

DISEÑO Y EJECUCIÓN DE PRUEBAS DE SOFTWARE PARA EL
SISTEMA DE INFORMACIÓN INTEGRADO: "COMUNIDAD COLCIENCIAS
– FASE 1" DEL DEPARTAMENTO ADMINISTRATIVO DE CIENCIA,
TECNOLOGÍA E INNOVACIÓN COLCIENCIAS

Jhorman Ferney Izaquita Gualdrón

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2016

DISEÑO Y EJECUCIÓN DE PRUEBAS DE SOFTWARE PARA EL
SISTEMA DE INFORMACIÓN INTEGRADO: "COMUNIDAD COLCIENCIAS
– FASE 1" DEL DEPARTAMENTO ADMINISTRATIVO DE CIENCIA,
TECNOLOGÍA E INNOVACIÓN COLCIENCIAS

Jhorman Ferney Izaquita Gualdrón

Proyecto de grado en modalidad de práctica empresarial para optar al título
de Ingeniero de Sistemas

Director:

PhD. Gabriel Rodrigo Pedraza

Tutor:

PhD. Homero Ortega Boada
Director Científico del Proyecto

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECHANICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2016

CONTENIDO

	Pág
INTRODUCCION.....	13
1. PRESENTACIÓN DEL PROYECTO.....	14
1.1 DESCRIPCIÓN DEL PROYECTO.....	14
1.1.1 Título.....	14
1.1.2 Objetivo general:	14
1.1.3 Objetivos Específicos	14
1.1.4 Indicadores de logros y objetivos	15
1.2 JUSTIFICACIÓN.....	17
1.2.1 Antecedentes y descripción del proyecto.....	17
1.2.2 Impacto.....	18
1.2.3 Viabilidad	19
2. MARCO CONCEPTUAL	21
2.1 MARCO DE REFERENCIA.....	21
2.1.1 Breve marco de referencia del grupo de investigación CIDLIS	21
2.1.2 Proyecto de Servicio de Pruebas de software del contrato: "la construcción del sistema de información integrado: "comunidad Colciencias – fase 1" y su puesta en servicio, mantenimiento y soporte.	22
2.2 MARCO TEÓRICO.....	24
2.2.1 Pruebas de software	24
2.2.2 Principios de las pruebas	26
2.2.3 ISO 9126 El estándar ISO 9126	28
2.2.4 Niveles de pruebas	30
2.2.4.1 Pruebas de componente	31
2.2.4.2 Pruebas de integración.....	32
2.2.4.3 Pruebas de sistema	32
2.2.4.4 Pruebas de aceptación.....	33

2.2.5	Tipos de pruebas	34
2.2.5.1	Pruebas funcionales	34
2.2.5.2	Pruebas no funcionales	36
2.2.5.3	Pruebas estructurales	37
2.2.5.4	Pruebas de regresión	37
3.	PLAN DE TRABAJO	39
3.1	CARGO INICIAL Y RESPONSABILIDADES.....	39
3.2	PRIMEROS RESULTADOS.....	40
4.	CICLO DE VIDA DE PRUEBAS	42
4.1	PLANIFICACIÓN.....	42
4.2	DESARROLLO.....	43
4.3	ESTRATEGIA DE PRUEBA.....	43
4.4	METODOLOGÍA.....	46
4.5	VALIDACIÓN Y VERIFICACIÓN.....	47
4.6	EJECUCIÓN DE CASOS DE PRUEBA	48
4.7	ANÁLISIS Y DISEÑO DE CASOS DE PRUEBA	50
5.	DEFECTOS.....	53
5.1	ESTADOS DE LOS DEFECTOS.....	53
6.	RESULTADOS.....	55
6.1	DEFECTOS ENCONTRADOS.....	56
6.2	EJECUCIONES DE PRUEBA.....	57
6.3	DISEÑOS DE CASOS DE PRUEBA.....	58
7.	CONCLUSIONES.....	61
8.	RECOMENDACIONES	62
	BIBLIOGRAFIA.....	63

LISTA DE FIGURAS

Figura 1 ciclo en la estrategia de pruebas	30
Figura 2 Niveles de pruebas	31
Figura 3 Modelo de pruebas de Caja Negra	36
Figura 4 Proceso de Depuración	42
Figura 5 Ciclo de pruebas. Ingeniería de software, Un enfoque práctico 7ma edición- Roger S. Pressman	44
Figura 6 Metodología de desarrollo de la práctica	46
Figura 7 Relación de casos de prueba ejecutados y estados reportados.....	57
Figura 8 Resultados de ejecución	58

LISTA DE TABLAS

Tabla 1 Indicadores de logros y objetivos	16
Tabla 2 Distribución de casos de uso	24
Tabla 3 Modelo de presentación de un caso de uso en un documento de especificación	35
Tabla 4 Esquema del documento empleado para el diseño de casos de prueba..	51
Tabla 5 Clasificación de la severidad de las incidencias	54
Tabla 6 Resumen de incidencias reportadas en la herramienta de gestión de defectos mantis (Tomada directamente de la herramienta)	56
Tabla 7 Resultados Obtenidos	59
Tabla 8 Resultados Obtenidos	60
Tabla 9 Diseño de Caso de prueba	64

LISTA DE ANEXOS

Anexo A.....	64
--------------	----

RESUMEN

TÍTULO:

DISEÑO Y EJECUCIÓN DE PRUEBAS DE SOFTWARE PARA EL SISTEMA DE INFORMACIÓN INTEGRADO: “COMUNIDAD COLCIENCIAS – FASE 1” DEL DEPARTAMENTO ADMINISTRATIVO DE CIENCIA, TECNOLOGÍA E INNOVACIÓN COLCIENCIAS.

AUTOR: Jhorman Ferney Izaquita Gualdrón **

PALABRAS CLAVE: Aseguramiento de la calidad, COLCIENCIAS, pruebas de software, ISTQB, analista de pruebas.

DESCRIPCIÓN:

El presente documento es un completo informe de la práctica empresarial llevada a cabo en UIS-COLCIENCIAS, en el cual se desempeñó el cargo de analista de pruebas de sistemas de software, formando parte del proyecto SERVICIOS DE PRUEBAS DE SOFTWARE DEL CONTRATO: "LA CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN INTEGRADO: “COMUNIDAD COLCIENCIAS – FASE 1” Y SU PUESTA EN SERVICIO, MANTENIMIENTO Y SOPORTE. Allí se llevaron a cabo labores tales como planeación, estimación, diseño y ejecución de estrategias y casos de prueba durante los cuatro meses que el estudiante estuvo realizando la práctica.

Los procesos de aseguramiento de la calidad en sistemas software es una inversión cada vez más fuerte dentro de las compañías de todos los sectores comerciales y de cualquier tamaño corporativo. La detección temprana de fallos en sistemas informáticos significa un beneficio económico considerable para las empresas, así como un voto de confianza para sus clientes ofreciendo productos y servicios óptimos.

El practicante participó profesionalmente en el proyecto, adquiriendo y aplicando conocimientos en extracción de requerimientos funcionales, diseño de estrategias de pruebas, ejecución de casos de pruebas y análisis de resultados obtenidos por medio de las ejecuciones. De igual manera el estudiante logró ser un miembro activo del equipo de trabajo UIS cumpliendo con las exigencias que el cargo le generaba.

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática
Director: Gabriel Rodrigo Pedraza, Tutor: Homero Ortega Boada.
Convenio UIS-COLCIENCIAS.

ABSTRACT

TITLE: DESIGN AND IMPLEMENTATION OF SOFTWARE TESTING FOR INTEGRATED INFORMATION SYSTEM: "COMMUNITY COLCIENCIAS - PHASE 1" ADMINISTRATIVE DEPARTMENT OF SCIENCE, TECHNOLOGY, AND INNOVATION COLCIENCIAS

AUTOR: Jhorman Ferney Izaquita Gualdrón **

KEYWORDS: Quality assurance, Colciencias, Software Testing, ISTQB, test analyst

DESCRIPTION:

This document is a detailed report of business practice carried out in UIS COLCIENCIAS, in which the position of analyst testing software systems served as part of the project SERVICIOS DE PRUEBAS DE SOFTWARE DEL CONTRATO: "LA CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN INTEGRADO: "COMUNIDAD COLCIENCIAS – FASE 1" Y SU PUESTA EN SERVICIO, MANTENIMIENTO Y SOPORTE. There were carried out tasks such as planning, assessment, design and implementation of strategies and test cases During the four months that the student was doing the practice.

The processes of quality assurance in software systems is an investment increasingly strong within companies of commercial sectors and companies with any size corporate. Early detection of faults in computer systems means a considerable economic benefit for businesses, as well as a vote of confidence for its customers by offering optimal products and services.

The practitioner professionally involved in the project, acquiring and applying knowledge extraction of functional requirements, design testing strategies, test case execution, and analysis of results obtained through executions. In the same way the student obtained an active member of the UIS work team fulfilling the demands that the charge generated him.

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática
Director: Gabriel Rodrigo Pedraza, Tutor: Homero Ortega Boada.
Convenio UIS-COLCIENCIAS.

INTRODUCCIÓN

Las grandes compañías requieren innovar en los productos que ofrecen a sus actuales y potenciales clientes, para ello realizan fuertes inversiones en el desarrollo de nuevos productos que les permitan alcanzar un alto nivel de competitividad en sus respectivos mercados. Pero no basta enfocarse en adquirir los servicios de más alta gama de distinguidos proveedores, se hace estrictamente necesario poner a prueba el producto antes de ponerlo a disposición para su uso operativo.

Los sistemas de software forman parte de la vida de la mayoría de personas, si uno de estos sistemas no funciona como se debería hacerlo, su impacto negativo se vería reflejado en grandes proporciones. Un error humano puede desencadenar una serie de fallos que potencialmente afectaría a un gran número de personas, por ejemplo, en las aplicaciones comerciales como la banca, aplicaciones de transacciones electrónicas, telemedicina, transporte aéreo, etc. Es por eso que es de vital importancia realizar grandes esfuerzos en implementar mecanismos para la detección de estos defectos en el software que se planea implementar y para ello se demandan grandes recursos para reducir los riesgos en la puesta en producción de los sistemas.

El proyecto Servicios de pruebas de software del contrato: "la construcción del sistema de información integrado: "comunidad Colciencias – fase 1" y su puesta en servicio, mantenimiento y soporte, ha sido para el grupo de investigación CIDLIS-UIS un proyecto de gran importancia dada su relevancia, magnitud e impacto a nivel nacional. Las actividades de pruebas enmarcadas dentro del proyecto son realizadas en las instalaciones del Departamento Administrativo de Ciencia, Tecnología e Innovación COLCIENCIAS, contando con un equipo de trabajo en total de 10 personas las cuales asumen los siguientes cargos: Gerente

de proyecto, Asegurador de Calidad, Líder de Pruebas, Analistas de Pruebas y Tester, los cuales en su mayoría son Ingenieros de Sistemas, de telecomunicaciones o electrónicos.

1. PRESENTACIÓN DEL PROYECTO

1.1 DESCRIPCIÓN DEL PROYECTO

1.1.1 Título. Diseño y ejecución de pruebas de software para el sistema de información integrado: “comunidad Colciencias – fase 1” del departamento administrativo de ciencia, tecnología e innovación Colciencias.

1.1.2 Objetivo general. Realizar el diseño de casos de prueba de software para el Sistema de Información Integrado: “COMUNIDAD COLCIENCIAS – FASE 1” del Departamento Administrativo de Ciencia, Tecnología e Innovación Colciencias en el que se obtendrá información objetiva e independiente sobre la calidad del producto.

1.1.3 Objetivos Específicos

- Comprender el funcionamiento de cada una de las herramientas empleadas para el diseño y ejecución de casos de prueba.
- Aplicar la metodología de desarrollo de casos de prueba implementadas por el grupo de investigación CIDLIS.
- Efectuar el diseño de casos de prueba funcionales de cada uno de los casos de uso asignados, correspondientes a las iteraciones uno y dos del paquete dos enmarcados dentro del plan maestro de pruebas diseñado por el grupo de investigación CIDLIS.

- Realizar la ejecución de casos de prueba funcionales asignados por el líder de pruebas funcionales del grupo de investigación CIDLIS.

