

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 SUBPROCESO DE PRUEBAS AUTOMÁTICAS DE LA DISTRIBUCIÓN GNU-LINUX NOVA”

Autor:

Yusdel Pablo Delgado Morejón

Tutor(a):

Ing. Mónica María Albo Castro

Ciudad de La Habana

Junio 2012

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2012.

Firma del Autor

Yusdel Pablo Delgado Morejón

Firma del Tutor

Mónica María Albo Castro



“En la carrera por la calidad no hay línea
de meta.”

Robert Kearns



Agradecimientos

A mis padres: por la educación que me dieron.

A mi novia: por quererme y apoyarme en todo momento.

A la Revolución Cubana: por otorgarme el privilegio de estudiar en una universidad como la UCA.

A todos mis profesores: por intentar enseñarme todo lo que saben.

A mis compañeros: por su amistad desinteresada.

A todas las personas buenas que he conocido en estos 6 años.

Dedicatoria

Este trabajo de diploma está dedicado a mi familia por brindarme todo su apoyo a mi carrera y por confiar en mí.



Resumen

En el proceso de desarrollo de software siempre existe la posibilidad de cometer errores en todas las etapas. Actualmente es de vital importancia comprobar que el software sea liberado con la menor cantidad de errores posibles y que cumpla con los requisitos del sistema. Para llevar a cabo esta verificación, el software debe pasar por un proceso de pruebas con el fin de asegurar su calidad. En la actualidad los productos de la distribución GNU/Linux Nova pasan por un proceso de pruebas el cual no incluye indicadores de calidad que permitan la evaluación de los resultados de las pruebas automáticas. El presente trabajo de diploma tiene como objetivo definir el subproceso de pruebas automáticas de la distribución GNU/Linux Nova, parte esencial en la comprobación de ciertos aspectos de calidad de todo software moderno.

Con este propósito fueron definidos los requisitos de calidad e indicadores que debe cumplir una distribución GNU/Linux teniendo en cuenta los principales estándares de calidad internacionales. A partir de estos fue elaborado el subproceso de pruebas automáticas de la GNU/Linux Nova, el cual fue puesto a prueba con un caso de estudio y por expertos en las temáticas relacionadas. De esta evaluación se obtuvieron resultados satisfactorios, lo que confirma que la propuesta puede contribuir a la mejora de la calidad de los productos que se liberan en Nova.

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

Índice

Declaración de Autoría	II
Agradecimientos	IV
Dedicatoria	V
Resumen	VI
Índice.....	VII
Introducción	1
Fundamentación teórica de las pruebas de Software.	6
<i>Pruebas de Software</i>	<i>6</i>
<i>Estándares de calidad.....</i>	<i>10</i>
NC-ISO/IEC 9126-1: 2005 [13].....	10
ISO/IEC 14598 [17].....	14
ISO/IEC 25000 [23].....	14
<i>Herramientas para Pruebas Automáticas</i>	<i>18</i>
Nessus [24].....	18
Dogtail [25].....	19
Phoronix Test Suite [26].....	20
Mago [27].....	20
Snort [28].....	21
Ethereal [29].....	22
<i>Proceso de mejoras de la calidad en la UCI</i>	<i>22</i>
Subproceso de pruebas automáticas de la Distribución GNU/Linux Nova.	25
<i>Indicadores de Calidad para Pruebas Automáticas.....</i>	<i>25</i>
Requisitos de calidad de la Distribución GNU/Linux Nova.....	26
Métricas del Subproceso de Pruebas Automáticas.....	30
<i>Pruebas Automáticas del Phoronix Test Suite</i>	<i>33</i>
<i>Subproceso de Pruebas Automáticas</i>	<i>36</i>
¿Cómo diseñar casos de prueba que se correspondan con los requerimientos de una prueba automática?	37
Selección e implementación de pruebas automáticas.....	38
Ejecución de las pruebas.....	39
Evaluación de las pruebas.....	39
Evaluación del Subproceso de Pruebas Automáticas	41
<i>Métodos de Evaluación.</i>	<i>41</i>
<i>Evaluación de la viabilidad de la propuesta sobre la base del criterio de expertos. El método Delphi.</i>	<i>42</i>
Selección de los Expertos.....	43
Conformación del cuestionario.....	44
Desarrollo práctico y explotación de resultados.....	44
<i>Caso de estudio</i>	<i>49</i>
Ejecución de la propuesta.....	50
Conclusiones.....	53
Recomendaciones	54
Referencia Bibliográfica.....	55
Bibliografía Consultada.....	58
Anexos.....	59

