

Universidad de las Ciencias Informáticas

Facultad 1



*Trabajo de Diploma para optar por el
Título de Ingeniero en Ciencias Informáticas.*

*Título: “Indicadores de calidad en el proceso de pruebas manuales de la
distribución cubana de software libre GNU/Linux Nova”*

Autor:

Rosa María López Rangel

Tutor(a):

Ing. Mónica María Albo Castro

Ciudad de La Habana

Junio 2012

FRASES:

“Cuando puedas medir de lo que estás hablando, y expresarlo en números, entonces conoces algo al respecto; pero cuando no puedas... tu conocimiento es de tipo insatisfactorio y escaso.”

Lord Kelvin, 1889.

“No puedes controlar, lo que no puedes medir...”

De Marco, 1982.

Declaro ser autor de la presente tesis y reconozco al Centro de Informatización Universitaria y a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los _____ días del mes de _____ del año _____.

Rosa María López Rangel

Ing. Mónica María Albo Castro

Firma del autor

Firma del tutor

Agradezco a:

... mi mamita linda por ser la luz de mi vida. Gracias por apoyarme, por ser mi guía, por enseñarme a ser mejor persona cada día. Gracias por ser mi mejor amiga...

... mi tía Verónica que siempre se ha preocupado por mí...

... Juliver y Yurisai por brindarme su cariño...

... mis primitos bellos Andrés y Lorena por extrañarme y quererme tanto...

... mi novio Yoe por ser lo mejor que me ha pasado en la vida. Por brindarme su amor y haceme feliz. Te amo...

... mi familia por parte de madre por preocuparse por mí...

... mi papá por acordarse de mí...

... mis familiares por parte de padre...

... mis amigas de siempre: Lidia, Yanet, Amelia y Milena por formar parte de mi vida...

... mis amigas de la UCI: Sheila, Licet, Lisandra, Isel, Liyanet, Elisa y Yanet por todos los momentos compartidos...

... mis amigos de la UCI: Yoani, Yoandy (pequeño), Leandro y VanBrakle...

... mis compañeros del grupo 8 de primer año, el mejor grupo...

... mi grupo de 5to año...

... todas las amistades que he hecho en la Universidad...

... todos aquellos profesores que me enseñaron y ayudaron a lo largo de mi carrera...

... mi tutora Mónica por ayudarme y enseñarme siempre. Una mejor es difícil de encontrar...

... las chicas del apto 122104. Siempre recordaré su limpieza y organización...

... Fidel y a la Revolución por brindarme la oportunidad de vivir esta experiencia y graduarme de ingeniera...

... Dios por dame fuerzas para seguir adelante sin importar los obstáculos que se presenten y por permitir que mis seres queridos estén siempre conmigo...

Muchas Gracias.

Dedico la tesis a:

... mi madre querida del alma Belkis Rangel Pupo. Por brindarme su amor incondicional. Por ser puro sacrificio y bondad. Por ser mi fuerza y mi razón de ser.

Resumen

En el proceso de desarrollo de un software es de vital importancia aplicar pruebas sistemáticas en sus diferentes etapas, debido a que la probabilidad de cometer errores es bastante alta. El proceso de pruebas debe establecerse respetando los estándares y procedimientos concordados, así como las exigencias del cliente, asegurándose la detección y corrección de la mayor cantidad de errores, lo cual permite liberar un producto con la calidad requerida. Con el objetivo de obtener una mayor calidad de la distribución cubana de software libre GNU/Linux Nova se aplican indicadores de calidad a su proceso de pruebas manuales, como proceso de gran importancia para la revisión de la calidad del software. Para lograr este objetivo se realiza un estudio de los procesos de pruebas manuales de las diferentes distribuciones de software libre, así como los indicadores de calidad definidos en los distintos estándares internacionales.

En el presente trabajo se redefine el proceso de pruebas manuales de la distribución GNU/Linux Nova aplicándole indicadores de calidad que se determinan a partir de los requisitos de calidad que debe cumplir los productos de Nova. Se definen métricas para cada uno de estos indicadores y una nueva actividad en el proceso que es precisamente donde se realiza el cálculo de las mismas para dar una evaluación cuantitativa de los resultados. Se puede constatar la efectividad de la propuesta a partir de los resultados que se muestran luego de aplicarla a una personalización de la distribución que se encontraba en la fase final de su desarrollo.

Palabras claves: distribución GNU/Linux Nova, proceso de pruebas, indicadores de calidad, métricas de calidad.

Índice

Introducción.....	1
Capítulo 1: “Fundamentos teóricos de las pruebas a las distribuciones GNU/Linux.”	5
1.1 Pruebas de Software [7]	5
1.1.1 Diseño de casos de pruebas	6
1.1.2 Estrategias de pruebas del software	6
1.2 Procesos de Pruebas de las Distribuciones Libres.....	7
1.2.1 Proceso de pruebas de la distribución Debian [10]	7
1.2.2 Proceso de pruebas de la distribución Ubuntu Linux [11]	9
1.2.3 Proceso de pruebas de la distribución GNU/Linux Nova [9]	10
1.3 Estándares de Calidad de Software	11
1.3.1 Estándar de Calidad ISO 9001 [14].....	13
1.3.2 ISO/IEC 9126 [15].....	13
1.3.3 NC-ISO/IEC 9126	16
1.3.4 Estándar ISO/IEC-14598 - Evaluación del producto software [16]	18
1.3.5 ISO/IEC 12119:1994 [17]	19
1.3.6 ISO/IEC 15939 - Estándar de Medición Software [18]	19
1.4 Calidad en la Universidad de las Ciencias Informáticas.....	20
Capítulo 2: “Proceso de pruebas manuales para la distribución GNU/Linux Nova”	23
2.1 Indicadores de calidad. Uso de Métricas.	23
2.1.1 Métricas de calidad de software.....	24
2.1.2 Cálculo de las métricas de calidad de software.	31
2.2 Proceso de pruebas actual de la distribución GNU/Linux Nova.	45
2.2.1 Diseño de las pruebas	45
2.2.2 Estrategia de pruebas.....	46
2.3 Proceso de pruebas de la distribución GNU/Linux Nova evaluando los indicadores de calidad. ...	47
Capítulo 3: “Evaluación del proceso de pruebas manuales propuesto.”	54
3.1 Método de Evaluación.	54
3.2 Aplicación de la propuesta.	54
3.2.1 Definición del caso de estudio.....	54

3.2.2 Ejecución de la propuesta	56
3.2.3 Análisis de los resultados	59
3.3 Evaluación de la propuesta	60
Conclusiones.....	62
Recomendaciones	63
Bibliografía	64
Referencias Bibliográficas:	64
Bibliografía Consultada:.....	66
Anexos	67
Anexo 1: Casos De Pruebas	67
Anexo 2: Informe de evaluación	73

Introducción

Las tecnologías de la información facilitan el desarrollo de las actividades diarias de la sociedad, expanden las vías de comunicación entre los seres humanos y brindan formas de almacenamiento de datos. El avance de las tecnologías de la información y las comunicaciones contribuyó a la evolución del desarrollo científico técnico del ser humano, el cual posteriormente inventó las computadoras.

Nuevas técnicas, procedimientos de recopilación y manejo de información fueron surgiendo, eran tan complejos que fueron difíciles de controlar. Siendo evidente la necesidad de encontrar alguna forma de proteger al usuario de la complejidad del hardware, lo cual dio lugar al sistema operativo; que no es más que el programa fundamental de la computadora, el cual controla todos sus recursos y establece la base sobre la que pueden escribirse los programas de aplicación [1]. Creándose aplicaciones de usuarios y, surgiendo a partir de la evolución de estas la industria de software que es la que involucra la investigación, desarrollo, distribución y comercialización de este tipo de productos [2].

Existen sistemas operativos como Windows, un software propietario de los más usados [3], pero que no brinda soporte gratuito, no permite copiarlo, modificarlo, redistribuirlo o mejorarlo, haciendo dependiente al usuario de este producto en específico. Los desarrolladores de software se vieron en la necesidad de crear un software que hiciera posible las cooperaciones entre ellos, eliminando los obstáculos impuestos por los dueños de software privativo [4].

Dando lugar al movimiento de software libre, el cual cogió auge a partir del año 1991, cuando el proyecto GNU liberó su primer sistema operativo libre denominado GNU/Linux. Este estaba conformado por un núcleo (*kernel*) llamado Linux, programa para alojar recursos e interactuar con el hardware, desarrollado por Linus Torvalds¹, y un conjunto de aplicaciones, bibliotecas y herramientas de programación que el proyecto GNU había implementado hasta el momento [5].

El sistema operativo GNU/Linux fue rápidamente divulgado debido a que era de bajo costo de adquisición, su código fuente público y en ocasiones no se tenían que pagar costosas licencias. Permitiendo las libertades del software libre, al poder copiarlas, difundirlas, y modificarlas sin problema alguno, surgieron versiones que se daban a conocer con la misma rapidez, por lo que se le comenzaron a llamar distribuciones GNU/Linux o más conocidas como distribuciones Linux solamente. Una

¹ Estudiante de la Universidad de Helsinki en Finlandia.

distribución de software [6], también conocida como distro, es un compilado de software específico (o una colección de múltiple software, incluso un sistema operativo), ya compilado y configurado.

En el año 2005 se define en Cuba una política de migración hacia el Software Libre con el propósito de lograr la independencia tecnológica y como parte de esta se comenzaron a introducir con más auge los sistemas operativos libres. La Universidad de las Ciencias Informáticas (UCI) ha desempeñado un papel protagónico en este proceso, con el desarrollo del proyecto Nova de la actual Facultad 1 creado en el año 2005. Donde la idea original era crear un entorno de desarrollo más cómodo y flexible para los programadores pero que atendiera a la necesidad tecnológica del país, surgiendo así la distribución creada en este proyecto bajo el mismo nombre.

El proyecto Nova estaba centrado en crear una distribución de software libre que pueda sustituir los sistemas privativos que se usan en las empresas cubanas actualmente. Aunque en sus inicios funcionó como proyecto, continuó desarrollándose hasta convertirse en lo que es hoy, el departamento SODTL(Sistemas Operativos y Desarrollo de Tecnologías Libres) del centro CESOL(Centro de Software Libre) que está integrado por varios proyectos que se encargan no solo de las variantes principales de la distribución (Nova para escritorios, Nova ligero, Nova para servidores), sino también de otros desarrollos que le brindan un valor agregado al producto tales como el Sistema Operativo Base, Osplugger, Xerberos, Seguridad, Arquitectura y Kernel (AK). Especializándose en implementar personalizaciones para entornos específicos de determinados clientes.

Esta distribución como software que es, necesita de un proceso de desarrollo en el cual se realicen de forma sistemática diferentes pruebas para al final obtener un producto con la calidad esperada. El proceso que se defina debe seguir estándares y procedimientos acordados por los desarrolladores del proyecto. Actualmente el proceso de pruebas realizado en Nova solo tiene en cuenta las aplicaciones que incluye la distribución, no mide elementos que la evalúen como sistema operativo, exceptuando algunas pruebas automáticas que evalúan el rendimiento del sistema en determinados hardware.

Existen estándares de calidad que definen los indicadores que evalúan la calidad de determinado tipo de producto de software y la calidad de los procesos de desarrollo de estos. En el actual proceso de pruebas de la distribución GNU/Linux Nova se aplica el estándar CMMI que propone buenas prácticas para establecer procesos de desarrollo de software de buena calidad. Sin embargo no se tiene en cuenta, ningún estándar que permita probar que el producto de software cumpla con indicadores de calidad y por tanto permita evaluar la calidad del producto de forma cuantitativa. Por lo anterior esta

investigación plantea como **problema a resolver**: ¿Cómo obtener una distribución de software libre GNU/Linux Nova que cumpla con los indicadores de calidad mínimos requeridos para un producto de este tipo?

El **objeto de estudio** del presente trabajo de diploma son los indicadores de calidad de software, donde se define como **campo de acción**: los indicadores de calidad de software dentro del proceso de pruebas para la distribución GNU/Linux Nova. Partiendo de esta temática **el objetivo general** es aplicar indicadores de calidad del software en el proceso de pruebas manuales de la distribución cubana de software libre GNU/Linux Nova.

Para alcanzarlo se han propuesto como objetivos específicos los siguientes:

1. Sistematizar la situación actual de los procesos de pruebas de las distribuciones GNU/Linux.
2. Determinar los indicadores de calidad requeridos por las distribuciones GNU/Linux, así como los métodos de medición de los mismos.
3. Determinar los elementos del proceso de pruebas actual de la distribución que cumplen con los indicadores de calidad determinados.
4. Definir el proceso de pruebas manuales de la distribución GNU/Linux Nova.
5. Validar el proceso propuesto para determinar las ventajas y desventajas del mismo.

Con la realización del presente trabajo, se defiende la idea de que con un proceso de pruebas manuales que aplique los indicadores de calidad requeridos por una distribución GNU/Linux, se evaluaría de forma cuantitativa el resultado obtenido en las pruebas y se podría elevar la calidad de los productos que se obtengan en Nova.

Como Métodos Teóricos para esta investigación se utilizan:

- Analítico - Sintético: Durante la investigación se analizan documentos, páginas de Internet y otros artículos de donde se extraen los elementos más importantes relacionados con las pruebas manuales de software para los sistemas operativos libres. Además se realizan resúmenes y valoraciones de las características más relevantes relacionadas con los factores de calidad a tener en cuenta en estas pruebas.
- Inductivo - Deductivo: Según el estudio general de las diferentes pruebas manuales se propone una mejora para hacer más efectivas dichas pruebas.
- Histórico – Lógico: Durante la investigación se estudió la evolución de la calidad, y las pruebas de software. Se realizó un estudio crítico de los trabajos anteriores sobre los procesos de pruebas a

distribuciones libres, para la utilización de los mismos como punto de referencia y comprobación de los resultados alcanzados.

Como Métodos Empíricos para esta investigación se utilizan:

- Experimental: Realiza ejemplos reales en los que se presenta una historia positiva sobre los beneficios que un producto o servicio le ha traído a determinados usuarios. Se pone en práctica el proceso diseñado para al menos una iteración de las pruebas de un caso de estudio seleccionado, que puede ser una personalización o una versión de la distribución.

Como técnica de recopilación de información para esta investigación se utiliza:

- Entrevista: Se pone en práctica a especialistas, para poder adquirir información sobre los estándares de calidad y su utilización en la UCI, así como de los requisitos de calidad de la distribución GNU/Linux Nova. Además para recopilar los datos referentes al caso de estudio que se utiliza para evaluar la propuesta se entrevista a los desarrolladores de la personalización NovaMedia.

El presente documento se ha estructurado en capítulos de la siguiente forma:

- Capítulo 1: *Fundamentos teóricos de las pruebas a las distribuciones GNU/Linux*: se realiza un estudio de los procesos que existen para realizar pruebas a sistemas operativos libres, enunciándose los conceptos básicos necesarios para comprender el funcionamiento de dichos procesos.
- Capítulo 2: *Proceso de pruebas manuales para la distribución GNU/Linux Nova*: definir un proceso de pruebas manuales para la distribución GNU/Linux Nova que tenga presente indicadores de calidad, además de explicar cómo medir los mismos.
- Capítulo 3: *Evaluación del proceso de pruebas manuales propuesto*: evaluación de la efectividad de este proceso, así como, su impacto en la calidad de los productos de la distribución cubana de software libre GNU/Linux Nova.

Capítulo 1: “Fundamentos teóricos de las pruebas a las distribuciones GNU/Linux.”

El desarrollo de sistemas de software implica una serie de actividades de producción donde las posibilidades de que aparezca el fallo humano son enormes. Los errores pueden empezar a darse desde el inicio del proceso, en el que los objetivos pueden estar especificados de forma incorrecta, así como en el diseño y desarrollo posteriores. Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo de software ha de ir acompañado de una actividad que garantice la calidad. Las pruebas del software son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación.

Las técnicas para encontrar problemas en un programa son extensamente variadas y van desde el uso del ingenio por parte del personal de pruebas hasta herramientas automatizadas que ayudan a aliviar el peso y el costo de tiempo de esta actividad. De nada serviría conocer todas las técnicas de prueba de software, si un programa carece de documentación, el código es confuso, o no se han seguido pasos para la planificación y desarrollo. Es por eso que a continuación se define no solo las técnicas y métodos de prueba sino también los estándares e indicadores de calidad que se van a utilizar, como medir estos últimos y los procesos de prueba de algunas distribuciones libres.

