peticiones a una funcionalidad concurrente.

### **Instrumentación del ambiente de pruebas por parte del equipo de desarrollo**

El presente protocolo no tiene en cuenta pruebas de caja blanca dado que está destinado principalmente al equipo de calidad en una perspectiva de cliente. Sin embargo, en caso de que el equipo de desarrollo desee replicar las pruebas y requiera más información detallada al respecto de los fallos, se recomienda Instrumentar el ambiente de pruebas con herramientas que permitan analizar los diferentes componentes que hacen parte de él y de esa manera visualizar la forma como se usan los recursos, para determinar si el fallo presentado se encuentra en la base de datos, en el servidor de aplicaciones, etc.

## BIBLIOGRAFÍA

Agarwal, B. T. (2009). Software Engineering & Testing: An Introduction. Massachusetts: Jones and Bartlett Publishers.

Binder, R. (2000). Testing Object-Oriented Systems: Models, Patterns, and Tools. Addison-Wesley.

CNN. (30 de Septiembre de 1999). Metric mishap caused loss of NASA orbiter. CNN.

Dowson, M. (Marzo de 1997). The Ariane 5 Software Failure. Software Engineering Notes, pág. 84.

González Montoya, P. A. (2012). Caracterización de un protocolo de pruebas. Santiago de Cali: Universidad de San Buenaventura.

Hetzel, B. (1988). The complete guide to Software Testing. Information Sciences Inc.

Humphrey, W. S. (1989). Managing The Software Process .

IEEE. (1990). Standard Glossary of Software Engineering Terminology. . IEEE Standars Board.

Patton, R. (2005). Software Testing, Second Edition. USA: Sams Publishing.

Roger, P. (2002). Ingeniería del Software - Un enfoque práctico. México: McGraw-Hill.

Somerville, I. (2011). Ingeniería del Software. México: PEARSON EDUCATION.