1.1.4 Indicadores de logros y objetivos. La siguiente tabla muestra un índice de los objetivos propuestos y los módulos desarrollados:

Tabla 1 Indicadores de logros y objetivos

Objetivo	Ítem en el documento
Diseñar estrategias y planes de pruebas para solicitudes de cambio de software de los sistemas integrados utilizados para los servicios ofrecidos por COLCIENCIAS	4.3 Estrategia de prueba
Llevar a cabo la ejecución de pruebas estáticas orientadas a detectar defectos o validar aciertos según los requerimientos documentados en las especificaciones del proyecto.	4.1 Planificación 4.4 Validación y Verificación
Reportar y hacer seguimiento de defectos o incidencias detectadas en la ejecución de las pruebas del software según estándares de calidad con el fin de dar lugar a las correcciones respectivas.	5. Defectos 6.1 Defectos encontrados
Estimar el impacto generado con el aporte del practicante, su incidencia desde su llegada al proyecto UIS-COLCIENCIAS.	3.2. Primeros resultados 6. Resultados

1.2 JUSTIFICACIÓN

1.2.1 Antecedentes y descripción del proyecto. Desde hace casi 1 año, cuando se dio puesta en marcha al proyecto del grupo de investigación CIDLIS-UIS en COLCIENCIAS, se ha consolidado un eficiente grupo de trabajo con claros objetivos enfocados al aseguramiento de la calidad en el sistema de información que se utilizará. La carga de trabajo es directamente proporcional a la cantidad de módulos y funcionalidades que la empresa cliente (Colciencias) quiera implementar y la cantidad de módulos o funcionalidades que la empresa proveedor (Tecnocom) desarrolla.

El objetivo principal que rige en el proyecto desde sus inicios es el certificar un margen de calidad aceptable por el cliente en los productos entregados por la fábrica de desarrollo, para posteriormente dar visto bueno a la implementación de los cambios a nivel productivo con un mínimo riesgo de fallos. En ningún proyecto de aseguramiento de la calidad se puede garantizar en un 100% que no se detecten errores en entornos productivos, pero gracias a un exhaustivo plan de gestión de pruebas se puede lograr un producto fiable que de un parte de tranquilidad tanto al usuario como al cliente.

El estudiante inició su vinculación en el proyecto, antes de comenzar la práctica empresarial, como auxiliar del grupo de investigación Cidlis, ejecutando las labores de cargue de diseños de casos de pruebas a la herramienta de gestión de pruebas Testlink, dado que la labor permitía la relación directa con el contenido de los documentos de diseño generados, el estudiante logró ir conociendo la metodología de pruebas aplicada en el proyecto, además del comportamiento general y la relación de los casos de uso suministrados por parte de Colciencias, dicha actividad fue realizada por el estudiante durante más de dos meses.

Dado el grado de conocimiento adquirido durante el periodo de tiempo dedicado como auxiliar del grupo y la posibilidad de integración del estudiante como practicante, el equipo de trabajo de Cidlis-Uis junto con el estudiante toman la decisión de incluirlo como practicante en el proyecto, viendo así una serie de oportunidades en las cuales se podrían tener mayores resultados con su dedicación de tiempo completo.

Una vez asumido el rol de practicante por parte del estudiante éste inició su introducción inmediatamente y fue instruido en los objetivos que tiene trazado el proyecto. Se brindó capacitación en las actividades y tareas que se realizan en cada una de las fases de testeo con el líder de pruebas funcionales. El proyecto requería de un probador técnico-funcional y las primeras responsabilidades asignadas al practicante estuvieron orientadas a adiestrarse en dicho perfil y al manejo de las herramientas requeridas para la ejecución de las actividades a desarrollar.

La práctica empresarial se desarrolló en un entorno óptimo, en donde se pudieron llevar a cabo actividades propias de un ingeniero de sistemas orientado al aseguramiento de la calidad del software. El equipo de trabajo conformado de ingenieros de sistemas profesionales con años de experiencia en el campo logró enriquecer enormemente las competencias del estudiante e influyó notablemente en dirigir sus esfuerzos a un enfoque orientado al proceso de pruebas.

A medida que el practicante cumplía con las tareas asignadas y reportaba resultados, fue adquiriendo cada vez más responsabilidades de diferentes roles.

1.2.2 Impacto. Los beneficios que ha generado integración de estudiantes en periodo de aprendizaje son inmensos y a pesar que no es fácil cuantificar los resultados, los balances favorables y la progresión que tiene el proyecto en sus reportes periódicos son suficiente justificación para continuar con esta medida. Una muestra manifiesta del éxito de CIDLIS-UIS con sus buenas prácticas es la

reciente renovación del contrato con su cliente COLCIENCIAS y con ello afianza su buena gestión y desarrollo en el marco del proyecto, además de buenas proyecciones con proyectos futuros que denotan la confianza entregada por parte de Colciencias.

Es altamente significativa la trascendencia que el proyecto realizado mediante el convenio CIDLIS-UIS-COLCIENCIAS inyecta en el perfil profesional del practicante ya que dada la experiencia del grupo de investigación y del grupo de trabajo contratado para la ejecución de las labores de pruebas se han esmerado en la divulgación del conocimiento y en labores de enseñanza para que el practicante logre contribuir altamente en el desarrollo de cada una de las actividades propuestas por la gerencia del proyecto.

En el caso específico que se está exponiendo, el estudiante al concluir su práctica desempeñaba labores de Analista de pruebas. En este rango dentro del proyecto se ubicaba transicionalmente en un punto intermedio entre ejecutor de casos de pruebas o *tester* y líder técnico-funcional. Un cargo muy importante para ser desempeñado por un estudiante, pero otorgado gracias a la dedicación, esfuerzo en la ejecución de cada una de las labores asignadas y el alto interés por el aprendizaje de nuevas actividades.

Tal fue el impacto recíproco que generó la práctica realiza por el estudiante a lo largo de los 4 meses de participación en el proyecto que a su término fue ofrecido el cargo de analista de pruebas funcionales para la continuación en la contribución al proyecto.

1.2.3 Viabilidad

- Técnica

La viabilidad de técnica del proyecto de grado en la modalidad de práctica empresarial es sólida dada la visión del proyecto que exige que sus recursos humanos se desempeñen en el área de las tecnologías de la información, la

ingeniería de software y la planeación de proyectos a nivel corporativo. Dentro del proyecto se cuenta, en su mayoría, con ingenieros de sistemas, electrónicos, de telecomunicaciones. Todo este portafolio de profesionales garantiza que trabajar dentro del proyecto sea altamente fructífero para cualquier integrante y debido a la realimentación a la que se someten los integrantes del equipo.

- Económica

El grupo de investigación CIDLIS tiene los suficientes recursos económicos para poder dar cumplimiento a la labor designada por parte del cliente (COLCIENCIAS) y poder disponer de los recursos tanto humano como tecnológico. De igual forma el grupo de investigación en su visión estratégica del proyecto tenía los recursos necesarios para incluir al practicante en la participación del proyecto y así poderle aportar el valor equivalente a un SMMLV para sus gastos.

- Social

Son diversos los niveles sociales en los que impacta positivamente con la realización de este proyecto. De primera mano el máximo beneficiario es Colciencias, en su división de tecnología, de forma específica los usuarios finales de los sistemas que se someten a las pruebas de Cidlis-UIS pues se entrega un producto de calidad con un muy bajo riesgo a fallos.

A un nivel más bajo se favorecen todos los usuarios de Colciencias que disfrutan de sus productos y servicios ofrecidos, los cuales son gestionados y mantenidos por los sistemas software que Cidlis-Uis certifica como aceptables para su paso a producción.

Se pretende a futuro ser una influencia positiva para estudiantes que deseen adoptar a la práctica empresarial como su metodología de proyecto de grado. Que este trabajo sirva como guía para todas personas que buscan asesoría en temas de aseguramiento de calidad y pruebas de software.

2. MARCO CONCEPTUAL

2.1 MARCO DE REFERENCIA

2.1.1 Breve marco de referencia del grupo de investigación CIDLIS¹. El grupo de investigación Cidlis creado en el año 1991-1, ha estado bajo la dirección del PhD. Ricardo Llamosa quien ha orientado al grupo hacia la investigación y el fortalecimiento de las áreas de conocimiento de Computación y Ciencias de la Información.

En sus objetivos tanto generales como específicos se fundamentan en principios direccionados por el sistema de gestión de calidad implementado al interior del grupo. Estos son básicamente los que se presentan a continuación: Incrementar el nivel de satisfacción de los clientes internos (investigadores, estudiantes, y demás recurso humano del CIDLIS) y externos (organizaciones industriales, comerciales y académicas, que manufacturen, comercialicen o utilicen el software como infraestructura para su realización estratégica, logística y táctica).

Los principales retos contemplados por el grupo, y dentro de su proyección se encuentran el fortalecimiento del programa de mercadeo para conseguir y divulgar proyectos externos que le den estabilidad económica al centro. De otra parte, está el potenciar el uso de estándares en la industria del software, mediante el entrenamiento, capacitación, asesoría y consultoría a empresas del sector productivo, fundamentados en el conocimiento y aplicación de estándares internacionales. Así mismo se contempla la vinculación de un mayor número tanto

¹ <http://scienti.colciencias.gov.co:8080/gruplac/jsp/visualiza/visualizagr.jsp?nro=0000000003003>

de investigadores como de estudiantes (semillero de investigadores en formación) que permitan soportar mayormente la labor de investigación en cada una de las líneas del grupo. Promover y aumentar la producción escrita de los desarrollos e investigaciones realizadas en cada una de las líneas del grupo de investigación. Desarrollar proyectos de investigación aplicada de manera que integre diferentes entidades del sector productivo, para lo cual, se especificará el Modelo Funcional que busque las mejores prácticas tanto de una empresa educativa como de una empresa del sector productivo. Además, convertirse en un grupo consultor y laboratorio de métodos de acuerdo al modelo industrial ISO12207, CMM, PSP, TSP y BSC, con reconocimiento de la Industria informática en el área de Ingeniería de sistemas, calidad y modelado organizacional.

2.1.2 Proyecto de Servicio de Pruebas de software del contrato: "la construcción del sistema de información integrado: "comunidad Colciencias – fase 1" y su puesta en servicio, mantenimiento y soporte. El proyecto de Servicios de pruebas contratado entre Colciencias y la UIS estipula una serie de compromisos en cuanto a las responsabilidades de la UIS como contratista las cuales van encaminadas al buen desarrollo del proyecto y al uso de buenas prácticas que permitan dar el manejo adecuado a cada una de las dificultades que se presenten a lo largo del desarrollo de las labores de pruebas de software, entre ellas:

- Verificar los requerimientos establecidos por cada uno de los interesados en los casos de uso para los ciclos de desarrollo determinados.
- Analizar y verificar los requerimientos de las capacidades y cualidades de las aplicaciones establecidas en los casos de uso de cada ciclo de desarrollo.

- Diseñar las pruebas considerando la matriz de trazabilidad definida, conforme al plan y monitorear, analizar e informar los resultados de la ejecución de las pruebas de acuerdo al diseño de cada una de las aplicaciones.
- Realizar las pruebas y emitir conceptos de aceptación o rechazo de acuerdo a los resultados de las mismas
- Gestionar incidencias presentadas en cada ciclo de pruebas.
- Reportar indicadores de calidad del software por cada ciclo de pruebas.

Entre otras.

Con el fin de dar cumplimiento a las responsabilidades adquiridas por del equipo de trabajo de Cidlis-Uis se desarrolló un plan maestro de pruebas, el cual tiene como objetivo abarcar la totalidad de casos de uso desarrollados por parte del Departamento Administrativo de Ciencia, Tecnología e Innovación Colciencias, dando como resultado la división de cada uno de estos grupos de casos de uso por paquetes e iteraciones, vale aclarar que un paquete puede estar conformado por una o más iteraciones. A continuación, se hace referencia a lo anteriormente descrito mediante una tabla, la cual permite visualizar la cantidad de paquetes e iteraciones obtenidas luego del análisis y diseño del plan maestro de pruebas, con el cual se dará solución a las necesidades expuestas por parte del Departamento Administrativo de Ciencia, Tecnología e Innovación Colciencias.