1.1 Pruebas de Software [7]

Las pruebas son un elemento crítico para la calidad del software. La importancia de los costos asociados a los errores, promueve la definición y aplicación de un proceso de pruebas minuciosas y bien planificadas. Las pruebas permiten validar y verificar el software, entendiéndose como validación del software el proceso externo al equipo de desarrollo, que determina si el software satisface los requisitos, y verificación como el proceso interno que determina si los productos de una fase satisfacen las condiciones de la misma.

Este sistema de pruebas, como cualquier otro en ingeniería, puede realizarse de dos formas: a) conociendo la función específica para la que fue diseñado; y b) conociendo el funcionamiento del producto. El primer enfoque se centra en las llamadas pruebas de caja negra y el segundo en las pruebas de caja blanca.

El objetivo principal de realizar una prueba es detectar la mayor cantidad de errores, basándose en un caso de pruebas correctamente diseñado. Por ello es fundamental diseñar pruebas que sistemáticamente descubran diferentes tipos de errores con menor tiempo y esfuerzo. Recordando siempre que las pruebas no pueden asegurar que no existen errores sólo puede mostrar que existen defectos en el software.

1.1.1 Diseño de casos de pruebas

Tal como se menciona puede realizarse siguiendo dos criterios:

1. El conocimiento del funcionamiento del producto (Caja blanca).
2. El conocimiento de la función específica para la que fue diseñado el producto (Caja negra).

Pruebas de Caja Blanca: se desarrolla con el fin de asegurar que todas las piezas del sistema tienen una operación interna que se ajusta a las especificaciones y que todos sus componentes internos se han aprobado en forma adecuada. En las pruebas de caja blanca está la prueba de camino básico y la de la estructura de control.

Pruebas de caja negra: se realizan con el fin de asegurar que el producto es operativo. Ejemplo de ellas son los métodos de prueba basados en grafos, las de partición equivalente, análisis de valores límites y pruebas de comparación.

1.1.2 Estrategias de pruebas del software

La estrategia de pruebas proporciona un plano para el desarrollo del software, para la organización de control de calidad y para el cliente. Es una guía que describe los pasos a seguir como parte de las pruebas, cuándo se deben planificar y realizar esos pasos, y cuánto esfuerzo, tiempo y recursos se van a requerir. Por lo tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de los casos de prueba, la ejecución de las pruebas, la agrupación y la evaluación de los datos resultantes.

Como estrategias de pruebas se realizan las pruebas de unidad, las de integración, las de validación y las del sistema. En el caso del diseño de las pruebas se realiza teniendo en cuenta si las mismas van a ser manuales o automatizadas. Las técnicas para el diseño de los casos de prueba y el método aplicado

a la ejecución de las pruebas se deben definir conjunto con el proceso de desarrollo del software específico [8].

1.2 Procesos de Pruebas de las Distribuciones Libres

El proceso de pruebas es un proceso técnico especializado de investigación que requiere de profesionales altamente capacitados en lenguajes de desarrollo, métodos de pruebas, técnicas de pruebas y herramientas especializadas. El conocimiento que debe manejar un ingeniero de prueba, es muchas veces superior al del desarrollador de software.

En el caso de las distribuciones GNU/Linux la existencia de las comunidades de desarrollo como parte del equipo de diseño, implementación y pruebas ha generado una tendencia a la liberación de versiones mínimamente funcionales, con el objetivo de someterlas a prueba por la comunidad. Aquí se introducen los conceptos de imagen Alpha (α): la cual es una primera versión donde se le añaden nuevas funcionalidades, Beta (β): donde se corrigen nuevos errores encontrados en la etapa anterior en ocasiones se libera a la comunidad para que la pruebe. Testing: es la versión que se libera para que la comunidad lo pruebe y vea si está acorde con sus necesidades, Release: donde todos los errores han sido eliminados y esta lista para ser distribuida. [9]

Entre las diversas distribuciones GNU/Linux existen un conjunto que basan su desarrollo en las comunidades, como es el caso de GNU/Linux Debian. Otras son desarrolladas por empresas, con el apoyo de la comunidad, entre estos casos pueden encontrar a Ubuntu. Por último se encuentran las distribuciones propietarias que poseen generalmente mayor robustez a cambio de un costo monetario por obtener soporte del sistema, caso típico de la distribución Red Hat. Por ello se hace referencia solamente al actual proceso de Nova, al proceso de pruebas de la distribución Ubuntu Linux y se brindan algunas informaciones de Debian; de estos dos últimos atendiendo a lo que está publicado en sus sitios oficiales. No se brinda mas información respecto a otras distribuciones GNU/Linux porque en la mayoría de los casos mencionan el uso de algún proceso de pruebas pero no comparten su documentación.

1.2.1 Proceso de pruebas de la distribución Debian [10]

Debian es un sistema operativo (S.O.) libre, el cual utiliza el núcleo de Linux y la mayoría de sus herramientas básicas provienen del Proyecto GNU. Este S.O. provee 29000 paquetes binarios,

programas precompilados distribuidos en un formato que hace más fácil su gestión, por ello se dice que Debian ofrece más que un S.O. puro.

Desarrolla tres principales distribuciones:

- la distribución "estable",
- la distribución "de prueba", que es a veces "congelada",
- la distribución "inestable".

Junto a estas, está la "antigua estable", esta es la estable anterior, y la "experimental". La "experimental" se utiliza para los paquetes que todavía están en desarrollo y con un alto riesgo de estropear el sistema. También es utilizada por los desarrolladores a quienes les gusta estudiar y probar el software, pero los usuarios no deben usar estos paquetes, ya que pueden ser peligrosos y perjudiciales incluso para los más experimentados.

Los paquetes se instalan en el directorio de la distribución "**prueba**" (testing) después de haber sufrido algunas pruebas en la rama inestable. Estos paquetes deben estar en sintonía con todas las arquitecturas para las cuales se han construido y no deben tener dependencias que sean inestables. Lo cual quiere decir que tienen que tener menos fallos críticos que las versiones actuales que están siendo probadas en la rama inestable. De esta manera, "**prueba**" está siempre a punto de ser una versión candidata.

¿Cómo funciona prueba?

La distribución prueba se genera de forma automática partiendo de la distribución inestable mediante una serie de scripts que intentan pasar los paquetes sobre los que sea razonable pensar que carecen de fallos críticos para su publicación. Además tienen en cuenta que se puedan resolver todas las dependencias de los paquetes en la distribución de prueba.

Un paquete (una versión en particular) pasará a prueba cuando satisfaga todos los siguientes criterios:

1. Debe haber estado en inestable 10, 5 o 2 días, dependiendo de la urgencia marcada al enviarlo;
2. Debe estar compilado y al día en todas las arquitecturas en las que ha sido compilado anteriormente en inestable;
3. Debe tener el mismo número o menos de fallos críticos que la versión actualmente en prueba;

4. Todas sus dependencias deben poder ser satisfechas mediante paquetes ya en prueba o mediante el grupo de paquetes que va a ser instalado al mismo tiempo;
5. La operación de instalar un paquete en prueba no debe afectar adversamente a ningún paquete que ya esté en prueba.

De un paquete que satisface las tres primeras condiciones se dice que es un candidato válido. El script de actualización muestra el momento en que cada paquete pasará de inestable a prueba. La salida es doble:

- Las excusas en la actualización: lista de las versiones de todos los paquetes candidatos y el estado básico de su propagación a prueba; es más corta y tiene mejor aspecto.
- La salida de actualización: constituye la salida completa y sin gestionar de los scripts de prueba, según se van examinando a los candidatos.

1.2.2 Proceso de pruebas de la distribución Ubuntu Linux [11]

La distribución Ubuntu ha creado un conjunto determinado de pruebas con el objetivo fundamental de tratar de mejorar el software a través de pruebas estructuradas y exploratorias de los paquetes y las imágenes (ISO) del sistema. Además de detectar la mayor cantidad de errores como sea posible, antes de que los CDs se liberen al público en general.

Las pruebas se realizan a través de un conjunto de actividades documentadas en la “lista de actividades”. Estas pruebas se efectúan días antes de la liberación de una determinada imagen ISO y se basan en descargar la última versión del producto, grabarla en un CD y probarla. Lo anterior ayuda a detectar varios errores y no conformidades no vistas por los desarrolladores, principalmente en los CD de instalación.

Otro tipo de pruebas son las llamadas pruebas de humo diarias, se realizan sobre el uso continuo del sistema normalmente en una máquina virtual. Estas pruebas se basan en que las regresiones, los errores y fallas generales sean detectados lo más temprano posible antes de que el producto sea liberado. Las pruebas de humo diarias están dadas principalmente a los distintos tipos de instalación, aunque en dependencia del tiempo se prueban también las aplicaciones en los sistemas instalados. Una de las pruebas más importantes son las pruebas de actualizaciones a versiones estables (Stable

Release Update: SRU), debido a que los usuarios esperan un alto grado de estabilidad de estas versiones liberadas. Estas actualizaciones tienen como objetivo fundamental, corregir los errores de alto impacto, tales como: vulnerabilidades de seguridad, errores de integridad de los datos, nuevas versiones de software comercial, entre otros.

Realizan además pruebas para las características principales del Release, las pruebas generales y las de aplicaciones. Para realizar estas pruebas se definen un conjunto de casos de pruebas que se clasifican para probar el soporte de hardware, el sistema, la instalación y las aplicaciones a nivel de escritorio. Se define un plan donde se especifica que hacer, que reportar y cómo hacerlo. En el sitio de pruebas de Ubuntu se encuentra publicado un plan para las pruebas de humo y las pruebas pre-liberación. Dentro de estas últimas se incluyen casos de pruebas para la instalación desde el LiveCD 2, los accesos directos, la suspensión y reanudación, la hibernación, el teclado, distintos tipos de mouse, entre otros.

De esta forma queda conformado el proceso de pruebas de la distribución Ubuntu, que no solo involucra a su equipo de QA (Aseguradores de la calidad, por sus siglas en inglés), sino a toda persona de la comunidad que desee integrarse y por ello exponen las formas en que debe llevarse a cabo esta importante fase del desarrollo. Se considera como uno de los procesos de prueba más versátiles, por su fácil adaptación a las necesidades de la distribución, así como la inclusión de la comunidad al proceso de aseguramiento de la calidad lo cual maximiza las oportunidades de encontrar errores antes de la liberación de una versión estable.

1.2.3 Proceso de pruebas de la distribución GNU/Linux Nova [9]

El proceso de pruebas de la distribución cubana GNU/Linux Nova comienza con la captura de los requerimientos del sistema, por los cuales se formularan los casos de prueba a utilizarse. También se tendrá un cronograma de las tareas o pruebas a ejecutarse, cada una con su respectivo responsable, la

² Un CD autónomo, lo que significa poder iniciar el sistema operativo sin seguir procedimientos complicados ni instalarlo en el ordenador.

fecha de ejecución y los recursos asignados a cada una. Así como un reporte de las no conformidades o errores hallados, donde se encontraron los mismos, junto con una breve descripción, para su posterior corrección. Se cuenta con los casos de pruebas que están clasificados en pruebas de aplicaciones, sistema, soporte de hardware e instalación del sistema.

Este proceso de pruebas a Nova aplica el Modelo W combinado con CMMI, teniendo de esta forma una buena organización del ciclo de pruebas. Partiendo de la selección del producto a probar, pasando por la definición del ambiente de pruebas hasta la selección de los criterios de prueba, definiendo siempre de forma explícita cómo realizar el proceso de pruebas.

Las pruebas unitarias son garantizadas por los desarrolladores de cada componente, debido a que estos son aplicaciones completas en la mayoría de los casos. Solo el núcleo del sistema y la funcionalidad de ejecutarse sobre un CD no lo son. Por lo que se comienza directamente con las pruebas de integración, donde actualmente la única funcionalidad a probar es el trabajo desde un CD la cual consiste en verificar que el LiveCD inicie correctamente. Además se verifica el funcionamiento correcto de las aplicaciones sobre el sistema y que este sea estable. Algunas pruebas de integración del sistema con las aplicaciones incluidas, como es el caso del gestor de paquetes, solo se realizan luego de instalar el sistema en una computadora.

Se realizan pruebas del sistema, y dentro de estas las que se realizan previas a la primera liberación de la distribución. En cuanto a las pruebas de aceptación del usuario son aquellas que se hacen luego de la liberación de la versión Beta y que son ejecutadas tanto por los usuarios de la comunidad como por el equipo de pruebas.

1.3 Estándares de Calidad de Software

Los estándares de calidad son directrices para el aseguramiento externo e interno de la calidad [12]. Siendo muy importante implementarlos debido a que la calidad de un producto no se mide solo al terminarlo. Se debe establecer un sistema de gestión de la calidad durante todo el desarrollo del producto, para ir aplicando indicadores que permitan obtener finalmente un producto de buena calidad. La complejidad de los problemas que hoy día buscan una solución en el software está creciendo tanto que ha retado a la habilidad de desarrollar y mantener el software por parte de las organizaciones

dedicadas a esto. Por una parte las organizaciones quieren ser capaces de desarrollar y entregar software confiable, a tiempo y apegado al presupuesto acordado con el cliente. Por otro lado, está la perspectiva del cliente, el cual quiere saber con certeza que todo lo anterior se cumplirá. Por esto las organizaciones deben buscar una norma, estándar o modelo que pueda ayudarlas a conseguir su meta de calidad (competitividad).

La competitividad no debe ser la única razón por la cual se busque la calidad de un software, hay que darle importancia a cada programa que se desarrolla porque un defecto en un producto de software podría traer graves consecuencias, ocasionando serios daños y hasta perjudicar físicamente a una persona. Los sistemas son cada vez más rápidos, más complejos y automáticos aumentando la posibilidad de ocurrencia de una falla catastrófica a la par del potencial de daño que podría ocasionar. Hay que saber distinguir entre simple y fácil, porque un error simple no necesariamente tiene que ser fácil de encontrar, por tanto hay que estar involucrado en la calidad del producto, al ser responsable del trabajo realizado.

Hay que ser consiente también del costo económico que puede traer consigo la identificación tardía de un defecto, representando un costo adicional. Un error identificado en la misma fase donde se produjo es mucho más barato de resolver que el mismo defecto encontrado en una fase posterior, y aún más caro si éste sale a la luz después que el producto ya ha sido entregado [13].

Para asegurar la buena implementación de cualquier norma o modelo se deben tomar en cuenta tres componentes:

- Las prácticas: deben institucionalizarse.
- Las herramientas: ayudan a las personas a dar seguimiento a las prácticas correspondientes al ser herramientas especializadas. Logrando que las personas no vean el proceso como algo hostil y fastidioso.
- Las personas: deben ser capaces y responsables de seguir cada una de las prácticas que están definidas por toda organización.

Otro punto importante es el ciclo de vida de los procesos; el hecho de haber definido, documentado, medido e institucionalizado los procesos no significan que sean los mejores. Todo proceso está sujeto a cambios, porque tener un proceso que no evoluciona representa más un obstáculo que una ayuda. Un

último punto sería el enfoque con que se ven los procesos, los cuales deben ayudar a lograr un objetivo de la organización más no son ellos mismos el objetivo [13].

Una de las organizaciones generadoras de estándares de reconocimiento mundial es la Organización Internacional para la Estandarización (ISO por sus siglas en inglés). Siendo la agencia internacional especializada en la estandarización, abarcando actualmente los cuerpos nacionales de los estándares de 91 países. La ISO se compone de aproximadamente 180 comités técnicos y cada uno de ellos es responsable de una de las áreas de la especialización. El propósito de la ISO es promover el desarrollo de la estandarización y de las actividades relacionadas en el mundo para facilitar el intercambio internacional de mercancías y de servicios, y para desarrollar la cooperación en actividad intelectual, científica, tecnológica y económica.

1.3.1 Estándar de Calidad ISO 9001 [14]

El estándar que ha sido adoptado por más de 130 países para su uso, se está convirtiendo en el medio principal con el que los clientes pueden juzgar la competencia de un desarrollador de software. Uno de los problemas con el estándar ISO 9001 está en que no es específico de una industria: está expresado en términos generales, y puede ser interpretado por los desarrolladores de diversos productos como cojinetes de bolas, secadores de pelo, automóviles, equipamiento deportivo, televisores, así como por los desarrolladores de software. Se han realizado muchos documentos que relacionan el estándar con la industria del software, pero no entran en una gran cantidad de detalles.

Para la industria del software los estándares relevantes son:

- *ISO 9001*: este es un estándar que describe el sistema de calidad utilizado para mantener el desarrollo de un producto que implique diseño.
- *ISO 9001-3*: este es un documento específico que interpreta el ISO 9001 para el desarrollador de software.
- *ISO 9004-2*: este documento proporciona las directrices para el servicio de facilidades del software como soporte de usuarios.

1.3.2 ISO/IEC 9126 [15]

El estándar ISO/IEC 9126 ha sido desarrollado en un intento de identificar los atributos clave de calidad

para el software. El estándar está dividido en cuatro partes las cuales rigen lo siguiente: modelo de calidad, métricas externas, métricas internas y métricas de calidad en el uso. El modelo de calidad establecido en la primera parte del estándar ISO/IEC 9126-1, clasifica la calidad del software en un conjunto estructurado de características y subcaracterísticas. Donde cada una (como adaptabilidad) está dividida en atributos, estos son una entidad la cual puede ser verificada o medida en el producto software. Los atributos no están definidos en el estándar, ya que varían entre diferentes productos software. La estructura del modelo es la siguiente:

- Funcionalidad - Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas.
 - Idoneidad.
 - Precisión.
 - Interoperabilidad.
- Confiabilidad - Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.
 - Madurez.
 - Recuperabilidad.
 - Tolerancia a fallos.
- Usabilidad - Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.
 - Comprensibilidad.
 - Instructibilidad.
 - Operabilidad.
 - Atractivo.
- Eficiencia - Conjunto de atributos vinculados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.
 - Comportamiento en el tiempo.
 - Utilización de recursos.

- Mantenibilidad - Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema de software.
 - Analizabilidad.
 - Cambiabilidad.
 - Estabilidad.
 - Ensayabilidad.
- Portabilidad - Conjunto de atributos relacionados con la capacidad de un sistema de software para ser transferido desde una plataforma a otra.
 - Adaptabilidad.
 - Instalabilidad.
 - Co-Existencia.
 - Remplazabilidad.

La subcaracterística **Conformidad** no está listada arriba ya que se aplica a todas las características.

El estándar provee un entorno para que las organizaciones definan un modelo de calidad propio para el tipo producto de software que desarrolla. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad las cuales evalúan el grado de presencia de los atributos. Estas métricas pueden ser clasificadas de la siguiente forma:

Métricas internas son aquellas que no dependen de la ejecución del software (medidas estáticas).

Métricas externas son aquellas aplicables al software en ejecución.

La calidad en las métricas de uso están sólo disponibles cuando el producto final es usado en condiciones reales. Idealmente, la calidad interna no necesariamente implica calidad externa y esta a su vez la calidad en el uso.

Este estándar proviene del modelo establecido en 1977 por McCall y sus colegas, los cuales propusieron un modelo para especificar la calidad del software. El modelo de calidad McCall está organizado sobre tres tipos de características de calidad:

- Factores (especificar): Describen la visión externa del software, como es visto por los usuarios.
- Criterios (construir): Describen la visión interna del software, como es visto por el desarrollador.
- Métricas (controlar): Se definen y se usan para proveer una escala y método para la medida.

La ISO/IEC 9126 distingue entre fallo y no conformidad, donde fallo es el incumplimiento de los requisitos previos, mientras que la no conformidad es la violación de los requisitos especificados.

1.3.3 NC-ISO/IEC 9126

Existe actualmente una variante cubana, la NC- ISO/IEC 9126 que es una revisión de la ISO/IEC 9126(1991), conservando las mismas características de calidad de los productos de software. Las diferencias fundamentales son las siguientes:

- la introducción de sub-características normativas, la mayor parte de las cuales están basadas en las sub-características informativas de la ISO/IEC 9126(1991);
- la especificación de un modelo de calidad;
- la introducción de la calidad en el uso;
- la exclusión del proceso de evaluación (que es ahora especificado en la ISO/IEC 14598);
- la coordinación del contenido con la ISO/IEC 14598-1;

En la siguiente tabla se representan las similitudes y diferencias de los estándares 9126 atendiendo a la terminología utilizada para las características y subcaracterísticas de calidad:

<u>ISO/IEC 9126</u>	<u>NC ISO/IEC 9126</u>
Funcionabilidad	Funcionalidad
Idoneidad	Idoneidad
Precisión	Precisión
Interoperabilidad	Interoperabilidad
	Seguridad
Conformidad con la Funcionabilidad	Conformidad con la Funcionalidad
Confiabilidad	Confiabilidad
Madurez	Madurez
Tolerancia al defecto	Tolerancia ante fallos
Recuperabilidad	Recuperabilidad

Conformidad con la Fiabilidad	Conformidad con la Confiabilidad
Usabilidad	Usabilidad
Comprensibilidad	Comprensibilidad
Instructibilidad	Cognoscibilidad
Operabilidad	Operabilidad
Atractivo	Atracción
Conformidad con la Utilizabilidad	Conformidad con la Usabilidad
Eficiencia	Eficiencia
Comportamiento en el tiempo	Rendimiento
Utilización de los recursos	Utilización de los recursos
Conformidad con la Eficacia	Conformidad de la Eficiencia
Mantenibilidad	Mantenibilidad
Analizabilidad	Diagnosticabilidad
Cambiabilidad	Flexibilidad
Estabilidad	Estabilidad
Ensayabilidad	Contrastabilidad
Conformidad con la Mantenibilidad	Conformidad de la Mantenibilidad
Portabilidad	Portabilidad
Adaptabilidad	Adaptabilidad
Instalabilidad	Instalabilidad

Coexistencia	Coexistencia
Reemplazabilidad	Reemplazabilidad
Conformidad con la Portabilidad	Conformidad con la Portabilidad

1.3.4 Estándar ISO/IEC-14598 - Evaluación del producto software [16]

El estándar ISO/IEC 14598 es actualmente usado como base metodológica para la evaluación del producto de software debido a que es necesario considerar mediciones en el proceso empleado para diseñar, desarrollar, probar y controlar el producto. Este ofrece una visión general, define los términos técnicos utilizados, contiene requisitos generales para la especificación y evaluación de la calidad del software, clarifica los conceptos generales y explica la relación entre su serie y el modelo de calidad de la ISO/IEC 9126. Guiando el proceso de valoración de la calidad del software según los criterios de la 9126, por lo cual la ISO/IEC 14598-1 está prevista para que se use conjuntamente con la ISO/IEC 9126-1.

Es importante señalar que la serie de normas ISO/IEC 14598 proporciona un marco de trabajo para evaluar la calidad de todos los tipos de productos de software e indica los requisitos para los métodos de medición y para el proceso de evaluación de los productos de software. La ISO/IEC 14598 describe los requisitos del proceso de evaluación en tres situaciones diferentes: Requisitos para desarrolladores (parte 3), Requisitos para compradores (parte 4), Requisitos para evaluadores (parte 5).

Audiencia destino para la norma ISO/IEC14598:

- proveedores de productos de software,
- compradores de productos de software,
- organizaciones encargadas de las evaluaciones del producto de software,
- usuarios del producto y gente que hace su mantenimiento.

El propósito de la evaluación de la calidad del software es hacer que tanto el desarrollo como la adquisición del software cumplan las expectativas y necesidades del usuario. Esta norma define el proceso de evaluación y provee los requerimientos y las guías que conducen a evaluaciones de calidad, a través de 6 partes. Visión General(Parte 1), Planificación y Gestión (Parte 2), El Proceso para

Desarrolladores (Parte 3), El Proceso para Compradores(Parte 4), El Proceso para Evaluadores (Parte 5), y Documentación de los Módulos de Evaluación (Parte 6).

1.3.5 ISO/IEC 12119:1994 [17]

La ISO/IEC 12119 define las características de calidad para un software COTS (Commercial off the shelf)³. Es aplicable a los paquetes de software y establece un conjunto de requisitos para los mismos (requisitos de calidad); e instrucciones sobre cómo probar un paquete de software con estos requerimientos. Dichas instrucciones son para las pruebas, en particular para las pruebas a terceros.

Se trata sólo de paquetes de software (por ejemplo, productos de software para las funciones administrativas, ofimática y programas de utilidad). No se ocupa de su proceso de producción, donde se incluyen las actividades y productos intermedios. El sistema de calidad desde el punto de vista de un proveedor (por ejemplo ISO 9001:1994), no se contempla en la norma ISO/IEC 12119.

1.3.6 ISO/IEC 15939 - Estándar de Medición Software [18]

ISO/IEC 15939 (Proceso de Medición de Software) norma internacional que se está desarrollando actualmente y que define un proceso de medición de software aplicable a toda la ingeniería relacionada con el software y las disciplinas de gestión. En esta norma dicho proceso se describe a través de un modelo que define las actividades del mismo, que son necesarias para especificar adecuadamente la información requerida. Además, describe cómo las medidas y los resultados de los análisis se van a utilizar, y la forma de determinar la validez de estos resultados dentro de un proyecto o de la estructura organizacional. El proceso de medición es flexible, tolerable, y adaptable a las necesidades de diferentes usuarios.

ISO/IEC 15939:2007 identifica un proceso que apoya la definición de un conjunto adecuado de medidas que abordan las necesidades específicas de información. En él se determinan las actividades y tareas que son necesarias para identificar, definir, seleccionar, aplicar y mejorar la medición dentro de un proyecto global o la estructura de medición de la organización. También proporciona definiciones de los

³COTS: componentes comerciales usados en prácticas de reutilización de software.

términos de medición de uso común dentro del sistema y del software.

El objetivo y los resultados del proceso de medición de ISO/IEC 15939 han sido añadidos a la revisión del estándar ISO 12207 dentro de un nuevo proceso de soporte denominado Medición y a la norma ISO 9003 (aplicación de la norma ISO 9001:2000 al software).

Actualmente, los conceptos del dominio de la medición de ISO/IEC 15939 han sido añadidos al estándar ISO/IEC 15288 (Procesos de Ciclo de Vida del Sistema). De la misma forma, la nueva terminología de la medición ha sido coordinada con las revisiones en los estándares ISO/IEC 9126 (Calidad del Producto Software) e ISO/IEC 14598 (Evaluación de Productos Software) con el objetivo de que todos los estándares que usen el dominio de la medición estén basados en una misma terminología.

El área Medición y Análisis de CMMI proporcionan una metodología para evaluar si un programa de medición de un proyecto es acorde con el estándar ISO 15939, por lo que utiliza este estándar como referencia de entrada.

1.4 Calidad en la Universidad de las Ciencias Informáticas

La Universidad de las Ciencias Informáticas (UCI) en la actualidad produce una gran cantidad de productos de software, razón por la cual se hace necesario controlar la calidad de estos productos a fin de aumentar la valía de los mismos. En función de lograr este objetivo se crea el Centro de Calidad para Soluciones Informáticas (CALISOFT), que es la entidad oficial de la calidad en la universidad además de centro nacional de calidad de software. Dentro de sus funciones está brindar el servicio de pruebas a los productos que se confeccionen en la UCI, así como en el resto de país, incluyendo Nova. CALISOFT cuenta con un Laboratorio Industrial de Pruebas de Software (LIPS) donde se realizan las pruebas a todos los productos que soliciten este servicio, ya sean de la universidad o de empresas desarrolladoras de software del país.

En ese centro se han realizado varias investigaciones en función de los procesos de pruebas de software, de las cuales resalta como referencia importante; el *Catálogo automatizado de métricas de calidad para evaluar los productos en las pruebas*. Este se realizó para aplicarlo al proceso de pruebas de software contribuyendo a obtener medidas cuantitativas de la calidad del producto. El catálogo, basándose en las características y sub-características de calidad del estándar ISO/IEC 9126, es puesto

en práctica en el LIPS, para facilitar la evaluación de los productos de software en las pruebas realizadas. Estos indicadores y métricas son aplicables a productos de software genéricos, sin embargo hay desarrollos específicos, como las distribuciones GNU/Linux y los sistemas operativos que tienen otras características particulares; por lo que no se ajusta a Nova.

CALISOFT también está aplicando un proceso de mejoras enfocado a mantener un proceso de desarrollo con Nivel 2 de CMMI. El proceso de mejoras define las políticas, un ciclo de vida de los proyectos y actividades para llevar a cabo un proceso de desarrollo administrado, las cuales se encuentran dentro de siete áreas de procesos. Dentro del ciclo de vida del proyecto que propone este programa, esta investigación se concentra en las pruebas internas y de liberación. De las áreas de procesos del Nivel 2 la que está relacionada más directamente con la calidad del producto y por consecuencia con el proceso de pruebas es el área de Aseguramiento de la Calidad del Proceso y el Producto (PPQA). Dentro de esta área de procesos las políticas fundamentales son evaluar la adherencia del proceso y los productos, e identificar las no conformidades.

Las actividades descritas en los procesos del programa de mejoras están enfocadas en la calidad del proceso. No obstante un proceso de desarrollo administrado lleva un expediente de proyecto con toda documentación requerida organizada, eso incluye los artefactos del proceso de pruebas. El expediente del programa de mejoras propone como artefactos relacionados o resultantes de las pruebas:

- Diseño de Casos de Pruebas: en nuestro caso el que interesa es el basado en requisitos.
- Planilla de Solicitud de Pruebas de Liberación: se detallan las características del proyecto desde el punto de vista del negocio. Exponiéndose los artefactos a liberar, los requisitos de software y hardware, etc.
- Planilla de Artefactos para las Pruebas de Liberación: es el documento donde se definen los roles y responsabilidades en el proceso de pruebas, el escenario, el modelo de despliegue, los recursos y demás elementos a tener en cuenta para este proceso.
- Plan de Pruebas: es muy similar al anterior con la diferencia que es el plan donde se incluyen tanto las pruebas internas como las de liberación.
- Planilla Problemas, Desviaciones y Acciones: en esta planilla se deben registrar todas las desviaciones y problemas detectados, aunque no se esclarece como varía entre el proceso y el

producto.

El proceso de pruebas actual de la distribución de software libre GNU/Linux Nova está definido de forma genérica y similar a los procesos de las distribuciones libres estudiadas. En ninguno de los procesos de pruebas investigados se analizan o aplican indicadores que midan la calidad de este tipo de productos, por lo que en la presente investigación se continuará con el proceso de la distribución GNU/Linux Nova.

Aunque existen actualmente diversos estándares de calidad de software, en la presente investigación solo se hizo referencia a los que se enfocan en la calidad del producto. Especificando como guía para la definición de los indicadores de calidad de la distribución GNU/Linux Nova la NC-ISO/IEC 9126 y el estándar ISO/IEC 9126 como complemento para la enunciación de las métricas. Al considerarse NC-ISO/IEC 9126 un modelo completo que recoge la mayoría de las características de calidad de los demás modelos de calidad y además es establecido por CALISOFT para las pruebas de software en la UCI.

Capítulo 2: “Proceso de pruebas manuales para la distribución GNU/Linux Nova”

En este capítulo se describe el proceso de pruebas manuales para la distribución cubana de software libre GNU/Linux Nova que se obtiene como propuesta de la presente investigación. Para ello se realiza un análisis crítico del actual proceso de pruebas, determinando cuales indicadores de calidad se evidencian en el mismo y a partir de estos incluir los restantes para verificar la calidad de la distribución. Se explica cómo se van a aplicar los indicadores de calidad escogidos, dentro del proceso de pruebas de Nova, así como las métricas para evaluarlos y el análisis del resultado obtenido en el proceso.

2.1 Indicadores de calidad. Uso de Métricas.

Teniendo en cuenta lo estudiado en los estándares 9126 (la ISO/IEC 9126 y la NC ISO/IEC 9126) y su aplicación por CALISOFT a los procesos de prueba de la UCI, el primer paso para determinar los indicadores de calidad de un producto de software es definir los requisitos de calidad del mismo. De esta forma, una vez definidos los requerimientos de calidad de la distribución, se podrán seleccionar las métricas de calidad para observar el cumplimiento de los mismos y llevar a cabo su evaluación.