Tabla 2 Distribución de casos de uso

Pruebas Funcionales		
Paquetes	Iteraciones	Número de casos de uso
Paquete 1	Iteración 1	39
	Iteración 2	33
	Iteración 3	24
Paquete 2	Iteración 1	20
	Iteración 2	37
	Iteración 3	22
Paquete 3	Iteración 1	33
	Iteración 2	43
	Iteración 3	13
Paquete 4	Iteración 1	77
	Iteración 2	14
Total		355

2.2 MARCO TEÓRICO

2.2.1 Pruebas de software. Las pruebas de software intentan demostrar que un programa o software hace lo que se intenta que haga, así como descubrir defectos en el programa antes de usarlo. Al probar el software, se ejecuta un programa con datos artificiales. Hay que verificar los resultados de la prueba que se opera para

buscar errores, anomalías o información de atributos tanto funcionales como no funcionales del programa.

El proceso de pruebas tiene dos metas distintas:

- Demostrar al desarrollador y al cliente que el software cumple con los requerimientos. Para el software personalizado, esto significa que en el documento de requerimientos debe haber, por lo menos, una prueba por cada requerimiento. Para los productos de software genérico, esto quiere decir que tiene que haber pruebas para todas las características del sistema, junto con combinaciones de dichas características que se incorporarán en la liberación del producto.
- Encontrar situaciones donde el comportamiento del software sea incorrecto, indeseable o no esté de acuerdo con su especificación. Tales situaciones son consecuencia de defectos del software. La prueba de defectos tiene la finalidad de erradicar el comportamiento indeseable del sistema, como caídas del sistema, interacciones indeseadas con otros sistemas, cálculos incorrectos y corrupción de datos

Con la realización de las pruebas no se puede garantizar que el software esté libre de errores, así los planes de pruebas y las herramientas que se utilicen para su detección sean muy sofisticadas no se puede garantizar que el software esté 100% libre de fallos. Siempre es probable que una prueba que se haya pasado por alto pueda descubrir problemas adicionales en el sistema.

Generalmente el objetivo de la realización de pruebas a un software es darle un grado de confiabilidad tanto al desarrollador como al cliente de que dicho software es lo suficientemente bueno o aceptable para su uso puesta en producción. El testeo es un proceso que intenta proporcionar confianza en el software.

En ocasiones, algunos errores generados o causados por las personas al momento de desarrollar pueden desencadenar un defecto en el código del programa o en un documento, sí dicho defecto es ejecutado en el código, éste puede no hacer lo que debería hacer o simplemente puede no afectar el resultado, lo cual provocaría un fallo. Vale aclarar que algunos defectos de software, sistemas o documentos pueden dar lugar a fallos, pero no todos los defectos lo hacen, lo cual dificulta la labor de su detección.

El hecho de someter los sistemas y la documentación a pruebas rigurosas puede ayudar a reducir el riesgo de complicaciones durante las operaciones y contribuir a la calidad del sistema de software, siempre que los defectos detectados se corrijan antes que el sistema se ponga a disposición de sus usuarios finales.

2.2.2 Principios de las pruebas. En los últimos 40 años se ha propuesto una serie de principios que establecen pautas generales comunes a todas las pruebas:

- Principio 1 – Las pruebas muestran la presencia de defectos

Probar reduce la probabilidad de que existan bugs, pero nunca se puede asegurar que no quede ninguno oculto. Además, cada tipo de pruebas que se realice (de sistema, de integración, añadir auditorias de código, entre otras) son más efectivas para detectar un tipo de error.

El objetivo principal de llevar a cabo una prueba es para detectar defectos. Trabajando bajo la premisa de que cada producto contiene defectos de algún tipo, una prueba que revela los errores es generalmente mejor que una que no lo hace.

Todas las pruebas deben ser diseñadas para revelar tantos errores como sea posible.

- Principio 2 – Las pruebas exhaustivas son imposibles

Las pruebas exhaustivas tratan de cubrir todas las combinaciones posibles de datos en el software, a fin de garantizar que ninguna situación puede surgir, una vez probado el software se ha liberado. Excepto en aplicaciones muy simples, el número de combinaciones posibles de datos es demasiado alta, es más eficaz y eficiente que los ingenieros de pruebas se centren en las funcionalidades de acuerdo a riesgos y prioridades.

- Principio 3 – Pruebas tempranas

Un producto (incluyendo los documentos, tales como la especificación del producto) se puede probar tan pronto como se ha creado. ISTQB recomienda probar un producto tan pronto como sea posible, corregir los errores lo más rápido posibles. Los estudios han demostrado que los errores identificados al final del proceso de desarrollo por lo general cuestan más para resolver.

Por ejemplo: un error encontrado en las especificaciones puede ser bastante sencillo de solucionar. Sin embargo, si ese error se transfiere a la codificación de software, una vez descubierto el error puede ser muy costoso y consume tiempo.

- Principio 4 – Agrupación de defectos

Los estudios sugieren que los problemas en un elemento de software tienden a agruparse en torno a un conjunto limitado de módulos o áreas. Una vez que estas áreas han sido identificadas, los administradores eficientes de prueba son capaces de enfocar las pruebas en las zonas sensibles, mientras que siguen buscando a los errores en los módulos de software restantes.

- Principio 5 – Paradoja del pesticida

Al igual que el sobre uso de un pesticida, un conjunto de pruebas que se utilizan repetidamente en el disminuirá en su eficacia. Usando una variedad de pruebas y técnicas expondrá una serie de defectos a través de las diferentes áreas del producto

- Principio 6 – La prueba es dependiente del contexto

Las mismas pruebas no se deben aplicar en todos los ámbitos. Distintos productos de software tienen diferentes requisitos, funciones y propósitos. Una prueba diseñada para realizarse en un sitio web, por ejemplo, puede ser menos eficaz cuando se aplica a una aplicación de intranet. Una prueba diseñada para una forma de pago con tarjeta de crédito puede ser innecesariamente rigurosa si se realiza en un foro de discusión.

En general, cuanto mayor es la probabilidad y el impacto de los daños causados por el software fallado, mayor es la inversión en la realización de pruebas de software.

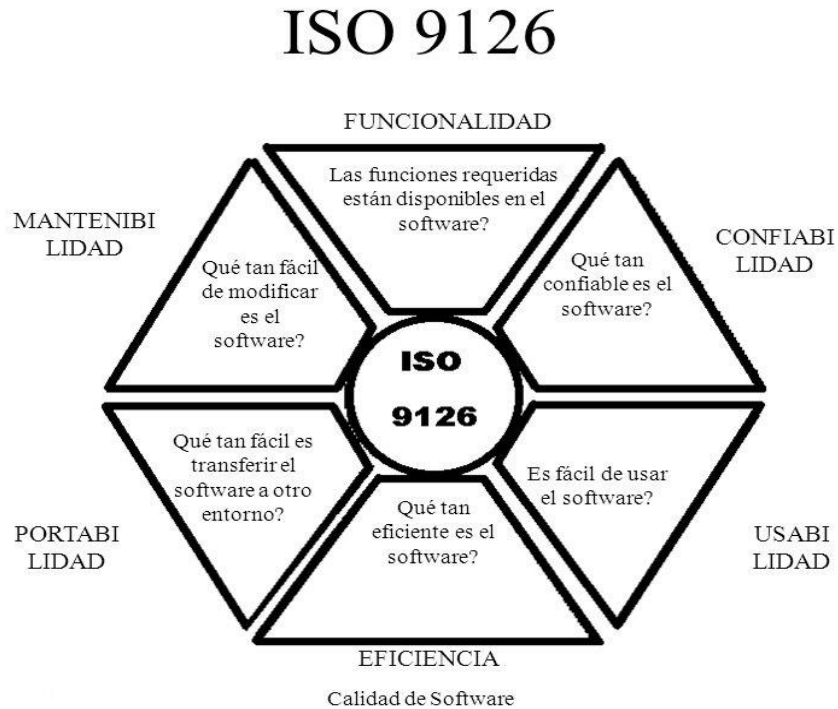
- Principio 7 – Falacia de ausencia de errores

Declarar que una prueba no ha descubierto ningún error no es lo mismo que declarar que el software es “libre de errores”. Con el fin de garantizar que los procedimientos adecuados de software de prueba se llevan a cabo en todas las situaciones, los evaluadores deben asumir que todo el software contiene algunos (aunque disimulada) errores.

2.2.3 ISO 9126 El estándar ISO 9126. Fue originalmente desarrollado en 1991 para proporcionar un esquema para la evaluación de calidad del software. El modelo establece seis atributos clave de la calidad:

- Funcionalidad: Grado en el que el software satisface las necesidades planteadas según las establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.
- Confiabilidad: Cantidad de tiempo que el software se encuentra disponible para su uso, según lo indican los siguientes atributos: madurez, tolerancia a fallas y recuperación.
- Usabilidad: Grado en el que el software es fácil de usar, según lo indican los siguientes sub-atributos: entendible, fácil de aprender y operable.
- Eficiencia: Grado en el que el software emplea óptimamente los recursos del sistema, según lo indican los sub-atributos siguientes: comportamiento del tiempo y de los recursos.
- Portabilidad: Facilidad con la que el software puede llevarse de un ambiente a otro según lo indican los siguientes atributos: adaptable, instalable, conformidad y sustituible
- Mantenibilidad: Facilidad con la que pueden efectuarse reparaciones al software, según lo indican los atributos que siguen: analizable, cambiable, estable, susceptible de someterse a pruebas.

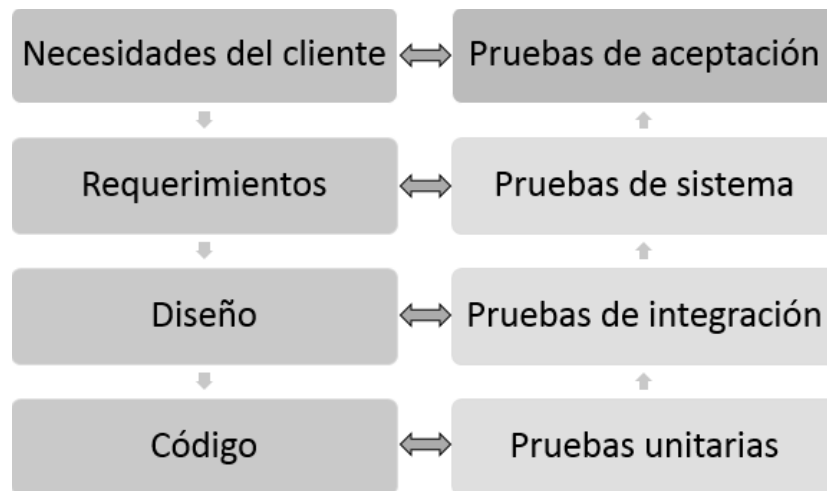
Figura 1 ciclo en la estrategia de pruebas



2.2.4 Niveles de pruebas. Por cada nivel de prueba, pueden identificarse los siguientes aspectos: los objetivos genéricos, los productos de trabajo a que se hace referencia para derivar los casos de prueba (base de pruebas), el objeto de la prueba (lo que se está probando), los defectos y fallos típicos a detectar, los requisitos de arnés de pruebas y soporte de herramientas, y los enfoques específicos y responsabilidades.

En la fase de planificación de pruebas deberá tenerse en cuenta los datos de configuración del sistema, si dichos datos forman parte del sistema.

Figura 2 Niveles de pruebas



2.2.4.1 Pruebas de componente. La base de las pruebas son los requisitos de componentes, el diseño de detalle y el código. Los objetos de prueba típicos son los componentes, los programas y la conversión de datos o programas de migración.

Las pruebas de componente (pruebas de unidad, módulo o programa) tienen por objetivo localizar defectos en y comprobar el funcionamiento de módulos de software, programas, objetos, clases, etcétera., que pueden probarse por separado. Pueden realizarse de manera independiente del resto del sistema, en función del contexto del ciclo de vida de desarrollo y del sistema. Para ellos pueden utilizarse “stubs”², controladores y simuladores.