Se le asigna a cada indicador de calidad una o más métricas directas que lo representarán y valores de estas que sirvan de requerimientos cuantitativos a dichos indicadores. En la propuesta se definen y seleccionan métricas externas de calidad ya que las mismas están destinadas para medir la calidad del producto software a través de la medición del comportamiento del sistema. Las métricas externas solo pueden ser usadas durante las etapas de pruebas del proceso del ciclo de vida y durante cualquier otra etapa operacional.

Como se menciona antes de la definición de las métricas a utilizar, fue necesario establecer los requerimientos de calidad (RC) genéricos de la distribución GNU/Linux Nova los cuales quedaron conformados de la siguiente forma:

RC 1: Las aplicaciones instaladas deben ser funcionales.

RC 2: Las aplicaciones instaladas deben cumplir con los requisitos.

RC 3: Las opciones del sistema (menú principal, opciones del panel, opciones de energía, etc.) deben de devolver el resultado esperado.

RC 4: El sistema no debe permitir la corrupción de los datos.

- RC 5: El sistema no debe permitir que usuarios no autorizados accedan a la información contenida.
- RC 6: El EE (Entorno de Escritorio) debe estar diseñado he implementado de acuerdo a los requerimientos.
- RC 7: Todas las aplicaciones instaladas, y las opciones del sistema deben ser probadas.
- RC 8: El sistema debe de tener un mecanismo de recuperación ante fallos.
- RC 9: El sistema debe ser capaz de actualizarse sin la ocurrencia de fallos.
- RC 10: Obtener el menor tiempo posible de recuperación del sistema ante cualquier fallo ocurrido.
- RC 11: Obtener el menor tiempo posible de inactividad del sistema ante cualquier fallo ocurrido.
- RC 12: Obtener posibles puntos de restauración ante cualquier fallo ocurrido.
- RC 13: Confeccionar demos, manuales de usuarios y ayudas tanto en el sistema como en formato duro.
- RC 14: Buena descripción de los demos, manuales de usuarios y ayudas confeccionados.
- RC 15: Permitir la mayor personalización del sistema por el usuario.
- RC 16: Permitir un fácil acceso a todas las aplicaciones instaladas y a las opciones del sistema.
- RC 17: Cumplimiento de los requisitos de eficiencia definidos.
- RC 18: Obtener el menor tiempo posible de respuesta del sistema.
- RC 19: El sistema debe funcionar correctamente en las condiciones especificadas en los requerimientos.
- RC 20: El sistema debe permitir el uso de varios dispositivos de hardware y protocolos de red además de los especificados.
- RC 21: Evitar que un fallo ocurrido se convierta en un fallo total del sistema.
- Estos requerimientos de calidad se determinaron a partir de la experiencia del equipo de calidad de la distribución GNU/Linux Nova y se consultaron con el equipo de desarrollo.

2.1.1 Métricas de calidad de software

Métricas de Funcionalidad

Las métricas externas de funcionalidad deben ser capaces de medir como es el comportamiento funcional del sistema.

➤ **Métricas de Idoneidad:**

Las métricas externas de idoneidad deben ser capaces de medir cuan satisfactorio o insatisfactorio es el comportamiento funcional del sistema.

Un funcionamiento insatisfactorio puede ser:

- Funcionamiento que no se desempeña de la forma especificada en el " Manual de usuario " o la especificación de requerimientos.
- Funcionamiento que no provee una salida aceptable o razonable al tomar en consideración un objetivo específico de las tareas del usuario.

Las métricas de calidad de idoneidad son:

- ✓ Métrica # 1: Adecuación funcional (Propone medir: ¿Cuán adecuadas son las aplicaciones instaladas?).
- ✓ Métrica # 2: Completitud del desarrollo funcional (Propone medir: ¿Cuán completa ha sido la construcción del sistema conforme con la especificación de requisitos?).

➤ Métricas de Precisión.

Las métricas externas de precisión o exactitud deben ser capaces de medir como es la frecuencia con que los usuarios se encuentren con la ocurrencia de una falta de precisión en la respuesta del sistema, como pueden ser:

- Inconsistencia entre el procedimiento operación actual y el descrito en el manual de operación.
- Diferencias entre el resultado actual y el razonablemente esperado producto de una tarea ejecutada durante la operación.

Las métricas de calidad de exactitud son:

- ✓ Métrica # 3: Exactitud esperada (Propone medir: ¿Existen diferencias entre los resultados actuales y los razonablemente esperados?).

➤ Métricas de seguridad

Una métrica externa de seguridad debe ser capaz de medir el número de ocurrencias de problemas de seguridad, que son:

- Impedir fugas de información o datos de salida segura.
- Evitar la pérdida de datos importantes.
- Defender contra el acceso ilegal u operaciones ilegales.

Las Métricas de calidad de seguridad son:

- ✓ Métrica # 4: Prevención de la corrupción de los datos (Propone medir: ¿Cuál es la frecuencia de eventos de corrupción de los datos?).
- ✓ Métrica # 5: Capacidad de control de acceso (Propone medir: ¿Cuál es la frecuencia de

accesos ilegales al sistema?).

Métricas de Confiabilidad

Las métricas externas de confiabilidad deben ser capaces de medir el comportamiento del sistema durante la ejecución de las pruebas para indicar la magnitud de la estabilidad del sistema.

➤ Métricas de madurez

Las métricas externas de madurez deben ser capaces de medir la ausencia de ocurrencia de fallos en el sistema.

Las métricas de calidad de la subcaracterística madurez son:

- ✓ Métrica # 6: Densidad de fallos totales contra casos de prueba (Propone medir: ¿Cuántos fallos totales fueron detectados durante un período de pruebas definido?).
- ✓ Métrica # 7: Grado de solución ante fallos totales (Propone medir: ¿Cuántas condiciones de fallo total están resueltas?).
- ✓ Métrica # 8: Densidad de fallos (Propone medir: ¿Cuántos fallos fueron detectados durante un período de pruebas definido?).
- ✓ Métrica # 9: Erradicación de fallos (Propone medir: ¿Cuántos fallos han sido corregidos?).
- ✓ Métrica # 10: Cobertura de las pruebas (Propone medir: ¿Cuántos casos de pruebas requeridos han sido ejecutados durante las pruebas?).
- ✓ Métrica # 11: Madurez de las pruebas (Propone medir: ¿Cuán eficientes son los casos de prueba del producto?).

➤ Métricas de tolerancia ante fallos:

Las métricas externas de tolerancia ante fallos deben estar relacionadas con la capacidad del software de mantener un nivel de ejecución específico en casos de fallos de operación.

- ✓ Métrica # 12: Evitación de operaciones incorrectas o fallos (Propone medir: ¿Cuántas funciones están implementadas con capacidad de evitar operaciones incorrectas o fallos?).

➤ Métricas de Recuperabilidad

Las métricas externas de recuperabilidad deben ser capaces de medir cuan capaz es el sistema de restablecer su nivel adecuado de ejecución y recuperar los datos directamente afectados en casos de fallos totales.

- ✓ Métrica # 13: Grado de disponibilidad (Propone medir: ¿Cuán disponible está el sistema para su

uso según los requerimientos?).

- ✓ Métrica # 14: Tiempo medio de inactividad (Propone medir: ¿Cuál es el tiempo promedio en que el sistema se mantiene no disponible cuando ocurre un fallo total y antes de la arrancada gradual?).
- ✓ Métrica # 15: Restaurabilidad (Propone medir: ¿Cuán capaz es el producto de auto restaurarse luego de un evento anormal o una solicitud?).

Métricas de Usabilidad

Las métricas externas de usabilidad miden la dimensión con que el sistema puede ser comprendido, estudiado, operado, atractivo y acorde con las regulaciones y guías relativas a la usabilidad. Resulta recomendable que la evaluación de estas métricas se haga por un grupo (7 u 8 aunque menores pueden obtener información de utilidad) de usuarios o evaluadores, usuarios simulados o clonados (pero representativos de un rango de usuarios) sin que reciban asistencia externa alguna.

➤ Métricas de Comprensibilidad

Las métricas externas de comprensibilidad deben ser capaces de valorar cómo un nuevo usuario podría comprender si el sistema es idóneo para el entorno al cual se destina:

- ✓ Métrica # 16: Accesibilidad a tutoriales (Propone medir: ¿A qué proporción de demos/tutoriales pueden acceder los usuarios?).
- ✓ Métrica # 17: Comprensibilidad del Entorno de Escritorio (Propone medir: ¿Qué proporción de las funciones de producto el usuario nuevo podrá entender correctamente?).

➤ Métricas de Cognoscibilidad

Una métrica externa de cognoscibilidad o instructibilidad debe ser capaz de evaluar cómo los usuarios pueden aprender a utilizar funciones específicas de forma autónoma y la eficiencia de los sistemas de ayuda y documentación. La instructibilidad está relacionada con la claridad de la documentación y sistemas de ayuda, con la comprensibilidad del EE y la medida en que el usuario puede ser capaz de aprender a partir del propio sistema. Es la capacidad del sistema para permitirle al usuario aprender su aplicación.

- ✓ Métrica # 18: Eficiencia de la documentación del usuario y/o sistemas de ayuda en el uso (Propone medir: ¿Qué proporción de las funciones pueden ser utilizados correctamente después de leer la documentación o el uso de sistemas de ayuda?).

➤ Métricas de Atracción

Las métricas externas de atracción deben ser capaces de evaluar la apariencia del software, y van a estar influenciadas por factores tales como el EE y su diseño.

- ✓ Métrica # 19: Adaptabilidad de la apariencia de la interfaz (Propone medir: ¿Qué proporción de los elementos de la interfaz puede ser, por su apariencia, adaptados por el usuario para satisfacción del mismo?)

Métricas de Mantenibilidad

Una métrica de Mantenibilidad debe ser capaz de medir la capacidad del sistema de sufrir modificaciones, que pueden incluir las correcciones, mejoras o adaptaciones del software a cambios en el ambiente, así como en los requisitos y las especificaciones funcionales.

➤ Métricas de Diagnosticabilidad

Una métrica externa de Diagnosticabilidad debe ser capaz de medir la capacidad del sistema de ser diagnosticado para detectar deficiencias o posibles fallos, o para identificar las partes que van a ser modificadas.

- ✓ Métrica # 20: Cumplimiento de las pruebas (Propone medir: ¿Cuántas aplicaciones u opciones del sistema se han probado?).

➤ Métricas de Flexibilidad

Una métrica externa de Flexibilidad debe ser capaz de medir la capacidad del sistema para permitir modificaciones específicas.

- ✓ Métrica # 21: Flexibilidad de la distribución (Propone medir: ¿Cuántas no conformidades encontradas en las pruebas no han sido resueltas?).

Métricas de Portabilidad

Una métrica externa de portabilidad debe ser capaz de medir el comportamiento del operador o del sistema durante la actividad de portabilidad.

➤ Métricas de adaptabilidad

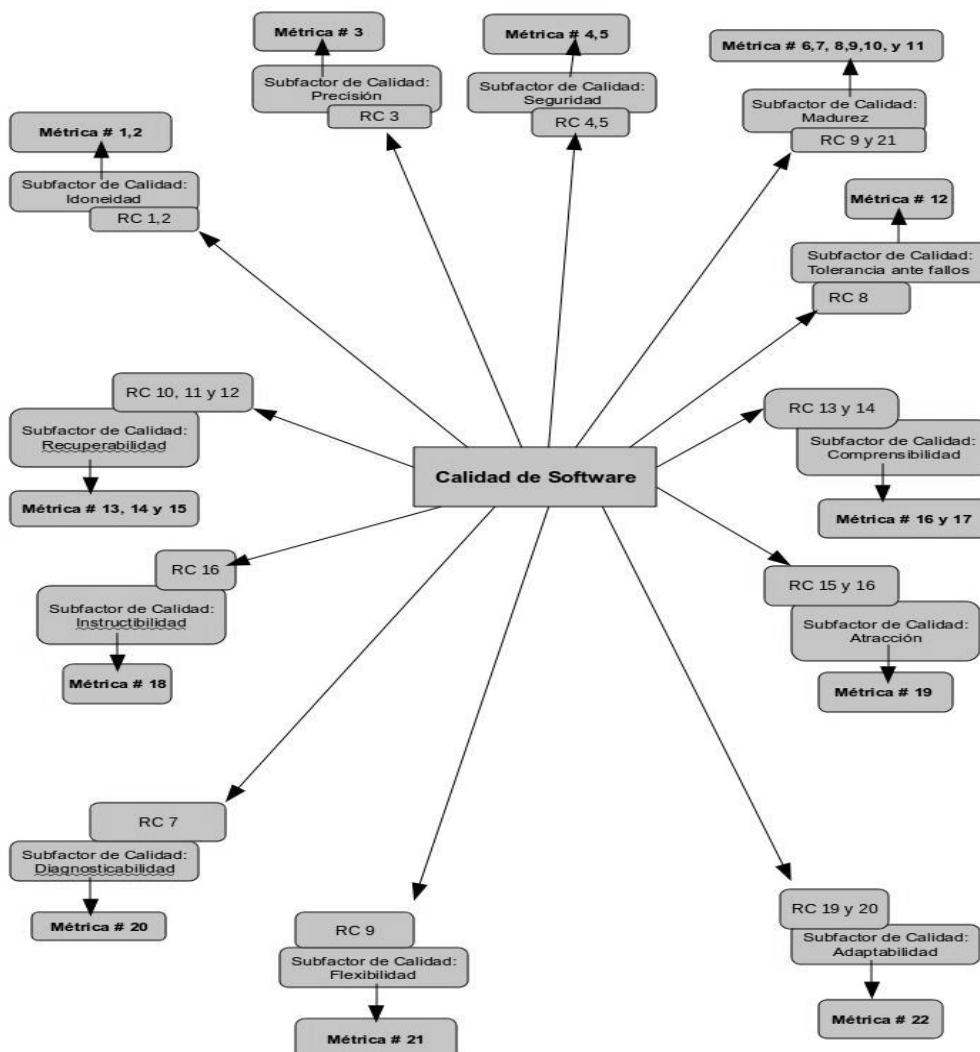
Una métrica externa de adaptación debe ser capaz de medir el comportamiento del sistema o el usuario que está tratando de adaptar el software a diferentes entornos especificados. Cuando un usuario tiene que aplicar un procedimiento de adaptación que no ha sido proporcionado por el sistema para una necesidad de adaptación específica, el esfuerzo necesario del usuario para la adaptación debe ser

medido.

Las métricas de adaptabilidad son:

- ✓ Métrica # 22: Capacidad de adaptación (Propone medir: ¿Cuántas condiciones especificadas en los requisitos no han satisfecho lo requerido?).

Cada una de estas métricas es un reflejo de los indicadores de calidad, y su definición está basada atendiendo a los requerimientos de calidad especificados. En la siguiente figura queda representado, la asociación de los requerimientos de calidad con los subindicadores de calidad y a la vez con las métricas que miden cada subindicador de calidad.



Fueron seleccionadas métricas externas de 5 de las 6 características de calidad (Funcionalidad, Confiabilidad, Usabilidad, Mantenibilidad y Portabilidad) que contiene la ISO/IEC 9126. No se definieron métricas para la característica Eficiencia debido a que esta investigación se centra en las pruebas manuales y este indicador es muy complejo de medir de esta manera. Actualmente en el Laboratorio Industrial de Pruebas de Software de la entidad rectora de calidad en la UCI, CALISOF, solo se realizan pruebas enfocadas a 5 de las 6 características de la ISO/IEC 9126, no se tiene en cuenta la Mantenibilidad. Atendiendo a las particularidades del producto de software que se analiza en esta investigación, se decide tener en cuenta la característica de Mantenibilidad, a la cual se le define una métrica.

A continuación se muestran las métricas, según el tipo de prueba a realizar:

Tipo de pruebas	Nombre de la métrica
Funcionales	Métrica # 1: Adecuación funcional.
	Métrica # 2: Completitud del desarrollo funcional.
	Métrica # 3: Exactitud esperada.
	Métrica # 6 : Densidad de fallos totales contra casos de prueba.
	Métrica # 7: Grado de solución ante fallos totales.
	Métrica # 8: Densidad de fallos.
	Métrica # 9: Erradicación de fallos.
	Métrica # 10: Cobertura de las pruebas.
	Métrica # 11: Madurez de las pruebas.
Seguridad	Métrica # 4: Prevención de la corrupción de los datos.
	Métrica # 5: Capacidad de control de acceso.
Recuperación y tolerancia a fallas	Métrica # 12: Evitación de operaciones incorrectas o fallos.
	Métrica # 13: Grado de disponibilidad.
	Métrica # 14: Tiempo medio de inactividad.
	Métrica # 15: Restaurabilidad.
Usabilidad	Métrica # 16: Accesibilidad a tutoriales.
	Métrica # 17: Comprensibilidad del Entorno de Escritorio.
	Métrica # 18: Eficiencia de la documentación del usuario y/o sistemas de ayuda en el uso.
	Métrica # 19: Adaptabilidad de la apariencia de la interfaz.

Configuración	Métrica # 21: Flexibilidad de la distribución.
---------------	--

Cada una de estas métricas puede ser calculada al terminar una iteración de las pruebas. Se observa el porcentaje de cumplimiento de las características de calidad en cada iteración, para comprobar si aumentan o disminuyen las mismas.

2.1.2 Cálculo de las métricas de calidad de software.

Para poder realizar el cálculo de las métricas, aplicando la fórmula correspondiente, se deben obtener los valores a utilizar. A continuación se muestra un ejemplo:

Nombre de la métrica	Variables	Datos
Adecuación funcional	A- Número de aplicaciones instaladas que no se adecuan a la función especificada en los requisitos. B- Número total de aplicaciones instaladas.	A=0 B=32

Cada métrica de calidad tiene una fórmula y una interpretación diferente de los resultados que arroja.

Sub-característica	Nombre de la métrica	Métrica	Nivel requerido	Valor
Idoneidad	Adecuación funcional	$X = 1 - A/B$	1 ($0 \leq X \leq 1$)	1

Métricas para la medición de la característica Funcionabilidad.

Métricas externas de la sub-característica de calidad Idoneidad.

Nombre de Métrica	Lo que se propone medir	Método de aplicación	Medición (Fórmula)	Interpretación de valor obtenido	Tipo de medida
Adecuación funcional	¿Cuán adecuadas son las aplicaciones instaladas?	Número de aplicaciones idóneas para ser ejecutadas en comparación con el número de aplicaciones a ser evaluadas.	$X = 1 - A/B$ A- Número de aplicaciones instaladas que no se adecuan a la función especificada en los requisitos. B- Número total de aplicaciones instaladas.	$0 \leq X \leq 1$ A mayor cercanía al 1 resulta más adecuada.	X= Contable/ Contable A = Contable B = Contable
Complejidad del desarrollo funcional	¿Cuán completa ha sido la construcción del sistema conforme con la especificación de requisitos?	Ejecutar las pruebas (de caja negra) funcionales de acuerdo con la especificación de requisitos. Cuenta el número de funciones perdidas detectadas y compare el resultado con el número de funciones descritas en la especificación	$X = 1 - A/B$ A- Número de problemas detectados o funcionalidades incumplidas de las aplicaciones instaladas detectadas en la evaluación. B- Número de	$0 \leq X \leq 1$ A mayor cercanía al 1 resulta mejor.	X= Contable/ Contable A = Contable B = Contable

Capítulo 2. Proceso de pruebas manuales para la distribución GNU/Linux Nova

			funcionalidades o especificaciones de requerimientos, de las aplicaciones evaluadas, descritas en la especificación de requisitos.		
--	--	--	--	--	--

Métrica de Precisión

Nombre de la Métrica	Lo que se propone medir	Método de aplicación	Medición (Fórmula)	Interpretación del valor obtenido	Tipo de medida
Exactitud esperada	¿Existen diferencias entre los resultados actuales y los razonablemente esperados?	Ejecutar los casos de pruebas de entrada versus salida y comparar los resultados actuales y los razonablemente esperados. Cuente el número de casos encontrados con diferencias inaceptables en relación con los resultados razonablemente esperados.	$X = A/T$ A- Número de casos encontrados con diferencias entre los resultados razonablemente esperados y aquellos resultantes más allá de lo permisible. T- Tiempo de operación.	$0 \leq X$ A mayor cercanía al 0 resulta mejor.	X= Contable/ Tiempo A= Contable T =Tiempo

Métrica de Seguridad

Nombre de la Métrica	Lo que se propone medir	Método de aplicación	Medición (Fórmula)	Interpretación del valor obtenido	Tipo de medida
Prevención de la corrupción de los datos	¿Cuál es la frecuencia de eventos de corrupción de los datos?	Contar las apariciones de eventos de corrupción de datos mayor y menor.	<p>a) $X=1-A/N$</p> <p>A- Número de veces que un evento de corrupción de los datos importantes se produce.</p> <p>N- Número de casos de prueba que provoca los eventos de corrupción de datos.</p> <p>b) $Y=1-B/N$</p> <p>B- Número de veces que un evento de corrupción de datos de menor importancia se produjo.</p> <p>c) $Z= A/T \text{ ó } B/T$</p> <p>T- Período de tiempo de funcionamiento.</p>	<p>a) $0 \leq X \leq 1$</p> <p>A mayor cercanía al 1 resultará mejor.</p> <p>b) $0 \leq Y \leq 1$</p> <p>A mayor cercanía al 1 resulta mejor.</p> <p>c) $0 \leq Z$</p> <p>A mayor cercanía al 1 resulta mejor.</p>	<p>X= Contable/Contable</p> <p>A=Contable</p> <p>B=Contable</p> <p>N=Contable</p> <p>Y= Contable/Contable</p> <p>T= Tiempo</p> <p>Z= Contable/Contable</p>

Capítulo 2. Proceso de pruebas manuales para la distribución GNU/Linux Nova

Capacidad de control de acceso	¿Cuál es la frecuencia de accesos ilegales al sistema?	Contar el número de operaciones ilegales permitidas en comparación con el número de operaciones ilegales intentadas.	$X=A/B$ A- Número de tipos de operaciones ilegales que están en las condiciones ocurridas. B- Número de diferentes tipos de operaciones ilegales intentadas.	$0 \leq X \leq 1$ A mayor cercanía al 1 resulta mejor, porque se ha identificado mayor cantidad de operaciones ilegales.	X= Contable/ Contable A=Contable B=Contable
--------------------------------	--	--	--	---	--

Métricas para la medición de la característica Confiabilidad.

Métricas de Madurez

Nombre de Métrica	Lo que se propone medir	Método de aplicación	Medición (Fórmula)	Interpretación de valor obtenido	Tipo de medida
Densidad de fallos totales contra casos de prueba	¿Cuántos fallos totales fueron detectados durante un período de pruebas definido?	Cuente el número de fallos totales detectados y el número de casos de pruebas.	$X= 1-A1 / A2$ A1- Número total de fallos totales detectados. A2- Número de casos de pruebas ejecutados.	$0 \leq X$ En dependencia del estadio de las pruebas. En etapas más avanzadas, mientras más pequeño, mejor.	X= Contable/ Contable A1= Contable A2= Contable
Grado de solución	¿Cuántas condiciones de	Cuente el número de fallos	$X= A1 / A2$	$0 \leq X \leq 1$	X= Contable/

Capítulo 2. Proceso de pruebas manuales para la distribución GNU/Linux Nova

ante fallos totales	fallo total están resueltas?	totales que no se repitieron en determinado período de pruebas bajo condiciones similares. Mantenga un reporte de solución de problemas describiendo la situación de todos los fallos totales.	A1- Número de fallos totales solucionados. A2- Número total de problemas reales detectados.	A mayor cercanía al 1 resulta mejor, cuantos más fallos totales estén resueltos.	Contable A1= Contable A2= Contable
Densidad de fallos	¿Cuántos fallos fueron detectados durante un período de pruebas definido?	Cuente el número de fallos detectados y la cantidad de casos de prueba ejecutados.	$X = A / B$ A- Número total de fallos detectados. B- Cantidad de Casos de Prueba ejecutados.	$0 \leq X$ Depende del estadio de pruebas. etapas más avanzadas, mientras más pequeño, mejor.	$X =$ Contable/ Tamaño A = Contable B = Tamaño más
Erradicación de fallos	¿Cuántos fallos han sido corregidos?	Cuente el número de fallos resueltos durante el período de pruebas y compárelo con el número total de fallos detectados y el número total de fallos pronosticados.	1) $X = A1 / A2$ A1- Número de fallos solucionados. A2- Número total de fallos reales detectados. 2) $Y = A1 / A3$	$0 \leq X \leq 1$ A mayor cercanía al 1 resulta mejor (Cuantos menos fallos queden). $0 \leq Y$	1) $X =$ Contable/ Contable A1= Contable A2= Contable 2) $Y =$ Contable/ Contable

Capítulo 2. Proceso de pruebas manuales para la distribución GNU/Linux Nova

			A3- Número total de fallos latentes pronosticados.	A mayor cercanía al 0 resulta mejor (cuantos menos fallos queden).	A3= Contable
Cobertura de las pruebas	¿Cuántos casos de pruebas requeridos han sido ejecutados durante las pruebas?	Cuente el número de casos de pruebas que han sido ejecutados durante las pruebas y compárelo con el número de casos de pruebas requeridos para obtener una adecuada cobertura de pruebas.	$X = A / B$ A- Número de casos de pruebas que han sido realmente ejecutados, y que representan el escenario de operación durante las pruebas. B- Número de casos de pruebas a ejecutar requeridos para cubrir los requisitos.	$0 \leq X \leq 1$ Mientras más cercano al 1, mejor cobertura.	X= Contable/ Contable A= Contable B= Contable
Madurez de las pruebas	¿Cuán eficientes son los casos de prueba del producto?	Cuente el número de casos de pruebas que han obtenido un resultado satisfactorio de los casos realmente ejecutados y compárelo con el número total de casos de pruebas requeridos para cubrir los	$X = 1 - A / B$ A- Cantidad de No Conformidades (NC) que fueron detectadas por utilizar los casos de pruebas. B- Número de casos de	$0 \leq X \leq 1$ A mayor cercanía al 1 resulta mejor.	X= Contable/ Contable A= Contable B= Contable

Capítulo 2. Proceso de pruebas manuales para la distribución GNU/Linux Nova

		requisitos.	pruebas a ejecutar para cubrir los requisitos.		
--	--	-------------	--	--	--

Métrica de Tolerancia ante fallos.

Nombre de Métrica	Lo que se propone medir	Método de aplicación	Medición (Fórmula)	Interpretación del valor obtenido	Tipo de medida
Evitación de operaciones incorrectas o fallos	¿Cuántas funciones están implementadas con capacidad de evitación de operaciones incorrectas o fallos?	Cuente el número de casos de prueba de operaciones incorrectas que fueron evitadas para que no causaran fallos totales críticos o serios, y compárelo con el número de casos de pruebas ejecutados a los patrones de operaciones incorrectas considerados en un período de pruebas bajo condiciones similares.	$X = A / B$ A- Número de ocurrencia de fallos totales críticos o serios evitada. B- Número de casos de pruebas ejecutados a los patrones de operaciones incorrectas (casos causantes de fallos) durante las pruebas.	$0 \leq X \leq 1$ A mayor cercanía al 1 resulta mejor, cuantas más operaciones incorrectas del usuario sean evitadas.	X= Contable/ Contable A1= Contable A2= Contable

Métricas de Recuperabilidad.

Nombre de Métrica	Lo que se propone medir	Método de aplicación	Medición (Fórmula)	Interpretación del valor obtenido	Tipo de medida
Grado de disponibilidad	¿Cuán disponible está el sistema para su uso según	Utilizar el sistema en un ambiente operacional	$X = A1 / A2$ A1- Total de casos de	$0 \leq X \leq 1$ A mayor cercanía	X= Contable/ Contable

Capítulo 2. Proceso de pruebas manuales para la distribución GNU/Linux Nova

	los requerimientos?	ejecutando todas las pruebas disponibles instrucciones del usuario.	al 1 resulta mejor. exitosamente, cuando se han intentado utilizar. A2- Total de casos de prueba que se han intentado utilizar durante el tiempo de observación.	A1= Contable A2= Contable	
Tiempo medio de inactividad	¿Cuál es el tiempo promedio en que el sistema se cae cada vez que el sistema se mantiene no disponible cuando ocurre un fallo total y antes de la arrancada gradual?	Mida el tiempo de inactividad durante un período de prueba especificado y compute el tiempo medio.	$X = T / N$ T- Tiempo total de inactividad. N- Número de desastres observados. El peor caso o la distribución del tiempo de inactividad deben ser medidos.	$0 < X$ Cuanto menor sea mejor, el sistema está inactivo por menos tiempo.	X= Tiempo/ Contable T= Tiempo N= Contable
Restaurabilidad	¿Cuán capaz es el producto de auto restaurarse luego de un evento anormal o una solicitud?	Cuente el número de restauraciones exitosas y compárelo con el número de restauraciones probadas requeridas por las especificaciones.	$X = A / B$ A- Número de casos de restauración exitosos. B- Número de casos de restauración probados por los requisitos.	$0 \leq X \leq 1$ A mayor cercanía al 1 mejor y el producto es más capaz de restaurarse en	X= Contable/ Contable A= Contable B= Contable

Capítulo 2. Proceso de pruebas manuales para la distribución GNU/Linux Nova

		Ejemplos de requisitos de restauración son: - función deshacer, - función rehacer,		casos definidos.	
--	--	--	--	------------------	--

Métricas para la medición de la característica Usabilidad.

Métricas de Comprensibilidad.

Nombre de Métrica	Lo que se propone medir	Método de aplicación	Medición (Fórmula)	Interpretación del valor obtenido	Tipo de medida
Accesibilidad de tutoriales	¿A qué proporción de demos/tutoriales pueden acceder los usuarios?	Conduzca las pruebas de usuario. Observe el comportamiento del usuario.	$X = A / B$ A- Número de demos/tutoriales a los que pueden acceder los usuarios exitosamente. B- Número total de demos/tutoriales a los que se puede acceder.	$0 < X \leq 1$ A mayor cercanía al 1 mejor.	X= Contable/ Contable A= Contable B= Contable
Comprensibilidad del EE.	¿Qué proporción de las funciones del producto el usuario nuevo podrá entender correctamente?	Llevar a cabo pruebas de usuario y entrevista con cuestionarios. Cuento el número de funciones de	$X = A/B$ A- Número de funciones de la interfaz, cuya finalidad es correctamente descrito por	$0 \leq X \leq 1$ A mayor cercanía al 1 resulta mejor.	X= Contable/ Contable A= Contable B= Contable

Capítulo 2. Proceso de pruebas manuales para la distribución GNU/Linux Nova

		interfaz de usuario cuando el usuario. el destino sea fácilmente comprensible para el usuario y compararlo con el número de funciones disponibles para el usuario.	B- Número de funciones disponibles en la interfaz.		
--	--	---	--	--	--

Métrica de Cognoscibilidad

Nombre de Métrica	Lo que se propone medir	Método de aplicación	Medición (Fórmula)	Interpretación del valor obtenido	Tipo de medida
Eficacia de la documentación del usuario y sistemas de ayuda en el uso	¿Qué proporción de las funciones pueden ser utilizadas correctamente después de leer la documentación o el uso de sistemas de ayuda?	Cuente el número de funciones que se usan correctamente después de leer la documentación o el uso de sistemas de ayuda y comparar con el número total de funciones.	$X = A/B$ A- Número de funciones que se pueden utilizar. B- Total de número de funciones que ofrece.	$0 \leq X \leq 1$ A mayor cercanía al 1 resulta mejor.	X= Contable/ Contable A= Contable B= Contable

Métrica de Atracción

Nombre de Métrica	Lo que se propone medir	Método de aplicación	Medición (Fórmula)	Interpretación del valor obtenido	Tipo de medida
Adaptabilidad de la apariencia de	¿Qué proporción de los elementos de la interfaz	Conduzca las pruebas de usabilidad. Observe el	$X = A / B$ A- Número de elementos	$0 \leq X \leq 1$ A mayor cercanía	X= Contable/ Contable

Capítulo 2. Proceso de pruebas manuales para la distribución GNU/Linux Nova

la interfaz	puede ser, por su apariencia, adaptado por el usuario para la satisfacción del mismo?	comportamiento del usuario.	de la interfaz del sistema cuya apariencia puede ser adaptada por el usuario. B- Número de elementos de la interfaz del sistema cuya apariencia quiere adaptar el usuario.	al 1 resulta mejor.	A= Contable A= Contable
-------------	---	-----------------------------	---	---------------------	----------------------------

Métricas para la medición de la característica Mantenibilidad.