Las pruebas de componente pueden incluir pruebas de funcionalidad y características no funcionales específicas, tales como el comportamiento de

² <http://xunitpatterns.com/Test%20Stub.html>

recursos o pruebas de robustez, además de pruebas estructurales. Los casos de prueba se derivan de productos de trabajo, tales como la especificación del componente, el diseño del software o el modelo de datos.

En general, las pruebas de componente se llevan a cabo de mediante el acceso al código objeto de las pruebas y con el soporte de un entorno de desarrollo. En la práctica, las pruebas de componente generalmente cuentan con la participación del programador que escribió el código.

2.2.4.2 Pruebas de integración. El objetivo de las pruebas de integración es comprobar la interacción entre elementos software (componentes) tras la integración del sistema.

La integración es la actividad en la cual se combinan componentes software individuales en subsistemas más amplios. La integración adicional de subsistemas también es parte del proceso de integración del sistema. Cada componente ya ha sido probado en lo referente a su funcionalidad interna (prueba de componente). Las pruebas de integración comprueban las funciones externas.

2.2.4.3 Pruebas de sistema . La base de las pruebas son la especificación de requisitos del sistema y software, las pruebas del sistema software integrado se realizan con el objeto de comprobar la conformidad con los requisitos especificados. La calidad del software es observada desde el punto de vista del usuario.

Las pruebas de sistema se refieren a (según ISO 91260)

- Requisitos Funcionales
 - Funcionalidad
- Requisitos no Funcionales
 - Fiabilidad
 - Usabilidad

- Eficiencia
 - Portabilidad
 - Mantenibilidad
- ✓ Las pruebas de sistema se desarrollan utilizando casos de prueba funcionales y no funcionales.
 - ✓ Las pruebas de sistema funcionales confirman que los requisitos para un uso específico previsto han sido satisfechos (validación)
 - ✓ Las pruebas de sistema no funcionales verifican los atributos de calidad no funcionales, por ejemplo, usabilidad, eficiencia, portabilidad, etc.
 - ✓ Con frecuencia, los atributos de calidad no funcionales son una parte implícita de los requisitos, esto hace difícil validarlos.

2.2.4.4 Pruebas de aceptación. Las pruebas de aceptación son pruebas formales llevadas a cabo con el objeto de verificar la conformidad del sistema con los requisitos. El objeto es aportar justificación a la confianza en el sistema para que pueda ser aceptado por el cliente.

Es habitual que sean las primeras pruebas en las cuales se vea involucrado al cliente (es recomendable involucrar al cliente durante el proceso de desarrollo, por ejemplo, dar soporte al desarrollo de prototipos)

El grado de implicación de los clientes puede variar en función del tipo de programa (software individual o un producto software comercial)

En conclusión, las pruebas de aceptación son las pruebas de sistema por parte del cliente. Además, estas pruebas son una actividad de carácter contractual, se verifica entonces que el software satisface los requisitos del cliente.

Las pruebas alpha y beta son pruebas ejecutadas por clientes reales o potenciales; en las dependencias del desarrollador (alpha) o en las dependencias del cliente (beta).

2.2.5 Tipos de pruebas. Las actividades de pruebas tienen por objetivo comprobar el sistema de software o alguna parte de él con base a un motivo u objeto específico.

Un tipo de prueba se centra en un objetivo de prueba en particular, que pueden ser:

- Una función a realizar por el software
- Una característica de calidad no funcional, tales como la fiabilidad o usabilidad
- La estructura o arquitectura del software o sistema

Puede desarrollarse y/o utilizarse un modelo de software en las pruebas estructurales, en las pruebas no funcionales, y en las pruebas funcionales.

2.2.5.1 Pruebas funcionales. Se denominan pruebas funcionales o Functional Testing, a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados, es común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales, esta etapa suele ser la última etapa de pruebas y al dar conformidad sobre esta el paso siguiente es el pase a producción.

Tabla 3 Modelo de presentación de un caso de uso en un documento de especificación

Código Caso de Uso	Nombre Caso de Uso
Descripción	
Actores	
Precondiciones	
Diagrama del caso de uso	
Secuencia básica de eventos	
Secuencia alternativa	
Secuencia de Excepciones	
Reglas de Negocio	
Notas y comentarios	

Las pruebas funcionales se basan en funciones y prestaciones y en su interoperabilidad con sistemas específicos, y pueden llevarse a cabo en todos los niveles de prueba.

Las técnicas basadas en la especificación sirven para obtener condiciones de pruebas y casos de prueba a partir de la funcionalidad de un software o sistema. Se consideran Pruebas de Caja Negra (“black-box testing”) puesto que valoramos el comportamiento externo del sistema. Las Pruebas de Seguridad o las Pruebas

de Interoperabilidad entre sistemas o componentes son casos especializados de las pruebas funcionales.

Figura 3 Modelo de pruebas de Caja Negra



2.2.5.2 Pruebas no funcionales. Las pruebas no funcionales incluyen las pruebas de: Rendimiento, Carga, Estrés, Usabilidad, Mantenibilidad, Fiabilidad o Portabilidad, entre otras. Por tanto, se centran en características del software que establecen “cómo trabaja el sistema”.

Estas pruebas también pueden ejecutarse en todos los niveles de pruebas. Las características no funcionales del software se pueden medir de diversas maneras, por ejemplo, por medio de tiempos de respuesta en el caso de pruebas de rendimiento o por número máximo de sesiones en pruebas de estrés.

Puesto que las Pruebas no Funcionales normalmente consideran el comportamiento externo del sistema, en la mayoría de los casos se utilizan técnicas de Pruebas de Caja Negra.

2.2.5.3 Pruebas estructurales. Las pruebas estructurales o pruebas de caja blanca pueden realizarse en todos los niveles de prueba. Las técnicas estructurales son las más idóneas, después de las técnicas basadas en la especificación, para ayudar a medir la exhaustiva de las pruebas mediante una evaluación de la cobertura de un tipo de estructura.

La cobertura es la medida en que un juego de pruebas ha probado una estructura, expresada como porcentaje de los elementos cubiertos. Si la cobertura no es del 100%, entonces podrán diseñarse más pruebas para probar los elementos que faltan para aumentar la cobertura.

En todos los niveles de prueba, pero especialmente en las pruebas de componente y las pruebas de integración de componentes, puede recurrirse a herramientas para medir la cobertura de código de los elementos, tales como sentencias o decisiones. Las pruebas estructurales pueden basarse en la arquitectura del sistema.

Los enfoques de las pruebas estructurales también pueden aplicarse a nivel de sistema, integración de sistemas o pruebas de aceptación.

2.2.5.4 Pruebas de regresión. Una vez detectado y corregido un defecto, el software debe volver a probarse para confirmar que el defecto original ha sido corregido con éxito. A esto se le denomina confirmación. La depuración o corrección de defectos es una actividad de desarrollo, no una actividad de pruebas.

Las pruebas de regresión son la prueba reiterada de un programa ya probado, después de haber sido modificado, con el fin a localizar defectos surgidos o no descubiertos como resultado del cambio o de los cambios. Estos defectos pueden estar en el software objeto de las pruebas, o en cualquier otro componente de software asociado o no asociado. Se realizan cuando el software, o su entorno,

sufren modificaciones. El alcance de las pruebas de regresión depende del riesgo de no encontrar defectos en el software que antes funcionada.

Las pruebas deben ser repetibles si desean utilizarse a efectos de las pruebas de confirmación o para dar soporte a las pruebas de regresión.

Las pruebas de regresión pueden realizarse en todos los niveles de prueba, e incluyen pruebas funcionales, no funcionales y estructurales. Los juegos de pruebas de regresión se ejecutan muchas veces y por lo general son de lenta evolución, por lo que las pruebas de regresión constituyen un gran potencial para la automatización.

3. PLAN DE TRABAJO

3.1 CARGO INICIAL Y RESPONSABILIDADES

Al inicio de la práctica, el estudiante llega al cargo de auxiliar de pruebas, cargo en el que se asignan las primeras obligaciones o tareas a ejecutar entre las cuales se destacan en las primeras semanas: el entrenamiento de las herramientas para la gestión de las pruebas y para el reporte de incidencias (TestLink y Mantis) además de recibir capacitación en cuanto a la forma en la que se había diseñado el plan maestro de pruebas y el significado que tenían cada uno de los nombres que se le daban a los casos de uso que se iban a manejar a lo largo de la práctica.

Culminadas las etapas de entrenamiento al practicante se le asignan los primeros casos de prueba para su ejecución y su correspondiente detección de errores, una vez realizadas las labores de ejecución de casos de prueba se procede a realizar el respectivo tratamiento a cada una de las incidencias reportadas, cabe resaltar que el manejo de las incidencias se realizó en la herramienta Mantis, mediante la cual se le asigna el nivel de severidad de la incidencia, prioridad con la que se debe tratar, nivel de reproducibilidad, entre otros campos que me permiten dar un nivel de detalle alto con el fin de lograr identificar de forma adecuada la incidencia.

Posteriormente dado el buen manejo de las herramientas y la facilidad con la que se comprendía el funcionamiento del sistema de información a probar, la persona líder de pruebas funcionales decide asignar casos de prueba con un nivel de complejidad más alto obteniendo en esta labor muy buenos resultados.

Luego de que el practicante lograra tener un grado de familiaridad alto con cada uno de los casos de uso y con el manejo del sistema de información, se le asigna la tarea de comenzar con el diseño de casos de prueba con los cuales se efectúa

la ejecución de las pruebas, labor para la cual el principal insumo son los casos de uso y teniendo en cuenta su estructura básica, tales como: Secuencias básica, reglas de negocio, secuencias alternativas, secuencias de excepciones, tablas de detalles de campos, se logran crear diseños óptimos para su ejecución, recibiendo el aval de los analistas de pruebas quienes supervisaban dicha actividad.

3.2 PRIMEROS RESULTADOS

La efectividad de un probador o tester se mide por los defectos que detecta en sus casos de prueba ejecutados, por la severidad y prioridad de los defectos y además por el porcentaje de defectos que son solucionados por el proveedor de desarrollo.

El cargo de tester fue en sus inicios la prioridad del estudiante, siempre realizando sus actividades bajo la supervisión de personal capacitado para esta labor y con experiencia.

Un buen probador debe demostrar una sana desconfianza en el desarrollo y debe estar muy atento a los detalles. Estas facultades están muy presentes en el practicante y por ello lo llevaron a adaptarse rápidamente a la carga laboral y a las metas propuestas por la persona líder de pruebas funcionales.

De manera intuitiva y por iniciativa tanto propia como de las personas que estaban a cargo de la supervisión de las actividades realizadas por el practicante, éste posteriormente asume el cargo de Analista de pruebas, cargo en el que no sólo se debe ejecutar una serie de pasos de una suite de casos de prueba, sino que opta una posición más crítica y analítica respecto al funcionamiento del sistema de información a probar. El practicante contribuye con el diseño de casos de prueba que permitan ejecutar una secuencia de pasos más entendible y siempre encaminados al objetivo principal del caso de uso sobre el cual se esté trabajando, además genera observaciones y conclusiones de hallazgos encontrados a lo largo

de la labor de diseño, información que es de vital importancia para la generación de informes de resultados entregados a la empresa contratante que para este caso es COLCIENCIAS. Todo lo anterior ayuda al proyecto a dar una certificación precisa y a garantizar un producto de salida con altísima calidad. Dado que el practicante aún no tiene la experiencia ni la confianza del proyecto suficientes para hacerse totalmente con los compromisos de las tareas, siempre se le es asignado un analista de pruebas que certifique que las actividades realizadas por el practicante han sido realizadas de manera correcta, de esta forma se impone un grado alto de confiabilidad a las labores de pruebas por parte del equipo de trabajo del grupo de investigación CIDLIS-UIS.