Métricas de Diagnosticabilidad

Nombre de Métrica	Lo que se propone medir	Método de aplicación	Medición (Fórmula)	Interpretación del valor obtenido	Tipo de medida
Cumplimiento de las pruebas	¿Cuántas aplicaciones u opciones del sistema se han probado?	Cuenta el número de aplicaciones y opciones del sistema que no han sido probadas y compárelas con el número total de aplicaciones instaladas y opciones del sistema descritas en los requisitos.	$X=1-A/B$ A- Número de aplicaciones y opciones del sistema que no han sido probadas. B- Número total de aplicaciones y opciones del sistema descritas en los requisitos.	$0 \leq X \leq 1$ A mayor cercanía al 1 resulta mejor.	X= Contable/ Contable A= Contable B= Contable

Métricas de Flexibilidad

Nombre de la Métrica	Lo que se propone medir	Método de aplicación	Medición (Fórmula)	Interpretación del valor obtenido	Tipo de medida
Flexibilidad de la distribución.	¿Cuántas no conformidades encontradas en las pruebas no han sido resueltas?	Cuente el número de no conformidades encontradas en las pruebas y compárelas con el número de no conformidades no resueltas.	$X=1-A/B$ A- Número de no conformidades encontradas en las pruebas que no han sido resueltas. B- Número total de no conformidades.	$0 \leq X \leq 1$ A mayor cercanía al 1 resulta mejor.	X= Contable/ Contable A= Contable B= Contable

Métricas para la medición de la característica Portabilidad.

Métricas de Adaptabilidad

Nombre de la Métrica	Lo que se propone medir	Método de aplicación	Medición (Fórmula)	Interpretación del valor obtenido	Tipo de medida
Capacidad de adaptación.	¿Cuántas condiciones especificadas en los requisitos no han satisfecho lo requerido?	Cuente el número de condiciones especificadas en los requerimientos y compárelas con el número de condiciones especificadas que no son suficientes para satisfacer el	$X=1-A/B$ A- Número de condiciones especificadas en los requisitos que no se completan o no son suficientes para satisfacer el	$0 \leq X \leq 1$ A mayor cercanía al 1 resulta mejor.	X= Contable/ Contable A= Contable B= Contable

Capítulo 2. Proceso de pruebas manuales para la distribución GNU/Linux Nova

		satisfacen lo pedido.	nivel adecuado especificado. B- Número total de condiciones descritas en los requisitos.		
--	--	-----------------------	---	--	--

2.2 Proceso de pruebas actual de la distribución GNU/Linux Nova.

El proceso de pruebas de Nova comienza con la captura de los requerimientos del sistema, los cuales se necesitan para realizar los casos de pruebas que se van a utilizar. Además es necesario mantener una fluida comunicación entre los aseguradores de la calidad y los desarrolladores, logrando una correcta coordinación entre ellos con el objetivo de responder rápidamente y solucionar los errores de software (bugs) y no conformidades que puedan aparecer. Con el propósito crucial de encontrar la mayor cantidad de errores y no conformidades en las versiones Alpha y Beta, lo que permitiría corregirlos antes de realizar una liberación oficial de la distribución, incidiendo directamente en la calidad y el prestigio de Nova.

2.2.1 Diseño de las pruebas

En el diseño actual de las pruebas de la distribución GNU/Linux Nova [9] está reflejado en el caso de las pruebas manuales, que se diseña según la aplicación a la que van destinadas, que darán cumplimiento a los requisitos descritos que inicialmente se especificaron. Este diseño está basado en el método de caja negra, y en la misma se describe un flujo de eventos que debe seguir el probador así como los resultados esperados. No se comprueba la respuesta del sistema y el comportamiento de la distribución con flujos de eventos alternos, escogiendo caminos erróneos durante las pruebas.

La ejecución de las pruebas se realiza luego de obtenerse la primera versión Alpha, se señalan las no conformidades encontradas, cerrándose el primer ciclo de pruebas. Una vez corregidas las no conformidades se comienza otro ciclo de prueba y así sucesivamente hasta que no se encuentren errores o los errores encontrados sean permisibles. Cuando sean superadas estas pruebas, esta imagen Alpha se convierte en la versión Beta, a la cual se le realizan las pruebas correspondientes, además de ser publicada conjunto con los elementos necesarios para que la comunidad la pruebe y verifique si la misma está de acorde a sus necesidades. Al final de estos dos procesos iterativos se obtiene una versión Release que es la versión que se libera al público como oficial y estable (Figura 2.1).

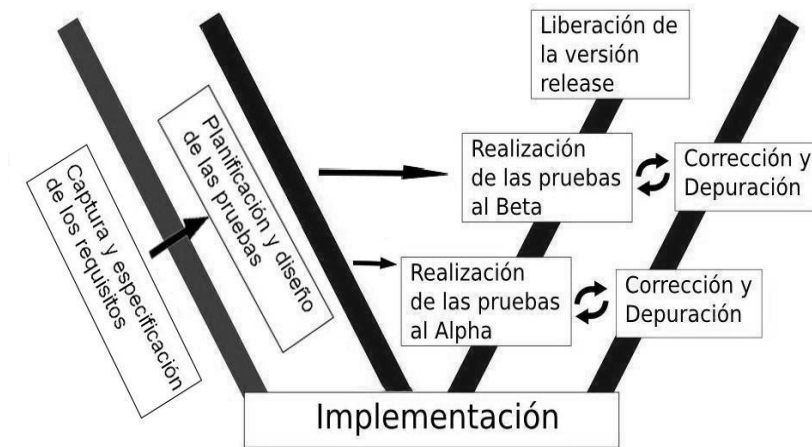


Figura 2.1 Proceso de Pruebas de la distribución GNU/Linux Nova.

2.2.2 Estrategia de pruebas

La estrategia de pruebas incluye las pruebas al sistema a la versión Alpha y las pruebas de aceptación a la Beta. En las pruebas al sistema están las pruebas al LiveCD, las de Instalación y las del Gestor de Paquetes. Mientras que en las pruebas de aceptación se encuentran las pruebas de actualización, las de uso diario y las de rendimiento (Figura 2.2). Pero como las pruebas no están diseñadas en función de indicadores de calidad que el sistema debe cumplir, sino en función de que no produzca errores, no se comprueba lo que significa el resultado obtenido de estas pruebas al concluir cada ciclo.

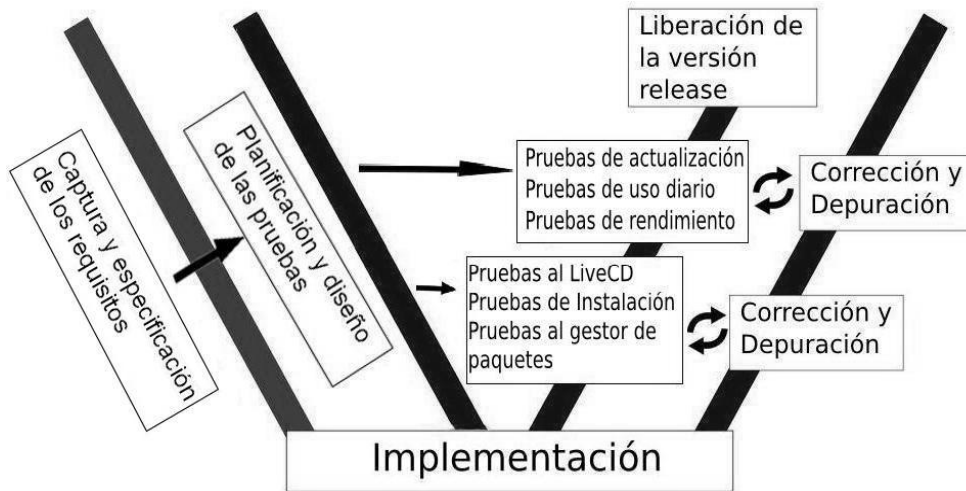


Figura 2.2 Estrategia de pruebas de la distribución GNU/Linux Nova.

2.3 Proceso de pruebas de la distribución GNU/Linux Nova evaluando los indicadores de calidad.

Después de definir y seleccionar las métricas de calidad atendiendo a los requisitos de calidad así como teniendo en cuenta las características de calidad según la ISO/IEC 9126-1, luego de analizado críticamente el proceso de pruebas actual de la distribución GNU/Linux Nova, se evalúa la presencia de los indicadores de calidad a través del cálculo de las métricas con los resultados de las pruebas obtenidos al concluir cada ciclo de pruebas.

Propósito: El propósito fundamental es encontrar la mayor cantidad de errores y no conformidades en las pruebas realizadas a las distintas versiones de la distribución GNU/Linux Nova, lo que permitiría corregirlos antes de liberar como estable y evaluar el grado de cumplimiento de los indicadores de calidad, lo cual daría una medida de la calidad o valor del producto, e incide directamente en la calidad y prestigio de Nova.

Alcance: Se considera que el alcance de este proceso de pruebas manuales es la mejora de la calidad de los productos liberados en Nova, mediante la realización de un proceso de pruebas manuales que evalúa el cumplimiento de los indicadores de calidad.

Roles involucrados:

Rol	Responsabilidad
Administrador de la calidad	Dirige y supervisa el proceso, definiendo las pautas a seguir en las pruebas al software, define el plan de pruebas. Es el encargado de dirigir la evaluación de los resultados de la ejecución de las pruebas y de transmitir esa información a los interesados.
Probador	Asesora en el diseño de los casos de pruebas basado en los requisitos. Ejecuta todos los casos de pruebas según los planes y cronogramas de cada producto. Reporta las no conformidades detectadas.
Gestor de las configuraciones	Controla los cambios de las versiones, evitando futuros errores.
Líder de proyecto	Solicita la realización de las pruebas, recibe la información de los resultados. Además es el encargado de asegurar que entre una iteración de pruebas y otra se corrija las no conformidades detectadas en el plazo acordado.
Analista de software	Diseña los casos de pruebas en función de los requisitos del producto.
Clientes o Usuarios ⁴	Define los parámetros de aceptación del producto y ejecuta de conjunto con los probadores las pruebas de aceptación ⁵ .

⁴En el caso de los productos, tales como las personalizaciones, que se desarrollan por encargo y tienen un cliente definido.

⁵En el caso de las pruebas de aceptación deben diseñarse con el cliente presente para que de su aprobación sobre la misma.

Entradas:

- Proyecto Técnico: describe de forma general las características del proyecto, cuáles son sus condiciones y los puntos de partida, así como la situación problemática que le dio inicio.
- Plan de Desarrollo de Software: describe como el proyecto interactúa con grupos externos y el ciclo de vida utilizado para el desarrollo del software, incluye toda la información necesaria para realizar la planificación del proyecto.
- Especificación de requisitos de software, en este documento se describe el sistema, y se especifican los requisitos funcionales a cumplir por la distribución, también se especifican los no funcionales teniendo en cuenta las diferentes categorías en que se pueden organizar los mismos.
- ISO, es el nombre con que se conocen las distribuciones GNU/Linux cuando son un producto terminado, es un fichero único listo para grabarse en un CD, DVD o dispositivo USB.
- Estándares de documentación:
 - Diseño de Casos de Pruebas basado en Requisitos (creado).
 - Plan de Pruebas (creado).
 - No Conformidades (creado).

Actividades:

- **Planificación de las Pruebas:**

Descripción: Esta es una actividad que tiene dos momentos fundamentales:

1.1 Durante la definición y planificación del producto se realiza una macro planificación de los momentos del desarrollo donde deben realizarse pruebas de software y si se realizan solo pruebas del sistema o si se incluyen pruebas de aceptación, se establecen las condiciones en que se ejecutarán las pruebas, los roles que intervienen y una propuesta de los recursos que podrían ser necesarios.

1.2 Después de la realización del diseño e implementación de los casos de pruebas, manuales, se define entonces un cronograma de las pruebas teniendo en cuenta la dimensión de estas y las características particulares de cada una.

Roles involucrados: Administrador de la calidad, Líder de proyecto.

Entradas:

- Proyecto Técnico.
- Plan de Desarrollo de Software.

Salidas:

▪ Plan de Pruebas, este es el artefacto donde se describe la planeación del proceso de pruebas, se caracteriza a los roles involucrados, el escenario de ejecución de las pruebas, los recursos necesarios, los requisitos a probar, la estrategia a seguir para la ejecución y como se evaluarán los resultados. Además no puede faltar el cronograma con cada actividad que se realiza en función de cumplir todo lo anterior y el responsable de hacerla en cada momento. [19]

- **Diseño de Casos de Pruebas:**

Descripción: El propósito de esta tarea es comprender los objetivos de la distribución GNU/Linux a desarrollar, para definir cuáles serán las pruebas a realizar, sus condiciones y pasos de ejecución. Los casos de pruebas se dividen en dos grandes grupos, según quien los ejecuta, las pruebas del sistema que las ejecutan los probadores y las de aceptación por el cliente. Estas últimas deben ser colegiadas y aceptadas por los clientes o usuarios hacia los que va dirigido el producto para garantizar que se satisfagan todos los posibles escenarios a probar y se tengan en cuenta en los resultados todas las alternativas de aceptación. Debe identificarse además los datos que serán necesarios para la realización de estas pruebas, que en el caso de las distribuciones un ejemplo de estos datos serían los ficheros de ejemplo para probar aplicaciones como reproductor de vídeo, música y visualización de imágenes.

Hay que tener presente que en el diseño de los casos de prueba se describe no solo el flujo principal de pasos para llevar a cabo, junto con la respuesta que debe dar el sistema, sino también al menos un flujo alternativo, escogiendo caminos erróneos y la respuesta que debe dar el sistema. El diseño de los casos de prueba debe realizarse teniendo en cuenta no solo los requisitos de la versión del sistema en análisis sino también teniendo en cuenta los requisitos de calidad especificados para la misma.

2.1 Definir indicadores de calidad según los requerimientos: atendiendo a los requerimientos del sistema de la personalización y a los requisitos de calidad de Nova, se definen los requisitos de calidad de la nueva versión los cuales son asociados los indicadores de calidad, para determinar los requeridos por el producto a probar.

2.2 Definir criterios de aceptación de cada requerimiento: definir los valores o rango de estos que se consideran como una respuesta aceptable al requisito a probar, se puede establecer guiado por las métricas para la evaluación de los indicadores determinados para el producto.

2.3 Diseñar casos de pruebas según los requerimientos: se basa en los requisitos del sistema y los de

calidad de la versión del producto a probar.

2.4 Identificar datos y recursos necesarios para la ejecución de cada caso de prueba: se puntualizan todos los datos y recursos necesarios para la ejecución de cada una de las pruebas.

Roles involucrados: analista de software, clientes o usuarios y administrador de la calidad.

Entradas:

- Proyecto Técnico.
- Especificación de requisitos de software.

Salidas:

▪ Diseño de Casos de Pruebas basado en Requisitos, en este documento se describen las condiciones de ejecución de las pruebas, se enumeran los requisitos a probar acompañados con las tablas donde se describen los escenarios de los casos de prueba, las variables a tener en cuenta en cada uno, la respuesta que debe dar el sistema, el flujo principal de pasos para llevarlo a cabo y en algunos casos al menos un flujo alternativo con una operación errónea. Además en un punto aparte se caracterizan todas las variables que intervienen en los distintos requisitos.

- **Ejecución de los Casos de Pruebas:**

Descripción: El propósito principal de esta tarea es verificar la calidad del producto, para ellos se ejecutan los casos de pruebas y se comunican los resultados al equipo de desarrollo. Es importante tener en cuenta en el momento que se realiza cada ciclo de pruebas pues de eso depende la batería de casos de prueba a ejecutar. La comunicación de los resultados al equipo debe hacerse en un lugar accesible a todos para que se solucionen con mayor prontitud las no conformidades y errores detectados. Los casos de prueba deben mantenerse actualizados en función de los cambios que pueden haber surgido durante el desarrollo, para que luego no queden reportes de no conformidades que no procedan.

3.1 Ejecutar pruebas de LiveCD.

3.2 Ejecutar pruebas de instalación.

3.3 Ejecutar pruebas de gestión de paquetes.

3.4 Ejecutar pruebas del sistema.

3.5 Ejecutar pruebas de actualización.

3.6 Ejecutar pruebas de aceptación.

3.7 Ejecutar pruebas de uso diario.

3.8 Publicación de los resultados de las pruebas (Esta subactividad se realiza al finalizar cada ciclo).

Roles involucrados: Probador, Administrador de la calidad, Clientes y Usuarios.

Entradas:

- Diseño de Casos de Pruebas basado en Requisitos.
- Plan de Pruebas.
- ISO.

Salidas:

▪ No Conformidades, este artefacto es el informe de todas las no conformidades detectadas y reportadas. En él se describen los aspectos generales de como se llevaron a cabo las pruebas y las dificultades detectadas en el diseño de las mismas. Se enumeran los elementos probados y en el caso de los no probados la razón por la cual no se probaron, por último se caracterizan cada una de las no conformidades encontradas, según la criticidad y con qué requerimiento está relacionada.

- **Evaluación de los Casos de Pruebas:**

Descripción: El propósito principal de esta tarea es verificar la calidad del producto, así como el cumplimiento de los indicadores de calidad y comunicarlo al equipo de desarrollo para que tome decisiones al respecto. Para ellos se analizan los resultados, se calcula el cumplimiento de los indicadores de calidad a través de las métricas, y comunican los resultados al equipo.

4.1 Evaluación de las pruebas: Esta se realiza al finalizar cada ciclo a través del cálculo de cada una de las métricas que corresponda en dependencia de la versión del sistema que se está probando.

4.2 Publicación de los resultados de la evaluación: Esta se realiza al finalizar cada evaluación, atendiendo a los ciclos de prueba.

Entradas:

- Diseño de Casos de Pruebas basado en Requisitos.
- Plan de Pruebas.
- Especificación de requisitos de software.
- No Conformidades.

Salidas:

- Informe de evaluación (Ver Anexo 2).

Condiciones:

Para llevar a cabo el proceso debe existir:

- Información detallada de los requerimientos del producto.
- Un canal de comunicación fluido, permanente y amplio entre el Líder del proyecto que desarrolla el producto, los analistas de software, los administradores de la calidad, los probadores y los clientes o usuarios.
- Un producto que esté en estado mínimamente funcional.
- Disposición del equipo de desarrollo para solucionar las no conformidades detectadas en el menor plazo posible.

Salidas:

- Diseño de Casos de Pruebas basado en Requisitos.
- Plan de Pruebas.
- No Conformidades.

A partir de la definición de las métricas externas para el proceso de pruebas de la distribución GNU/Linux Nova se pudo constatar la existencia de pruebas que miden los indicadores. Sin embargo no se realizaba una evaluación de esas pruebas, obteniendo la cantidad de errores y no el grado de calidad de la distribución. El proceso obtenido propone tener en cuenta los requisitos de calidad desde el diseño de las pruebas con el objetivo de crear pruebas más abarcadoras. Además incluye una nueva actividad de evaluación de las pruebas donde se aplica el cálculo de las métricas externas definidas permitiendo obtener el porcentaje de cumplimiento de los indicadores de calidad de la distribución GNU/Linux Nova.

Capítulo 3: “Evaluación del proceso de pruebas manuales propuesto.”

En este capítulo se abordan temas referentes a la evaluación de la propuesta realizada en el capítulo anterior, los resultados obtenidos y los análisis realizados sobre estos resultados permiten verificar la calidad de la propuesta. Para llevar a cabo la evaluación se presentan los resultados cuantitativos obtenidos de las pruebas realizadas a una personalización de la distribución de software libre GNU/Linux Nova escogida, logrando medir la efectividad de la propuesta a partir de los resultados obtenidos.

3.1 Método de Evaluación.

Para evaluar la propuesta se tiene en cuenta el cumplimiento de las características de calidad de la ISO/IEC 9126 y se procede a aplicar el método de evaluación experimental. El cual es un proceso que muestra a grandes rasgos qué tan factible es el proceso de pruebas pues los resultados que se obtienen comprueban todos los beneficios y las oportunidades que brinda la propuesta desarrollada. Para llevar a cabo este método se utiliza una serie de variables o procesos que constituyen la solución o una parte de la solución que se prueba y de cada una de estas variables se seleccionan un conjunto de atributos que representan requisitos o criterios para la evaluación.

3.2 Aplicación de la propuesta.

Se define un caso de estudio para evaluar la propuesta planteada en la presente investigación.

3.2.1 Definición del caso de estudio

Se selecciona la personalización NovaMedia que se está desarrollando en el proyecto FreeViUX (*Free Videos Using GNU/Linux*) del centro CESOL. La misión principal del grupo de proyecto FreeViUX es la creación de audiovisuales utilizando software libre y al existir una amplia gama de herramientas libres en el diseño gráfico y la animación, se hace evidente la necesidad de separar las herramientas más significativas en el tratamiento de imágenes, trabajo con imágenes vectoriales, edición de sonido y video, y en la modelación y animación en dos y tres dimensiones.

NovaMedia se está desarrollando con el objetivo de construir una personalización de Nova enfocada al diseño gráfico y a la animación. Esta edición brinda una distribución GNU/Linux que consume pocos recursos y posee un entorno de escritorio ligero donde el rendimiento no afecta la usabilidad. Sus requisitos son básicamente los siguientes:

Capítulo 3. Evaluación del proceso de pruebas manuales propuesto

Requisitos Funcionales:

1. Permitir instalar la personalización.
2. Permitir la manipulación de imágenes en forma de mapa de bits en diferentes formatos y que estas se puedan guardar en diferentes extensiones para su utilización.
3. Crear nuevos gráficos vectoriales y aprobar guardar en diferentes formatos.
4. Permitir al usuario realizar animaciones en dos dimensiones y acceder guardar en diferentes formatos.
5. Permitir al usuario realizar animaciones en tres dimensiones y admitir guardar en diferentes formatos.
6. Crear y editar sonido avanzado y permitir guardarlo en diferentes formatos.
7. Crear y editar video avanzado y permitir guardarlo en diferentes formatos.
8. Permitir crear música y poder guardarla en diferentes formatos.
9. Permitir la reproducción de música y videos con diferentes formatos abiertos desde archivos o URL, y contar con opciones que mejoren su visualización.
10. Permitir al usuario la posibilidad de conectarse a la web.
11. Permitir realizar operaciones comunes como editar texto, hojas de cálculos y presentaciones.
12. Permitir mediante un cliente de mensajería instantánea la comunicación utilizando los protocolos XMPP, IRP y Yahoo.
13. Permitir grabar discos.
14. Permitir al usuario escoger si desea salir de la sesión: apagando, reiniciando, suspendiendo, o hibernando.
15. Permitir al usuario bloquear la pantalla.
16. Permitir al usuario gestionar ficheros.
17. Permitir al usuario enviar ficheros a la papelera de reciclaje.

Requisitos No Funcionales:

1. El entorno de escritorio que se utilice debe contar con las siguientes condiciones: simple de usar, interactivo y profesional.
2. El entorno de escritorio deberá funcionar sobre PC con pocas prestaciones.

Capítulo 3. Evaluación del proceso de pruebas manuales propuesto

3.2.2 Ejecución de la propuesta

A partir de los requisitos de calidad de Nova y atendiendo a los requerimientos del sistema, se determinan los indicadores de calidad principales requeridos por NovaMedia.

Característica	Sub-característica	Métrica
Funcionalidad	Idoneidad	Adecuación funcional.
Funcionalidad	Idoneidad	Complejidad del desarrollo funcional.
Funcionalidad	Exactitud	Exactitud esperada.
Funcionalidad	Seguridad	Prevención de la corrupción de los datos.
Funcionalidad	Seguridad	Capacidad de Control de Acceso.
Confiabilidad	Madurez	Densidad de fallos totales contra casos de prueba.
Confiabilidad	Madurez	Grado de solución ante fallos totales.
Confiabilidad	Madurez	Densidad de fallos.
Confiabilidad	Madurez	Erradicación de fallos.
Confiabilidad	Madurez	Cobertura de las pruebas.
Confiabilidad	Madurez	Madurez de las pruebas.
Confiabilidad	Recuperabilidad	Grado de disponibilidad.
Confiabilidad	Recuperabilidad	Tiempo medio de inactividad.
Mantenibilidad	Diagnosticabilidad	Cumplimiento de las pruebas.
Mantenibilidad	Flexibilidad	Flexibilidad de la distribución.

A partir de estos se definieron 5 Casos de Prueba con 12 escenarios, en función de los requerimientos principales y teniendo en cuenta los indicadores de calidad presentes en los requisitos.

No se realiza una planificación genérica pues se está llevando a cabo un proceso experimental no sincronizado desde un inicio con el proyecto de desarrollo de NovaMedia. Luego de diseñados los casos de prueba, se elabora un cronograma detallado de quién ejecuta que caso de prueba en que momento, y los recursos necesarios para ello. La tabla que se muestra a continuación es una representación del cronograma, las iteraciones de pruebas se separan por una fila vacía.

No.	Tarea	Fecha	Responsable	Participante	Observaciones
-----	-------	-------	-------------	--------------	---------------

Capítulo 3. Evaluación del proceso de pruebas manuales propuesto

1	Ejecutar casos de prueba del 1 al 5	13/05/2012	Rosa María López Rangel	Rosa María López Rangel	Se necesita una PC con Novamedia
2	Evaluar el resultado obtenido en los casos de prueba	14/05/2012	Rosa María López Rangel	Rosa María López Rangel	
3	Ejecutar casos de prueba del 1 al 5	16/05/2012	Rosa María López Rangel	Rosa María López Rangel	Se necesita una PC con Novamedia
4	Evaluar el resultado obtenido en los casos de prueba	17/05/2012	Rosa María López Rangel	Rosa María López Rangel	
5	Ejecutar casos de prueba del 1 al 5	20/05/2012	Rosa María López Rangel	Rosa María López Rangel	Se necesita una PC con Novamedia
6	Evaluar el resultado obtenido en los casos de prueba	21/05/2012	Rosa María López Rangel	Rosa María López Rangel	

Una vez realizadas las pruebas a la personalización escogida de la distribución, se trabaja con los resultados obtenidos. Se calcularon las métricas de calidad atendiendo a su fórmula y a la interpretación de los resultados que cada una ofrece. La forma de agrupar los datos para el cálculo de las métricas está reflejada en la siguiente tabla:

Versión de Nova	Artefacto	Iteración	Sub-característica	Métrica	Nivel requerido	Valor
NovaMedia	Aplicación	1	Idoneidad (Adecuación funcional)	$X = 1 - A/B$	1 ($0 \leq X \leq 1$)	0.2

Capítulo 3. Evaluación del proceso de pruebas manuales propuesto

Resultados de las métricas.

A continuación se muestran los resultados obtenidos de cada una de las métricas para cada iteración de pruebas de la distribución NovaMedia. Los porcentos representan el nivel de cumplimiento de la subcaracterística de calidad que cada métrica calculada representa.

Funcionalidad:

Sub-característica	Métrica	1ra iteración	2da iteración	3ra iteración
Idoneidad	Adecuación funcional	100,00%	100,00%	100,00%
Idoneidad	Compleitud del desarrollo funcional	85,00%	90,00%	97,80%
Exactitud	Exactitud esperada	98,00%	100,00%	100,00%
Seguridad	Capacidad de Control de Acceso	100,00%	100,00%	100,00%

Confiabilidad:

Sub-característica	Métrica	1ra iteración	2da iteración	3ra iteración
Madurez	Densidad de fallos totales contra casos de prueba	90,00%	100,00%	100,00%
Madurez	Grado de solución ante fallos totales	---	100,00%	100,00%
Madurez	Densidad de fallos	85,00%	90,00%	95,00%
Madurez	Erradicación de fallos	---	98,00%	100,00%
Madurez	Cobertura de las pruebas	100,00%	100,00%	100,00%
Madurez	Madurez de las pruebas	98,00%	100,00%	100,00%
Recuperabilidad	Grado de disponibilidad	100,00%	100,00%	100,00%
Recuperabilidad	Tiempo medio de inactividad	98,00%	100,00%	100,00%

Mantenibilidad:

Capítulo 3. Evaluación del proceso de pruebas manuales propuesto

Sub-característica	Métrica	1ra iteración	2da iteración	3ra iteración
Diagnosticabilidad	Cumplimiento de las pruebas	100,00%	100,00%	100,00%
Flexibilidad	Flexibilidad de la distribución.	98,00%	98,50%	100,00%

3.2.3 Análisis de los resultados

Una vez obtenidas las métricas de calidad se hace necesario comparar los resultados de las métricas obtenidas en cada iteración para ver el cumplimiento de las subcaracterísticas de calidad en el producto liberado. El porcentaje de cumplimiento de cada subcaracterística debe ir aumentando a medida que pasan las iteraciones lo que representa que las pruebas están cumpliendo su objetivo principal, ir incrementando y mejorando la calidad del producto. En la presente tabla se muestra el cumplimiento de los indicadores de calidad en las distintas iteraciones.

Características de calidad	Iteración 1	Iteración 2	Iteración 3
Funcionalidad	95,75%	97,50%	99,45%
Confiabilidad	95,10%	98,50%	99,37%
Mantenibilidad	99,00%	99,25%	100,00%

Con el cálculo de las métricas expuestas se logra obtener la medida en que son cumplidas las características de calidad que exige la NC-ISO/IEC 9126 y de esta forma evaluar los productos liberados en Nova, obteniendo medidas de la efectividad en las pruebas manuales de la distribución GNU/Linux Nova.

3.3 Evaluación de la propuesta

A partir del análisis de la problemática existente antes de que se pusiera en práctica el nuevo proceso de pruebas y después de elaborado este proceso de pruebas se puede conocer cuán factible es el mismo. Además para realizar la evaluación se determina realizar una estimación en niveles del 1 al 5 dónde (1 es mal, 2 más o menos regular, 3 regular, 4 bien y 5 excelente). Donde se evalúan cómo se han comportado los criterios de evaluación de la propuesta antes y después del proceso de pruebas y así conocer cuán beneficioso es.

El caso de estudio en cuestión está dado por una insuficiente evaluación de la calidad del producto debido a que no se tienen en cuenta indicadores que midan la calidad de este tipo de productos. Por tal motivo este proceso de pruebas tiene como propósito liberar el producto con la mayor calidad posible partiendo de una evaluación cuantitativa a través de métricas de calidad.

La variable seleccionada es la aplicación de los indicadores de calidad dentro del proceso de pruebas. El comportamiento de esta se ve dentro del proceso de pruebas en dos actividades fundamentalmente: Diseño de Casos de Pruebas y Evaluación de las Pruebas. En el diseño de los casos prueba a través del análisis de los requisitos de calidad y en la evaluación de las pruebas con la aplicación de métricas de calidad en el análisis de los resultados obtenidos. A continuación se muestran cómo se evidencia este criterio antes y después de las modificaciones al proceso de pruebas.

Antes

El proceso de pruebas de la distribución GNU/Linux Nova en su versión anterior realizaba el diseño de las pruebas solamente teniendo en cuenta los requerimientos analizados por los desarrolladores. Este proceso tras la ejecución de las pruebas solamente hacía un análisis cualitativo de los resultados obtenidos.