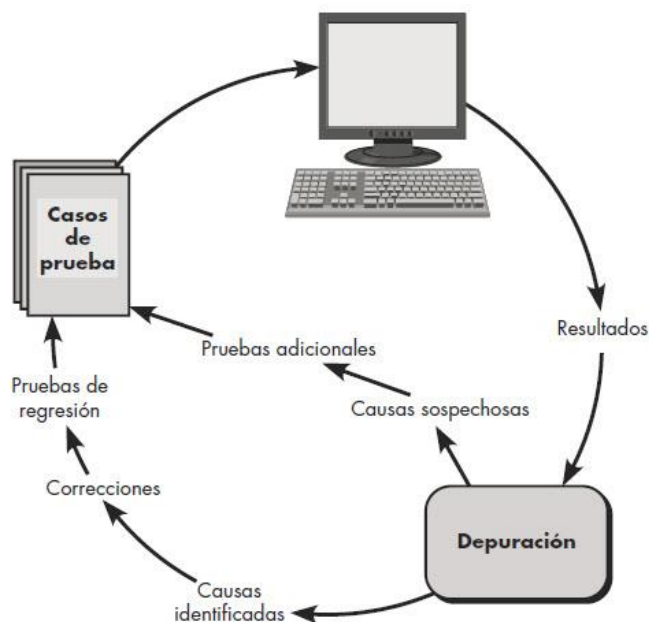
4. CICLO DE VIDA DE PRUEBAS

4.1 PLANIFICACIÓN

La planificación de pruebas es la actividad de definir los objetivos de las pruebas y la especificación de las actividades de pruebas con el fin de cumplir los objetivos y la misión establecidos.

La planificación de pruebas se realiza alrededor del desarrollo y la implementación de proyectos, así como en las actividades de mantenimiento de aplicaciones. La planificación se documenta en un plan maestro de pruebas y en planes de prueba por separado para niveles de pruebas tales como pruebas de sistema y pruebas de aceptación.

Figura 4 Proceso de Depuración



4.2 DESARROLLO

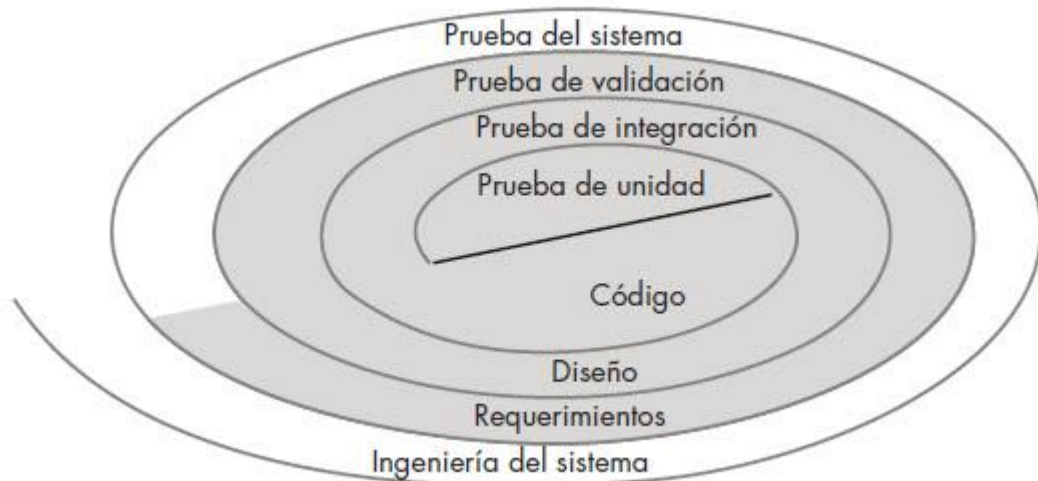
Como ya se ha indicado con anterioridad, el equipo de trabajo del grupo de investigación Cidlis-Uis es responsable exclusivamente de la gestión y la ejecución de casos y sets de pruebas. El área funcionales y de especificaciones técnicas es división de tecnología de Colciencias. Pero en cuanto a la labor de desarrollo se deben referir varios proveedores privados externos o uniones temporales entre alguno de ellos, para el caso del proyecto sólo se cuenta con una empresa desarrolladora (tecnocom). Es claro que el proceso de pruebas no es un proceso aislado; las actividades de pruebas están asociadas a actividades de desarrollo de software. El desarrollo de distintos modelos de ciclo de vida requiere distintos enfoques de pruebas.

4.3 ESTRATEGIA DE PRUEBA

El enfoque de pruebas es la aplicación de la estrategia de pruebas para un proyecto específico. El enfoque de pruebas se define y redefine en los planes y diseños de prueba. Generalmente, incluye las decisiones tomadas con base al objetivo y a la evaluación de los riesgos del proyecto de prueba. Constituye el punto de inicio para planificar el proceso de pruebas, para seleccionar las técnicas de diseño de pruebas y los otros tipos de pruebas a aplicar, y para definir los criterios de entrada y salida.

El enfoque seleccionado dependerá del contexto y puede tener en cuenta los riesgos, los peligros y la seguridad, los recursos disponibles y habilidades, la tecnología, la naturaleza del sistema (por ejemplo, sistema personalizada versus COTS), los objetivos de prueba y la normativa.

Figura 5 Ciclo de pruebas. Ingeniería de software, Un enfoque práctico 7ma edición- Roger S. Pressman



Para efectos de la práctica y del desarrollo del proyecto de Servicios de pruebas de software del contrato: "la construcción del sistema de información integrado: "comunidad Colciencias – fase 1" y su puesta en servicio, mantenimiento y soporte, las técnicas adoptadas para el diseño de pruebas fueron las técnicas basadas en la especificación o técnicas de caja negra, las cuales poseen entre sus objetivos principales:

- Escribir casos de prueba a partir de modelos de software dados aplicando partición de equivalencia, análisis de valores límites, tablas de decisión y diagramas/tablas de transición de estado
- Explicar el objetivo principal de cada una de las cuatro técnicas de pruebas, así como qué nivel y que tipo de pruebas podría utilizar y cómo puede medirse la cobertura
- Explicar el concepto de las pruebas de caso de uso y sus ventajas

Como insumo fundamental para el desarrollo de las labores de pruebas, Colciencias junto con la empresa desarrolladora del sistema de información diseñaron la totalidad de casos de uso necesarios o involucrados en la elaboración de dicho software, para ello fue necesario más de dos años para que el contenido de cada uno de los casos de uso fueran lo más completos posibles y se ajustaran a las necesidades estipuladas por parte de Colciencias.

Todo este material elaborado fue entregado al equipo de trabajo del grupo de investigación Cidlis-Uis, para lo cual, una vez entregado este material se procedió a realizar la planeación y el diseño del plan maestro de pruebas con el cual se abordaría la labor designada.

Posteriormente se procede con la elaboración de los diseños de casos de prueba, labor en la cual el estudiante durante alrededor de dos meses de la práctica apoyó, bajo la supervisión de uno de los analistas de pruebas funcionales y contribuyó altamente con el avance del desarrollo de esta actividad.

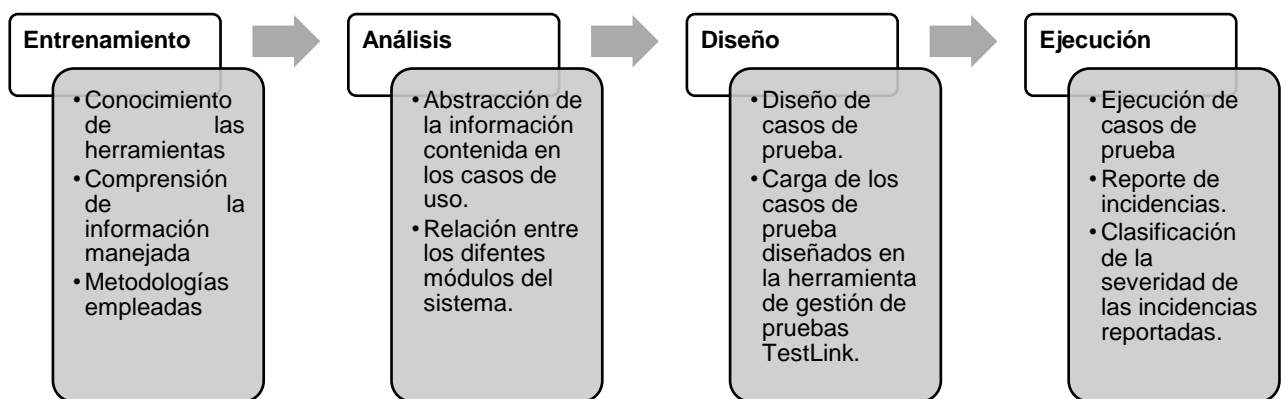
Una vez se tienen los diseños de casos de prueba ajustados y totalmente terminados y junto con la puesta en el ambiente de pruebas del respectivo entregable, se inicia la ejecución de cada uno de los casos de prueba, a la par con esta actividad es responsabilidad del probador reportar cada uno de los hallazgos encontrados, tanto incidencias como observaciones, en esta labor el estudiante contribuyó con su realización obteniendo excelentes resultados, los cuales en capítulos posteriores serán expuestos en detalle.

4.4 METODOLOGÍA

Con base a la metodología de desarrollo de pruebas de software expuesta por el Centro de Innovación y Desarrollo para la Investigación en Ingeniería del Software, CIDLIS-UIS, se creó una estrategia con la cual se abarcaron los objetivos establecidos en el proyecto con modalidad de práctica empresarial, la cual logró cubrir las diferentes etapas que el practicante tuvo que afrontar para poder afianzar sus conocimientos y lograr las metas establecidas por el líder de pruebas funcionales del equipo de trabajo.

En el siguiente gráfico se expone la metodología con la cual se enmarcó el proyecto y que se adaptó satisfactoriamente a la metodología usada por el grupo de investigación CIDLIS.

Figura 6 Metodología de desarrollo de la práctica



4.5 VALIDACIÓN Y VERIFICACIÓN

Las pruebas se consideran parte de un proceso más amplio de verificación y validación del software. Aunque ambas no son lo mismo, se confunden con frecuencia. Barry Boehm, pionero de la ingeniería de software, expresó de manera breve la diferencia entre las dos:

- “Validación: ¿construimos el producto correcto?”.
- “Verificación: ¿construimos bien el producto?”

Los procesos de verificación y validación buscan comprobar que el software por desarrollar cumpla con sus especificaciones, y brinde la funcionalidad deseada por las personas que pagan por el software. Dichos procesos de comprobación comienzan tan pronto como están disponibles los requerimientos y continúan a través de todas las etapas del proceso de desarrollo.

El objetivo final de los procesos de verificación y validación es establecer confianza de que el sistema de software es “adecuado”. Esto significa que el sistema tiene que ser lo bastante eficaz para su uso esperado. El nivel de confianza adquirido depende tanto del propósito del sistema y las expectativas de los usuarios del sistema, como del entorno del mercado actual para el sistema.

El equipo de revisiones del proyecto lleva a cabo las actividades primarias en el aseguramiento de la calidad del software en sus más tempranas etapas, estas tareas incluyen las revisiones técnicas, revisión de base de datos, revisión de controles de cambio y revisión de documentación.

Cabe resaltar que el proceso que lleva a cabo el proyecto con su personal de revisiones, representa un beneficio para las pruebas y las necesidades del cliente. Cuando se detecta una falta en una de las reglas de validación y verificación, el equipo encargado del desarrollo tiene la oportunidad de corregir y aprender, lo cual les permite ser más eficientes en sus próximas entregas.

El practicante no tuvo funciones ni participación directa dentro del grupo de revisiones, pero de igual manera ha sido su deber estar atento del estado de las solicitudes en todas sus etapas de entrega.

4.6 EJECUCIÓN DE CASOS DE PRUEBA

La ejecución de los casos de pruebas fue una de las labores desempeñadas por el practicante. Su función durante la ejecución de casos de pruebas era efectuar el paso a paso diseñado en las estimaciones de casos de pruebas.

Independiente de si era autor o no de las estimaciones y estrategias de pruebas, el practicante debía interpretar correctamente los objetivos de las pruebas y entender completamente los resultados que se esperaban al completar los casos.

De igual forma debía estar capacitado para adoptar cualquier caso de prueba. La herramienta empleada para la gestión de casos de prueba permite el acceso de todo el historial en el ciclo de vida de las pruebas, incluyendo el plan de pruebas con el conjunto de casos de pruebas, las ejecuciones que se van generando a lo largo de los ciclos de pruebas y, por supuesto, los defectos que se detectan en cada etapa.

No es admisible ejecutar un caso de prueba sin un soporte visual que lo demuestre, es por ello que se hace imprescindible el uso de una herramienta de captura de evidencias para que se integre con el diseño del paso a paso y contar con un sustento verídico de la calidad de la ejecución, para el caso específico de la práctica se usó un complemento web gratuito del navegador Google Chrome llamado screencastify. De igual manera se debe registrar cada ejecución en la

herramienta de seguimiento, sin importar la naturaleza o tipo de prueba que se lleve a cabo. Violar con estos deberes tiene consecuencias de alta severidad en materia de reportes finales de pruebas.

Al momento de realizar la ejecución de un caso de prueba se debe tener en cuenta los posibles estados que éste puede tomar:

- Pasado: El caso de prueba se comporta según las condiciones descritas dentro de la secuencia de pasos diseñada, no presenta ningún tipo de error y cumple con el objetivo establecido.
- Fallado: No se puede ejecutar de forma correcta la funcionalidad que se está probando, permite seguir el flujo normal de las descripciones establecidas en el caso de prueba, pero no cumple correctamente con lo descrito.
- Bloqueado: No permite continuar con la secuencia descrita en el caso de prueba e impide que se valide la funcionalidad a probar.

Una vez es ejecutado el caso de prueba y éste ha adoptado un estado, se debe tener en cuenta y realizar el respectivo seguimiento a las posibles incidencias asociadas, cabe resaltar que una incidencia debe crearse de forma obligatoria si el estado del caso de prueba ejecutado es fallado o bloqueado, pero cabe resaltar que es posible relacionar incidencias a casos de prueba pasados con el fin de proponer mejoras que no afectan el funcionamiento de la aplicación y que son realizadas por parte de la empresa cliente (Colciencias).

Una incidencia relacionada a un caso de prueba ejecutado puede adoptar uno de los siguientes estados:

- Abierto: Aún no se ha realizado el seguimiento al bug asociado.
- Resuelto: Se le ha dado el correcto tratamiento al bug encontrado y éste ya no tiene validez.
- Cerrado: Se ha acordado mediante mesas de concertación que el bug asociado no debe ser tenido en cuenta por diversas razones o pudo tratarse de un error en la ejecución por parte del tester y el bug reportado no tiene validez.

4.7 ANÁLISIS Y DISEÑO DE CASOS DE PRUEBA

El análisis y el diseño de pruebas es la actividad durante la cual los objetivos de las pruebas generales se transforman en condiciones de prueba y casos de prueba tangibles.

La actividad de análisis y diseño de pruebas consta de las siguientes tareas principales:

- Diseñar y priorizar los casos de prueba de alto nivel
- Diseñar la configuración del entorno de pruebas e identificar cualquier infraestructura y herramientas necesarias
- Crear una trazabilidad bidireccional entre la base de pruebas y los casos de pruebas
- Identificar y priorizar las condiciones de prueba en base al análisis de los elementos de prueba, la especificación, el comportamiento y la estructura del software

- Evaluar la testabilidad (facilidad y la rapidez con las cuales se pueden probar las características de un sistema.) de la base de prueba y los objetos de prueba

En cuanto a la labor de análisis y diseño de casos de prueba realizada por parte del practicante, el insumo principal para su elaboración fueron los casos de uso, de igual forma para el diseño de algunos casos de prueba más complejos se hacía necesario el uso de otros documentos guía tales como: términos de referencia (TDR) y anexos, estos permitían tener una comprensión más exacta en cuanto a las necesidades del cliente (Colciencias) y a lo que se debía contemplar para realizar las pruebas.

El diseño de casos de prueba es una actividad que el estudiante desarrolló durante más de un mes en su práctica, el esquema empleado para el diseño de los mismos fue el planteado por los analistas de pruebas y el líder de pruebas funcionales quienes a su vez apoyados en estándares orientaron al practicante en cuanto al contenido que debían tener cada uno de los diseños. A continuación, se presenta el esquema empleado para la creación de los documentos de casos de prueba.

Tabla 4 Esquema del documento empleado para el diseño de casos de prueba

Id		Nombre del caso de prueba	
Descripción			
Actores			
Precondiciones			
Número de paso	Descripción del paso	Resultado Esperado	Tipo de ejecución

Una vez realizado el diseño de los casos de prueba (en documento word), estos debían ser ingresados a la herramienta de gestión de casos de prueba Testlink para que pudieran quedar a disposición del líder de pruebas funcionales quien era el encargado de realizar la organización de estos en planes de prueba y pudieran ser asignados a los probadores correspondientes.

5. DEFECTOS

Teniendo en cuenta que uno de los principales objetivos de las pruebas es la detección de errores o defectos, las discrepancias entre los resultados reales y los resultados esperados deben registrarse como incidencias. Todas las incidencias deben ser investigadas, dado que algunas pueden construir defectos. Las incidencias y los defectos se rastrean desde su identificación y clasificación hasta la corrección y confirmación de su solución. Con el fin de gestionar todas las incidencias hasta su compleción, el equipo de trabajo de Cidlis-Uis estableció un proceso de gestión de incidencias y normas para su clasificación.

Las incidencias pueden surgir durante el desarrollo, la revisión, las pruebas o el uso de un producto de software. Asimismo, pueden surgir por problemas en el código o en el sistema de funcionamiento, o en cualquier tipo de documentación que incluya requisitos, documentos de desarrollo, documentos de prueba e información de usuario, como guías de ayuda o instalación.

5.1 ESTADOS DE LOS DEFECTOS

El equipo de trabajo de Cidlis-Uis estableció una serie de parámetros con el fin de estandarizar la selección del nivel de severidad de las incidencias a encontrar.

La severidad es una parte muy importante al momento de crear una nueva incidencia, pues de la severidad depende el tiempo de asignación, la persona a la

cual se asigna, el esfuerzo invertido y la urgencia con la cual debe ser resuelto el fallo. Para ello se elaboró la siguiente guía.

Tabla 5 Clasificación de la severidad de las incidencias

Tipo de error	Descripción	Ejemplo
Crítico	Falla en el sistema. No es posible continuar con la ejecución.	Un error crítico ha sido encontrado y no permite que se continúe con la operación de la aplicación. Fuga de datos, menús inoperantes.
Mayor	No es posible continuar con el proceso de la función seleccionada.	El componente clave no está disponible o la funcionalidad está incorrecta. Dificultades que inhiben parcial o totalmente el uso del programa.
Media	Funciones restringidas, pero la ejecución puede continuar.	Una parte menor del componente no es funcional, pero el aplicativo opera correctamente
Bajo	Una pérdida menor de funcionalidad o un problema al cual se le puede dar solución temporal	Errores de usabilidad, pantallas o reportes de errores que no afectan la calidad, el uso ni la funcionalidad del sistema.
Trivial	Cambio de forma menor.	Un problema cosmético, como puede ser una falta de ortografía o un texto desalineado.

6. RESULTADOS

En cuanto a los resultados obtenidos por parte del practicante a lo largo de su participación en el proyecto se evidencian significativos avances en cuanto a las dos actividades más importantes que desarrollo: Ejecución de casos de prueba y diseño de casos y set de pruebas.

Cabe aclarar que los resultados de un ejecutor de pruebas se miden con base en los defectos que detecta y el número de ejecuciones válidas, entiéndase por válidas a aquellos casos de prueba que se ejecutan de forma correcta teniendo en cuenta que las incidencias que reporte corresponden realmente a un defecto encontrado y con el nivel de severidad adecuado, dicho dictamen de bugs o incidencias bien reportadas es realizado entre el desarrollador involucrado en la sección o módulo que se está probando y el ejecutor que reportó el hallazgo de la incidencia. Se debe tener en cuenta los estados finales de cada uno de los defectos que se reportaron, ya sea Cerrado o Cancelado. En menor medida se presentan defectos pospuestos los cuales serán solucionados en próximos controles de cambios.

Los resultados obtenidos de la labor de ejecución de casos de prueba fueron extraídos de la herramienta software de gestión de calidad de las pruebas manejada por el equipo de trabajo Cidlis-Uis. No solo contribuye a llevar un historial y un registro de todos los procesos adelantados por el equipo de trabajo, sino que ayuda a estandarizar las pruebas, resolver defectos para optimizar la calidad, generación de informes en poco tiempo lo cual representa un ahorro económico. Los datos son confidenciales y por ende solo se presentan porcentajes filtrados por información producto del trabajo del estudiante.

6.1 DEFECTOS ENCONTRADOS

A continuación, se relacionan la cantidad de incidencias reportadas en la herramienta de gestión incidencias Mantis a lo largo del periodo de tiempo en el que el practicante se desempeñó en la labor de ejecución de pruebas, además se muestra el estado del caso de prueba sobre el cual se reportó la incidencia y el nivel de severidad de la misma.

Tabla 6 Resumen de incidencias reportadas en la herramienta de gestión de defectos mantis (Tomada directamente de la herramienta)

Resumen

por Estado	Abiertas	Resueltas	Cerradas	Total
asignada	2	-	-	2
resuelta	-	125	-	125
cerrada	-	-	149	149

por Severidad	Abiertas	Resueltas	Cerradas	Total
trivial	0	8	15	23
bajo	0	35	20	55
media	1	48	73	122
mayor	1	26	33	60
crítico	0	8	8	16

por Categoría	Abiertas	Resueltas	Cerradas	Total
General	0	0	1	1
Pruebas Aceptación	2	125	148	275

6.2 EJECUCIONES DE PRUEBA

Durante su periodo de práctica, el estudiante compartió actividades con otros 5 ejecutores (2 del equipo de trabajo Cidlis-Uis y otros 3 de Colciencias) para un total de 6 recursos en tareas de ejecución de casos de prueba. Tanto la ejecución de casos de prueba como el reporte de incidencias o hallazgos encontrados son unas de las actividades más importantes del proyecto, dado que son el mayor indicador de facturación del mismo.

Teniendo en cuenta que el alcance inicial del proyecto contemplaba la ejecución de casos de prueba correspondientes al paquete 2 – iteraciones 1 y 2 del plan maestro de pruebas, dicha actividad no se pudo realizar, dado que, debido a estrategias de ejecución orientadas a las buenas prácticas y el buen manejo del contenido a probar se decidió que el practicante realizara ejecuciones de casos de prueba correspondientes a los casos de uso del paquete 1 – iteraciones 2 y 3.

En la siguiente imagen se evidencia la cantidad de casos de prueba ejecutados por parte del estudiante a lo largo de su periodo de práctica (esta imagen fue tomada de la herramienta de gestión de pruebas Testlink).

Figura 7 Relación de casos de prueba ejecutados y estados reportados

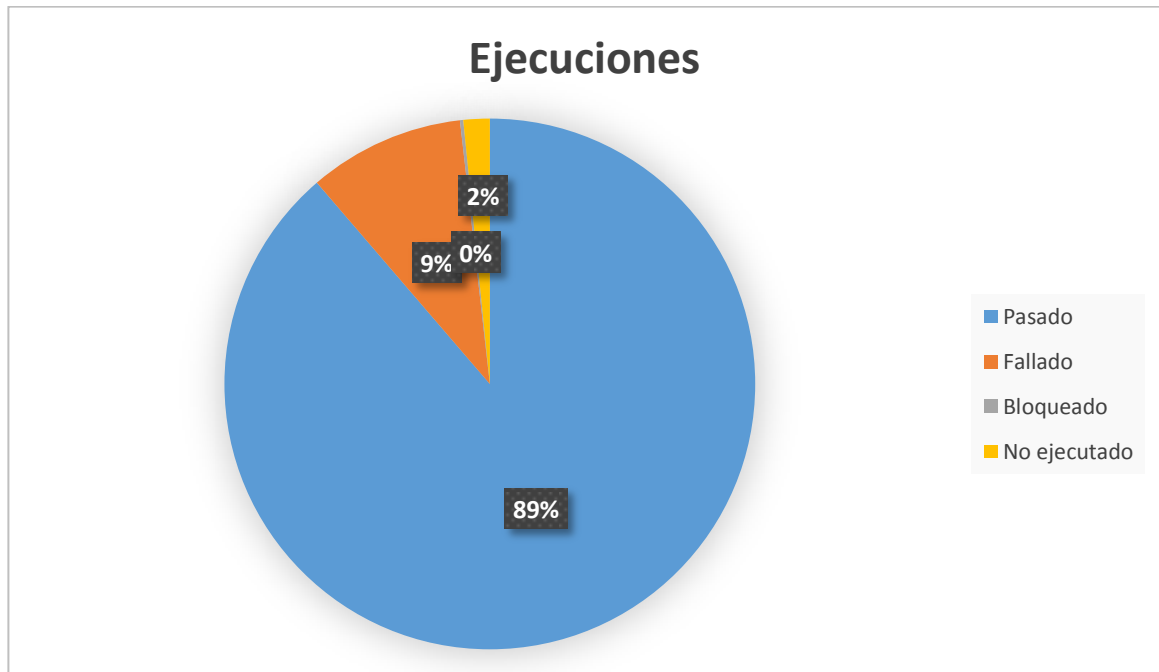


Usuario	Asignado	No Ejecutado	No Ejecuta...	Pasado	Pasado [%]	Fallado	Fallado [%]	Bloqu...	Bloqueado [...]	Progreso [%]
Build: 0.3.2 - Progreso 96.85% - Total time (hh:mm:ss) 00:00:00 (2 Items)										
ejusu02	568	9	1.6	504	88.7	54	9.5	1	0.2	98.4

Dado que el manejo de la información contenida en los casos de uso y la información detallada relacionada con la ejecución de los casos de prueba es restringida, la empresa cliente (Colciencias) autorizó al practicante la publicación algunas cifras que facilitarían la entrega de resultados requerida.