Indicadores de Calidad	Valor
Análisis de requisitos de calidad	2
Aplicación de métricas de calidad	1

Después

El proceso de pruebas de la distribución GNU/Linux Nova con las modificaciones propuestas plantea realizar el diseño de las pruebas teniendo en cuenta además de los requerimientos del sistema, los requisitos de calidad para ese producto. Este proceso incluye tras la ejecución de las pruebas una

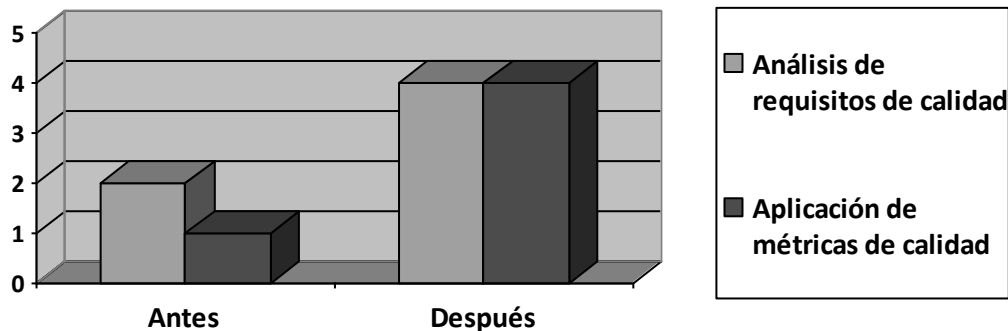
Capítulo 3. Evaluación del proceso de pruebas manuales propuesto

actividad para la evaluación de las pruebas, donde recomienda hacer conjuntamente con el análisis cualitativo de los resultados obtenidos una evaluación cuantitativa aplicando métricas de calidad.

Indicadores de Calidad	Valor
Análisis de requisitos de calidad	4
Aplicación de métricas de calidad	4

El cambio efectuado en el resultado de las pruebas se evidencia a continuación:

Aplicación de los indicadores de calidad después de la propuesta



La evaluación de la propuesta fue realizada a través de la ejecución de un proceso de pruebas manuales a la personalización NovaMedia que se encontraba en proceso de liberación y poseía la documentación necesaria para el diseño de los casos de prueba y el cálculo de las métricas. Se realiza el cálculo de cada métrica para determinar dado el resultado alcanzado en cada ciclo de pruebas, el nivel de cumplimiento de cada subcaracterística, obteniéndose resultados satisfactorios.

Con la presente investigación se obtiene una evidente mejora en la evaluación de la calidad de software de la distribución GNU/Linux Nova. Comprobándose de esta forma que las modificaciones realizadas al proceso de pruebas de la distribución GNU/Linux puede mejorar la calidad con que libera los productos desarrollados, lo cual contribuye a obtener mejor aceptación de la distribución por parte del público.

Conclusiones

- Teniendo en cuenta las características de la distribución GNU/Linux y tomando como guía las características de calidad del estándar NC-ISO/IEC 9126, se definieron los requisitos de calidad con los que deben cumplir los productos resultantes del proyecto y seleccionaron los indicadores de calidad requeridos.
- A partir de la definición de métricas externas que evalúan el comportamiento de estos indicadores de calidad se aplicaron los mismos en el proceso de pruebas manuales de la distribución cubana de software libre GNU/Linux Nova, lográndose evaluar cuantitativamente la calidad a través de la comprobación del cumplimiento de estos indicadores de calidad.
- Se analizó el comportamiento del proceso con la aplicación de los indicadores y las métricas de calidad, a través de la ejecución de un ciclo de pruebas de tres iteración a una personalización de la distribución GNU/Linux Nova y con los resultados obtenidos se pudo constatar que se puede lograr mejorar la calidad de los productos de la distribución GNU/Linux Nova, ganando en tiempo y recursos empleados, antes de ser entregada a CALISOFT para su liberación oficial.

Recomendaciones

Para el seguimiento de la presente investigación se pueden realizar algunas recomendaciones a partir de la experiencia percibida al aplicar la propuesta a la personalización de la distribución GNU/Linux Nova, las lecciones aprendidas y el criterio de los involucrados en la investigación.

- Continuar refinando los requisitos de calidad de la distribución GNU/Linux Nova definidos, a través de talleres con el equipo de desarrollo y encuestas a la comunidad de usuarios.
- Refinar las métricas de las subcaracterísticas de calidad para evaluar en las pruebas de la distribución y los productos que se obtengan en Nova para tener más elementos en la evaluación del software según el cumplimiento de las características de calidad de la NC-ISO/IEC 9126.
- Aplicar al proceso de pruebas de la distribución GNU/Linux Nova las métricas de calidad interna para aumentar la calidad del producto final.
- Incluir en el proceso de Gestión de Requisitos los indicadores de calidad a tener presente atendiendo a la NC-ISO/IEC 9126 para incrementar la especificación de estos en cuanto a los indicadores a medir posteriormente.

Bibliografía

Referencias Bibliográficas:

- [1] Tanenbaum, Andrew. *Sistemas Operativos Diseño e Implementación*-2da edición-1998
- [2] Wikipedia. Industria del Software, Disponible en: http://es.wikipedia.org/wiki/Industria_del_software, Consultado marzo 7 2012.
- [3] javaMexico. ¿Cuál es el sistema operativo que usas principalmente para trabajar? Disponible en: http://www.javamexico.org/poll/cual_es_el_sistema_operativo_que_usas_principalmente_para_trabajar, Consultado marzo 7 2012.
- [4] GNU. Visión general del sistema GNU. Disponible en: <http://www.gnu.org/gnu/gnu-history.es.html>, Consultado marzo 7 2012.
- [5] GNU. Disponible en <http://www.gnu.org/>, Consultado marzo 7 2012.
- [6] Wikipedia. Distribución de Software. Disponible en: http://es.wikipedia.org/wiki/Distribución_de_software, Consultado marzo 7 2012.
- [7] S Pressman, Roger. *Ingeniería del Software. Un enfoque práctico*. 5Ta Edición. 2002.
- [8] Pressman, Roger S. *Software Engineering. A Practitioner's Approach*. 7ma. Ed. McGraw-Hill, 2010. 870p. ISBN: 978-0-07-337 597 -7.
- [9] Monteagudo Suarez, Lorena. *Proceso de pruebas de la distribución GNU/Linux Nova*. Universidad de las Ciencias Informáticas, 2011.
- [10] Debian. Disponible en: <http://www.debian.org/>, Consultado 7 marzo 2012
- [11] Ubuntu. Disponible en: <http://www.ubuntu.com/>, Consultado 7 marzo 2012
- [12] Oecd-ilibrary.org. Estándares de calidad para la evaluación del desarrollo. Disponible en: http://www.oecd-ilibrary.org/development/estandares-de-calidad-para-la-evaluacion-del-desarrollo_9789264094949-es , Consultado 7 marzo 2012
- [13] García Romero, C.I, 2001. *El Modelo de Capacidad de Madurez y su Aplicación en Empresas Mexicanas de Software*. Tesis Licenciatura. Ingeniería en Sistemas Computacionales. Departamento de Ingeniería en Sistemas Computacionales, Escuela de Ingeniería, Universidad de las Américas-Puebla. Mayo. Disponible en: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/garcia_r_ci/capitulo2.pdf,

Consultado 7 marzo 2012.

[14] AWARDSPACE. Los Estándares de Calidad ISO para Desarrollo de Software. El Estándar de Calidad ISO 9001. Disponible en: http://tecnomaestros.awardspace.com/estandares_iso.php, Consultado 7 marzo 2012.

[15] Wikipedia. ISO/IEC 9126. Disponible en: http://es.wikipedia.org/wiki/ISO/IEC_9126, Consultado 7 marzo 2012

[16] Scribd Inc. Antología: Calidad en el Desarrollo de Software. Universidad Tecnológica de Puebla. 2011. Disponible en: <http://es.scribd.com/torito28/d/78595440/8-I-ISO-IEC-14598-Parte-1-Vision-General>, Consultado 7 marzo 2012.

[17] ISO. ISO/IEC 12119: 1994. Disponible en: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=1308, Consultado 7 marzo 2012

[18] ISO. ISO/IEC 15939: 2007. Disponible en: http://www.iso.org/iso/catalogue_detail.htm?csnumber=44344, Consultado 7 marzo 2012.

[19] Albo Castro, Mónica. Proceso de pruebas (v3) de la distribución GNU/Linux Nova. UCIENCIA 2012. Universidad de las Ciencias Informáticas, 2012.

Bibliografía Consultada:

1. Senn A, James. Análisis y diseño de sistemas de información.
2. Kendall. Análisis y diseño de sistemas.
3. S Presmman, Roger. Ingeniería de Software.
4. Burch G, Jonh. Diseño de sistemas de información Teórica y Práctica.
5. García Romero. El Modelo de Capacidad de Madurez y su Aplicación en Empresas Mexicanas de Software.
6. ISO/IEC. ISO/IEC 91261.2. Parte 1: Modelo de Calidad. 2005.
7. Capote, Tayché. Conceptualización e implementación de un Laboratorio Industrial de Pruebas de Software, Junio, 2011.
8. Góngora, Asnier. Catálogo automatizado de métricas de calidad para evaluar los productos de las pruebas, Diciembre, 2011.

Anexos

Anexo 1: Casos De Pruebas

Manipulación de imágenes (Gimp)

Escenario	Descripción	Variables	Respuesta del sistema	Flujo Central			
Gimp 1	Crear una imagen con el Gimp.		Crear y Guardar correctamente el archivo.	<ol style="list-style-type: none"> Haga clic en el menú principal -> Gráficos ->Editor de imágenes Gimp. Ir al menú Archivo ->Nuevo. Seleccione la primera plantilla y presione aceptar. Escoger el pincel y pintar su nombre. Seleccionar el menú Archivo ->Guardar. Escoger el directorio donde va a ser guardado el archivo así como el nombre y tipo de imagen y presionar guarda. 			
Gimp 2	Modificar una imagen con el Gimp.	<table border="1"> <tr> <td>imagen</td> </tr> <tr> <td>.jpg</td> </tr> <tr> <td>.png</td> </tr> </table>	imagen	.jpg	.png	Guardar correctamente la modificación hecha al archivo seleccionado.	<ol style="list-style-type: none"> Haga clic en el menú principal ->Gráficos ->Editor de imágenes Gimp. Ir al Menú Archivo ->Abrir. Escoger la imagen que desea modificar. Escoger el pincel y pintar su nombre. Seleccionar el menú Archivo ->Guardar. Modificar el nombre de la imagen y presionar guardar.
imagen							
.jpg							
.png							
Escenario	Descripción	Variables	Respuesta del sistema	Flujo Alterno			
Gimp 2	Modificar una imagen con el Gimp.		Mostrar un mensaje de error, diciendo que no pudo abrir el archivo especificado.	<ol style="list-style-type: none"> Haga clic en el menú principal ->Gráficos ->Editor de imágenes Gimp. Ir al Menú Archivo ->Abrir. Escoger un archivo con formato .doc Seleccionar ->Abrir. 			

Manipulación de sonido (Audacity)

Escenario	Descripción	Variable	Respuesta del sistema	Flujo Central
Audacity 1	Abrir un archivo.	audio	Abrir correctamente el archivo.	<ol style="list-style-type: none"> Haga clic en el menú principal -> Más Programas -> Sonido y video -> Audacity. Ir al Menú Archivo -> Abrir.
		.mp3		
		.wma		
Audacity 2	Modificar un archivo y guardar proyecto.	.mp3	Guardar correctamente la modificación hecha al archivo seleccionado.	<ol style="list-style-type: none"> Haga clic en el menú principal -> Más Programas -> Sonido y video -> Audacity. Ir al Menú Archivo -> Abrir. Escoger el archivo. Realizar modificación. Seleccionar el menú Archivo -> Guardar Proyecto. Modificar el nombre del archivo y presionar guardar.
		.wma		
Audacity 3	Modificar un archivo y exportar.	.mp3	Guardar correctamente la modificación hecha al archivo seleccionado.	<ol style="list-style-type: none"> Haga clic en el menú principal -> Más Programas -> Sonido y video -> Audacity. Ir al Menú Archivo -> Abrir. Escoger el archivo. Realizar modificación. Seleccionar el menú Archivo -> Exportar. Modificar el nombre del archivo y presionar guardar.
		.wma		

Escenario	Descripción	Variables	Respuesta del sistema	Flujo Alternativo
<i>Audacity 1</i>	<i>Modificar una imagen con el Gimp.</i>		<i>Mostrar un mensaje de error, diciendo que no pudo abrir el archivo especificado.</i>	<ol style="list-style-type: none"><i>1. Haga clic en el menú principal -> Más Programas -> Sonido y video -> Audacity.</i><i>2. Ir al Menú Archivo -> Abrir.</i><i>3. Seleccionar un archivo con formato .doc.</i><i>4. Seleccionar Abrir.</i>

Manipulación de video (Kdenlive)

Escenario	Descripción	Variable	Respuesta del sistema	Flujo Central
Kdenlive 1	Abrir un archivo.		Abrir correctamente el archivo.	<ol style="list-style-type: none"> Haga clic en el menú principal ->Más Programas ->Sonido y video ->Kdenlive. Ir al Menú Proyecto ->Abrir clip. Seleccione el clip y presione abrir.
Kdenlive 2	Guardar un archivo.		Guardar correctamente el archivo seleccionado.	<ol style="list-style-type: none"> Haga clic en el menú principal ->Más Programas ->Sonido y video ->Kdenlive. Ir al Menú Proyecto ->Abrir clip. Seleccione el clip y presione abrir. Ir al Menú File ->Save as, modificar el nombre del archivo y presionar Save.
Kdenlive 3	Procesar un archivo		Procesar correctamente el archivo.	<ol style="list-style-type: none"> Haga clic en el menú principal ->Más Programas ->Sonido y video ->Kdenlive. Ir al Menú Proyecto ->Abrir clip. Seleccione el clip y presione abrir. Ir a la Opción Procesar ->Procesar a rchivo. Modificar las propiedades del a rchivo y presionar procesar.

Animación (Blender)

Escenario	Descripción	Variable	Respuesta del sistema	Flujo Central
Blender	Crear una imagen en 3D.		Crear y Guardar correctamente el archivo.	<ol style="list-style-type: none"> 1. Haga clic en el menú principal ->Gráficos ->Blender. 2. Crear figura. 3. Ir al Menú Render ->Render Image. 4. Ir a la Opción Image ->Save as Image. 5. Escoger el directorio donde va a ser guardado el archivo así como el nombre y tipo de imagen y presionar guardar.

Control de Acceso

Escenario	Descripción	Variable	Respuesta del sistema	Flujo Central
Instalar programa	<i>Intentar acceder con una cuenta de tipo Usuario del escritorio</i>		<i>Mostrar en la misma ventana donde se intentó la autenticación el mensaje: Authentication Failure.</i>	<ol style="list-style-type: none"> 1. Haga clic en el menú principal -> Instalar software. 2. Buscar el software que se desea instalar. 3. Marcar la selección deseada. 4. Haga clic en Aplicar 5. Haga clic en continuar para añadir paquetes requeridos. 6. Escribir contraseña de Usuario del escritorio. 7. Haga clic en Authenticate.
Modificar cuenta	<i>Intentar modificar una cuenta, autenticándose con una cuenta de tipo Usuario del escritorio</i>		<i>Mostrar en la misma ventana donde se intentó la autenticación el mensaje: Authentication Failure.</i>	<ol style="list-style-type: none"> 1. Haga clic en el menú principal -> Más programas ->Preferencias ->Usuarios y grupos. 2. Escoger la cuenta a modificar. 3. Haga clic en el campo de esa cuenta que desee modificar. 4. Escribir contraseña de Usuario del escritorio. Haga clic en Authenticate.
Borrar información de otro usuario.	<i>Intentar eliminar información con una cuenta de tipo Usuario del escritorio</i>		<i>Mostrar una ventana que diga: permission denied.</i>	<ol style="list-style-type: none"> 1. Haga clic en el menú principal ->Mis documentos 2. Retroceda hasta la carpeta personal del usuario novamedia: /home/novamedia

			<p>3. Haga clic en la carpeta Escritorio. 4. Borre el archivo Guía Cubana de migración.</p>
--	--	--	---

Anexo 2: Informe de evaluación

Cumplimiento de los indicadores de calidad:

Indicador de calidad	Métrica	Porcentaje de cumplimiento del indicador	Resultado

Escala del Resultado:

Por-ciento de cumplimiento(pc)	Resultado
pc > 80%	Satisfactorio
60%<= pc <=80%	Aceptable
c < 60 %	No aceptable