En total, el practicante logró ejecutar 568 pasos de casos de prueba. Cabe resaltar que el usuario asignado por parte del equipo de trabajo Cidlis-Uis tanto para el manejo de la herramienta de gestión de las pruebas Testlink como para la herramienta de gestión de defectos Mantis fue: ej.usu02.

Figura 8 Resultados de ejecución



6.3 DISEÑOS DE CASOS DE PRUEBA

Al igual que la actividad de ejecución, el diseño de casos de prueba es una de las actividades más importantes realizadas a lo largo del proyecto, dado que dependiendo de la cantidad de pruebas ejecutadas y diseños realizados por semana se realizan las evaluaciones por parte del cliente (Colciencias) en cuanto a la eficiencia de la labor realizada por parte del grupo de trabajo de Cidlis-Uis.

A continuación, se relacionan la cantidad de diseños de casos de prueba realizados por parte del practicante durante el periodo de tiempo que desarrolló esta actividad.

Tabla 7 Resultados Obtenidos

Pruebas Funcionales		
Paquete	Iteraciones	Número de casos de uso
Paquete 2	Iteración 1	20
	Iteración 2	37
Total		57

La realización de la labor programada de diseños de casos de prueba se basó en las estimaciones de tiempos definidas por el equipo de trabajo de Cidlis-Uis y en especial del líder de pruebas funcionales quién determinó que para la asignación de la carga de actividades diarias del practicante este debía diseñar 2 casos de prueba al día. De esta forma el estudiante logró cumplir con el compromiso de diseñar 57 casos de prueba a lo largo del periodo de práctica. Cabe resaltar que dicha actividad contemplaba tanto el diseño de un documento en Word el cual era revisado por el analista de pruebas a cargo del practicante, como también el cargue correspondiente en la herramienta de gestión de pruebas Testlink.

Teniendo en cuenta el alcance establecido al inicio de la práctica, se evidencia que el estudiante cumplió a cabalidad con el objetivo trazado en cuanto a la labor de diseño de casos de prueba. Adicionalmente a la actividad de diseño realizada en el paquete 2 – iteraciones 1 y 2 el practicante contribuyó altamente en el desarrollo de los diseños correspondientes a la iteración 3 del paquete 2. Como evidencia de la actividad realizada ver anexo 1.

A continuación, se relacionan la cantidad de casos de prueba en los que el estudiante contribuyó para la elaboración de los diseños, debido al conocimiento

adquirido por los diseños realizados con anterioridad, se permitía y requería un asesoramiento al analista a cargo en dicha actividad.

Tabla 8 Resultados Obtenidos

Pruebas Funcionales			
Paquete	Iteraciones	Número de casos de uso Diseñados	Número de casos de uso Totales
Paquete 2	Iteración 3	10	22

7. CONCLUSIONES

- Se logró ser miembro integral del equipo de trabajo del proyecto, siendo asignado como probador y analista funcional durante todo el periodo que duró la práctica empresarial.
- Se alcanzó un conocimiento sólido en el proceso de calidad de software en un plano empresarial de grandes medidas, apoyando las labores desarrolladas por el equipo de trabajo Cidlis-Uis.
- Los resultados obtenidos de las tareas y labores asignadas como practicante, sirvieron como carta de presentación del estudiante para ser requerido por el equipo de trabajo de Cidlis-Uis en el cargo de analista de pruebas funcionales hasta la terminación del contrato entre Cidlis-Uis y Colciencias.
- Se logró cumplir con la totalidad de objetivos trazados al inicio de la práctica empresarial, tanto para la cantidad de diseños de casos de prueba propuestos como para las ejecuciones de casos de pruebas.

8. RECOMENDACIONES

- Ampliar el alcance de funciones y responsabilidades para alcanzar un cargo de mayor jerarquía que permita el desarrollo de las capacidades de liderazgo y gestión del equipo en proyectos gerenciales.
- Lograr la certificación de primer nivel de probador básico bajo las temáticas de fundamentos de pruebas dictadas por la International Software Testing Qualifications Board (ISTQB) y continuar con el crecimiento profesional en el área de la calidad del software.
- Ampliar el alcance de funciones y responsabilidades para alcanzar un cargo de mayor jerarquía que permita el desarrollo de las capacidades de liderazgo y gestión del equipo en proyectos gerenciales.

BIBLIOGRAFIA

BEIZER, Boris. Software System Testing and Quality Assurance. Nueva York, Estados Unidos de América, 1984.

PRESSMAN, Roger S.. Ingeniería del software Un enfoque práctico (Séptima ed.). México D.F.: McGraw-Hill. 2010. 337p

SOMMERVILLE, Ian. Ingeniería del software (Séptima ed.). Madrid: Pearson Educación S.A.. 2005. 469 p.

THOMAS, Müller. International Software Testing Qualifications Board, ISTQB. Foundation Level Syllabus, Probador certificado: Programa de estudio de nivel básico, 2010. 13 p.

ANEXOS

Anexo A

Ejemplo de diseño de caso de prueba correspondiente al caso de uso: PER04 - CTGC2 - Registrar información de Productos (Capítulo en Libro Resultado de Investigación) v 2.1 del paquete 2 iteración 1.

Tabla 9 Diseño de Caso de prueba

Nombre: PER04 - CTGC2 - Registrar información de Productos (Capítulo en Libro Resultado de Investigación) v 2.1	
Descripción: Permite a una persona crear, modificar, consultar y eliminar información de productos de tipo Capítulo en Libro Resultado de Investigación.	
Actores: Persona	
Precondiciones: El actor ha ejecutado el Caso de Uso COM20: Ingresar al sistema.	
Descripción CU	Resultados esperados CP
<p>1. Ingresar al sistema con rol de persona y seleccionar “Mis productos”/ Producción Bibliográfica / Libro Resultado de Investigación,</p> <p>El usuario selecciona el tipo de producto <i>Capítulo en Libro Resultado de Investigación</i>.</p>	El sistema muestra las pestañas indicadas.
<p>2. Ingresar al sistema con rol de persona y seleccionar “Mis productos”/Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto <i>Capítulo en Libro Resultado de Investigación</i>.</p> <p>Validar que el sistema lista los productos registrados para el tipo de producto mostrando los campos:</p>	El sistema recibe, guarda y valida la información ingresada en el formulario.

<ul style="list-style-type: none"> • Número (Consecutivo, automático) • Título • Año <p>Validar que por cada registro, se presenta la opción Detalles, Modificar, Eliminar y una opción “Nuevo Registro”.</p> <p>NOTA: la validación de campos se da al finalizar el formulario; Guardar y verificar si la información ingresada fue guardada exitosamente, los campos obligatorios cumplen su objetivo. Si los datos son inválidos, el sistema debe mostrar un mensaje de error.</p>	
<p>3. Ingresar al sistema con rol de persona y seleccionar “Mis productos”/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario selecciona la opción “Nuevo Registro”.</p> <p>Validar que para la opción “Nuevo Registro”, el sistema despliega un siguiente formulario, permite Registrar y guardar información para tipo de producto “Capítulo en Libro Resultado de Investigación”,</p>	<p>El sistema recibe, guarda y valida la información ingresada. El sistema permite actualizar la información en el listado de productos.</p>
<p>4. El usuario ha ingresado al sistema con rol de persona y ha seleccionado: “Mis productos”/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción “Nuevo Registro”.</p>	<p>El campo cumple con la condición descrita.</p>

<p>Verificar las condiciones de los campos de acuerdo a la tabla “Producción Bibliográfica”, validando nombre del campo, tipo de dato, longitud y valor posible, obligatoriedad y descripción.</p> <p>Para el campo “Título del capítulo” permita ingresar datos alfanuméricos.</p> <p>NOTA: la validación de campos se da al finalizar el formulario; Guardar y verificar si la información ingresada fue guardada exitosamente, los campos obligatorios cumplen su objetivo. Si los datos son inválidos, el sistema debe mostrar un mensaje de error.</p>	
<p>5. El usuario ha ingresado al sistema con rol de persona y ha seleccionado: “Mis productos”/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción “Nuevo Registro”.</p> <p>Validar el campo “fecha de publicación”</p> <p>Validar que el campo de año de la “Fecha de publicación” del Capítulo en Libro Resultado de Investigación es menor o igual al año de citación o reseña de todas las referencias en revista o libros vinculadas al producto.</p> <p>Validar que en Referencias en Revistas no Válidas: se debe resaltar las Referencias en Revistas que incumplen con el año de la</p>	<p>Los campos cumplen con las condiciones referidas.</p>

<p>fecha de publicación del Capítulo en Libro Resultado de Investigación debe ser menor o igual al año de citación o reseña de todas las referencias en revista o libros vinculadas al producto y verificar que se debe mostrar mensaje al usuario indicando el error.</p> <p>Validar que el campo de la “fecha de publicación” debe ser mayor a 1900 y a la fecha de nacimiento del usuario y menor o igual a la fecha actual. En caso de no se seleccionar día se debe asignar 01.</p> <p>NOTA: la validación de campos se da al finalizar el formulario; Guardar y verificar si la información ingresada fue guardada exitosamente, los campos obligatorios cumplen su objetivo. Si los datos son inválidos, el sistema debe mostrar un mensaje de error.</p>	
<p>6. El usuario ha ingresado al sistema con rol de persona y ha seleccionado: “Mis productos”/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción “Nuevo Registro”.</p> <p>Validar que el campo “Libro” invoca componente Libro. Una vez seleccionado el Libro, muestra en pantalla el campo Título del Libro.</p> <p>Validar el Componente Administrar Libros desde el caso de uso GEN-CMPXX: Administrar Libros, con las particularidades del caso de uso</p>	<p>El sistema cumple con las condiciones del campo.</p> <p>El sistema debe comportarse con lo descrito en el caso de uso: GEN-CMPXX: Administrar Libros, con las particularidades del caso de uso PER04_CTG_C2.</p>

<p>PER04_CTG_C2.</p> <p>7. El usuario ha ingresado al sistema con rol de persona y ha seleccionado: “Mis productos”/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción “Nuevo Registro”.</p> <p>Validar que el campo “Página Inicial” es un número entero positivo mayor a cero.</p> <p>NOTA: la validación de campos se da al finalizar el formulario; Guardar y verificar si la información ingresada fue guardada exitosamente, los campos obligatorios cumplen su objetivo. Si los datos son inválidos, el sistema debe mostrar un mensaje de error.</p>	<p>El campo cumple con la condición referida.</p>
<p>8. El usuario ha ingresado al sistema con rol de persona y ha seleccionado: “Mis productos”/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción “Nuevo Registro”.</p> <p>Validar que el campo “Pagina Final” es mayor o igual al campo Página Inicial.</p> <p>NOTA: la validación de campos se da al finalizar el formulario; Guardar y verificar si la información ingresada fue guardada exitosamente, los campos obligatorios cumplen su objetivo. Si los datos son inválidos, el sistema</p>	<p>El campo cumple con la condición referida.</p>

<p>debe mostrar un mensaje de error.</p> <p>9. El usuario ha ingresado al sistema con rol de persona y ha seleccionado: "Mis productos"/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción "Nuevo Registro".</p> <p>Validar el campo "número de páginas" tiene condición no editable y está calculado como Página Final – Página Inicial +1.</p> <p>NOTA: la validación de campos se da al finalizar el formulario; Guardar y verificar si la información ingresada fue guardada exitosamente, los campos obligatorios cumplen su objetivo. Si los datos son inválidos, el sistema debe mostrar un mensaje de error.</p>	<p>El campo cumple con la condición referida.</p>
<p>10. El usuario ha ingresado al sistema con rol de persona y ha seleccionado: "Mis productos"/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción "Nuevo Registro".</p> <p>Validar que el campo "Referencia en Revistas, invoca el componente Administrar Referencia indicando el tipo de la Referencia Revista.</p> <p>La tabla Referencia en Revistas muestra Nombre de la Revista, Año de Citación y opción Editar y Eliminar.</p>	<p>El sistema cumple con las condiciones del campo.</p> <p>La opción Desvincular elimina la relación entre la referencia y el producto.</p> <p>El sistema debe comportarse según lo descrito en el caso de uso "GEN-CMPXX Admin Ref v1.1"</p>

<p>La opción Editar invoca flujo alternativo A1 del componente Administrar Referencia. Validar la opción Desvincular. Validar que para “Referencias en Revistas” no válidas, el sistema identifica las Referencias en Revistas no válidas y muestra mensaje al usuario indicando el error.</p> <p>Para registrar la referencia en libro y referencia en revista se debe invocar el componente Administrar Referencia indicando el tipo de la Referencia si es Libro o Revista</p> <p>NOTA: la validación de campos se da al finalizar el formulario; Guardar y verificar sí la información ingresada fue guardada exitosamente, los campos obligatorios cumplen su objetivo. Sí los datos son inválidos, el sistema debe mostrar un mensaje de error.</p> <p>Verificar el componente Administrar Referencia desde el caso de uso "GEN-CMPXX Admin Ref v1.1" con las particularidades de Revista para el caso de uso PER04-CTGC2.</p>	
<p>11. El usuario ha ingresado al sistema con rol de persona y ha seleccionado: “Mis productos”/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción “Nuevo Registro”.</p> <p>Validar que el campo “Referencia</p>	<p>El sistema cumple con las condiciones del campo. La opción Desvincular elimina la relación entre la referencia y el producto.</p> <p>El sistema debe comportarse según lo descrito en el caso de uso "GEN-CMPXX Admin Ref v1.1"</p>

<p>en "Libro" invoca el componente Administrar Referencia. Se debe indicar al componente que se registrarán Referencias en "Libro".</p> <p>La tabla Referencia en Revistas muestra Título del Libro, Año de Citación y opción Editar y Eliminar. La opción Editar invoca flujo alternativo A1 del componente Administrar Referencia.</p> <p>Validar la opción Desvincular.</p> <p>Validar que para "Referencias en Libro" no válidas, el sistema identifica las Referencias en Revistas no válidas y muestra mensaje al usuario indicando el error.</p> <p>Para registrar la referencia en libro y referencia en revista se debe invocar el componente Administrar Referencia indicando el tipo de la Referencia si es Libro o Revista</p> <p>NOTA: la validación de campos se da al finalizar el formulario; Guardar y verificar si la información ingresada fue guardada exitosamente, los campos obligatorios cumplen su objetivo. Si los datos son inválidos, el sistema debe mostrar un mensaje de error.</p> <p>Verificar el componente Administrar Referencia desde el caso de uso "GEN-CMPXX Admin Ref v1.1" con las particularidades de Libro para el caso de uso PER04-CTGC2.</p>	
<p>12. El usuario ha ingresado al sistema con rol de persona y ha</p>	<p>Invoca componente Personas. La opción Desvincular elimina la</p>

<p>seleccionado: “Mis productos”/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción “Nuevo Registro”.</p> <p>Validar que el campo “Coautores” invoca el componente Personas. La tabla Coautores muestra nombre completo del integrante, opción Desvincular.</p> <p>Validar el Componente personas desde el caso de uso GEN-CMPXX Gestionar Personas v1 0 con las particularidades del caso de uso PER04_CTG_C2.</p> <p>NOTA: la validación de campos se da al finalizar el formulario; Guardar y verificar si la información ingresada fue guardada exitosamente, los campos obligatorios cumplen su objetivo. Si los datos son inválidos, el sistema debe mostrar un mensaje de error.</p>	<p>relación entre el integrante y el producto.</p> <p>El sistema debe comportarse con lo descrito en el caso GEN-CMPXX Gestionar Personas v1 0 con las particularidades del caso de uso PER04_CTG_C2.</p>
<p>13.El usuario ha ingresado al sistema con rol de persona y ha seleccionado: “Mis productos”/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción “Nuevo Registro”.</p> <p>Validar que el campo “Áreas de Conocimiento” invoca el componente Áreas de</p>	<p>Invoca componente Áreas de Conocimiento. La opción Desvincular elimina la relación entre el área de conocimiento y el producto.</p> <p>El sistema debe comportarse con lo descrito en el caso GEN-CMPXX Administrar Áreas de Conocimiento v1 0 con las particularidades del caso de uso PER04_CTG_C2.</p>

<p>Conocimiento. La tabla Áreas de Conocimiento muestra Gran Área, Área y Disciplina y opción Desvincular.</p> <p>Validar el Componente Areas de Conocimiento desde el caso de uso GEN-CMPXX Administrar Areas de Conocimiento v1 0 con las particularidades del caso de uso PER04_CTG_C2.</p> <p>NOTA: la validación de campos se da al finalizar el formulario; Guardar y verificar si la información ingresada fue guardada exitosamente, los campos obligatorios cumplen su objetivo. Si los datos son inválidos, el sistema debe mostrar un mensaje de error.</p>	
<p>14. El usuario ha ingresado al sistema con rol de persona y ha seleccionado: “Mis productos”/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción “Nuevo Registro”.</p> <p>Validar las condiciones del campo “Palabras Clave”, el cual Invoca componente Palabras Clave. La tabla Palabras clave muestra la palabra clave y la opción desvincular.</p> <p>La opción Desvincular elimina la relación entre el producto y la palabra clave.</p>	<p>Invoca componente Palabras Clave. La opción Desvincular elimina la relación entre el producto y la palabra clave</p> <p>El sistema debe comportarse con lo descrito en el caso de GEN-CMNXX Gestionar Palabras Clave v1 1 con las particularidades del caso de uso PER04_CTG_C2.</p>

<p>Validar el Componente Palabras Clave desde el caso de GEN-CMNXX Gestionar Palabras Clave v1 1 con las particularidades del caso de uso PER04_CTG_C2.</p> <p>NOTA: la validación de campos se da al finalizar el formulario; Guardar y verificar si la información ingresada fue guardada exitosamente, los campos obligatorios cumplen su objetivo. Si los datos son inválidos, el sistema debe mostrar un mensaje de error.</p>	
<p>15. El usuario ha ingresado al sistema con rol de persona y ha seleccionado: “Mis productos”/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción “Nuevo Registro”.</p> <p>Validar las condiciones del campo “Reconocimientos” el cual invoca el componente Reconocimiento. La tabla Reconocimiento muestra Nombre del Reconocimiento, Año de obtención y opción Desvincular.</p> <p>Validar el Componente Reconocimientos desde el caso de GEN-CMNXX Gestionar Reconocimientos v1 1 con las particularidades del caso de uso PER04_CTG_C2.</p> <p>NOTA: la validación de campos se</p>	<p>Se invoca componente Reconocimientos. La opción Desvincular elimina la relación entre el reconocimiento y el producto. El sistema debe comportarse con lo descrito en el caso de GEN-CMNXX Gestionar Reconocimientos v1 1 con las particularidades del caso de uso PER04_CTG_C2.</p>

<p>da al finalizar el formulario; Guardar y verificar sí la información ingresada fue guardada exitosamente, los campos obligatorios cumplen su objetivo. Si los datos son inválidos, el sistema debe mostrar un mensaje de error.</p>	
<p>16. El usuario ha ingresado al sistema con rol de persona y ha seleccionado: “Mis productos”/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción “Nuevo Registro”.</p> <p>Validar que el sistema identificará y suprimirá los espacios adicionales entre palabras que el usuario ingrese de modo que se pueda hacer validación precisa e identificar claramente si existen productos duplicados o inválidos.</p> <p>Validar que no se registren productos duplicados para la persona que está registrando el producto. Un Capítulo de Libro Resultado de Investigación es duplicado cuando los campos Título del Capítulo, Fecha de Publicación, Libro, Página Inicial, Página Final sean iguales a otro registrado por la misma persona.</p>	<p>El sistema muestra un mensaje de error indicando duplicidad de los productos.</p>
<p>17. El usuario ha ingresado al sistema con rol de persona y ha seleccionado: “Mis productos”/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción “Nuevo Registro”.</p>	<p>El sistema carga la información del registro seleccionado, mostrando todos los campos en modo solo lectura.</p>

<p>Detalles</p> <p>El usuario selecciona la opción Detalles del registro que desea consultar</p> <p>Validar que el sistema carga la información del registro seleccionado, mostrando todos los campos en modo solo lectura.</p> <p>Validar que el usuario visualiza la información y selecciona la opción Volver.</p> <p>Verificar que el sistema regresa al usuario a la lista de los productos registrados para el tipo de producto mostrando los campos:</p> <ul style="list-style-type: none"> - Número (Consecutivo, automático) - Título - Año <p>Por cada registro, se presenta la opción Detalles, Modificar, Eliminar. y una opción "Nuevo Registro"</p>	
<p>18. El usuario ha ingresado al sistema con rol de persona y ha seleccionado: "Mis productos"/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción "Nuevo Registro".</p>	<p>El sistema almacena la nueva información y muestra el listado de productos.</p>

<p>Modificar</p> <p>El usuario selecciona la opción Modificar del registro que desea.</p> <p>Validar que el sistema carga toda la información del registro en el formulario permitiendo la edición de los datos.</p> <p>Validar que el usuario modifica la información del formulario y selecciona la opción Guardar de acuerdo con la Tabla de Campos Producción Bibliográfica.</p> <p>NOTA: la validación de campos se da al finalizar el formulario; Guardar y verificar si la información ingresada fue guardada exitosamente, los campos obligatorios cumplen su objetivo. Si los datos son inválidos, el sistema debe mostrar un mensaje de error.</p> <p>Validar que el usuario visualiza la información y selecciona la opción Volver.</p> <p>Verificar que el sistema regresa al usuario a la lista de los productos registrados para el tipo de producto mostrando los campos:</p> <ul style="list-style-type: none">- Número (Consecutivo, automático)- Título- Año <p>Por cada registro, se presenta la opción Detalles, Modificar, Eliminar.</p>	
---	--

<p>y una opción "Nuevo Registro"</p>	
<p>19. El usuario ha ingresado al sistema con rol de persona y ha seleccionado: "Mis productos"/ Producción Bibliográfica / Libro Resultado de Investigación, tipo de producto Capítulo en Libro Resultado de Investigación. El usuario está dentro de la opción "Nuevo Registro".</p> <p>Eliminar</p> <p>El usuario selecciona la opción Eliminar para un registro.</p> <p>Validar que el sistema muestra un cuadro de confirmación de la eliminación del registro</p> <p>El usuario confirma la eliminación del registro.</p> <p>Verificar que el sistema elimina el registro y muestra mensaje del resultado de la acción.</p> <p>Verificar que el sistema regresa al usuario a la lista de los productos registrados para el tipo de producto mostrando los campos:</p> <ul style="list-style-type: none"> - Número (Consecutivo, automático) - Título - Año <p>Por cada registro, se presenta la opción Detalles, Modificar, Eliminar. y una opción "Nuevo</p>	<p>El sistema presenta el mensaje de confirmación de la eliminación del registro.</p> <p>El sistema elimina el registro y muestra mensaje del resultado de la acción.</p>

Registro”	
-----------	--