1.-Manual de usuario de la herramienta Phoronix Test Suite:.....	59
Instalación del Phoronix Test Suite	59
Instalación de las pruebas de Phoronix	59
2.-Encuesta realizada a los expertos para determinar sus conocimientos referentes a la propuesta.	65
3.-Encuesta realizada a los expertos para evaluar la propuesta.	65
4.-Casos de pruebas para la personalización NovaMedia.	67
CP1. Rendimiento del procesador	67
CP2. Utilización de los dispositivos de E/S	68

Introducción

Los avances científico-técnicos de la humanidad han traído consigo el desarrollo de la actual sociedad de la información. En esta el desarrollo de las computadoras y la informática han jugado un papel fundamental con la automatización de una gran parte de los procesos de la vida cotidiana. Una parte inseparable de estos progresos es la industria del software, compuesta cada vez por más compañías y pequeñas empresas que desarrollan las aplicaciones con las que interactúan las personas.

El número creciente de empresas y compañías desarrolladoras de software ha fomentado la existencia de una ferviente competencia por alcanzar la preferencia de los clientes o usuarios, garantizando de esta manera su cuota en el mercado internacional. Para lograr una alta competitividad las empresas deben obtener productos con una alta calidad, razón por la cual deben garantizar que estos sean liberados con la menor cantidad de errores posibles. La calidad se consigue implementando estrictos procesos que incluyen, entre otras, actividades de pruebas que pueden ser manuales o automáticas.

En los inicios de la industria del software los programas se probaban de forma manual y generalmente por los propios desarrolladores, lo cual suscitaba una serie de problemas:

- Los desarrolladores no prueban todo el código pues consideran que su código es perfecto.
- La realización de las pruebas manuales no permite comprobar una serie de aspectos relacionados con el rendimiento del equipo o la resistencia de los sistemas informáticos a ataques por fuerza bruta.
- Al ser probado por desarrolladores y no por ingenieros se obvian aspectos propios que influyen en el producto final tales como: la usabilidad y la accesibilidad, los cuales se reflejan en la aceptación del cliente.

Todas estas razones conllevaron a que en la actualidad se tome muy en serio la realización de pruebas que permitan comprobar desde varias ópticas la calidad del producto. Estos esfuerzos los agradece el cliente pues le permite comparar varios productos basándose en estándares de calidad. Organizaciones reconocidas a nivel mundial tales como la IEEE(Instituto de Ingenieros Eléctricos y Electrónicos), la ISO(Organización Internacional para la Estandarización) y el SEI (Instituto de Ingeniería de Software) de la Universidad Carnegie Mellon de los EEUU, entre otras, son responsables por algunos de los más importantes estándares de calidad.

Cuba, a pesar de ser una pequeña isla del Caribe no ha estado ajena a los avances de la informática y en

la Feria Informática'04 se proclama oficialmente como productora y comercializadora de software, a través de INCUSOFT (Industria Cubana del Software) [1]. En el mundo cada país ha creado organizaciones con el fin de adaptar las normas internacionales a sus intereses. En el caso de Cuba esta organización es el Comité Técnico Nacional, adjunto a la ONN (Oficina Nacional de Normalización de Cuba).

Esta última fue creada con el objetivo de que los productos desarrollados por las empresas cubanas obtuvieran la máxima calidad, para lo cual dictó una serie de normativas como la ISO/IEC Guía 62:1996 "Requisitos generales para los organismos que operan las evaluaciones y la certificación/registro de los sistemas de calidad". El Comité Técnico CTN 18 "Tecnología de la Información" que funciona en la Isla es el responsable de la NC ISO/IEC 9126 [NC 2003], la cual identifica los parámetros de calidad necesarios para garantizar la calidad del software.

Desde sus inicios en el año 2002 la UCI (Universidad de las Ciencias Informáticas) fue concebida como pilar fundamental de la INCUSOFT. En la actualidad produce una gran cantidad de sistemas y soluciones informáticas, razón por la cual se hace imprescindible controlar la calidad de estos productos a fin de aumentar la valía de los mismos y elevar el prestigio de Cuba como productora de software a nivel mundial. Con este propósito se crea en la UCI Calisoft, organización que tiene por objetivo garantizar la máxima calidad de cada producto de software desarrollado en la universidad. En la actualidad es necesario un certificado de calidad emitido por Calisoft antes de que el producto sea entregado al cliente o liberado al mercado.

El bloqueo imperialista impuesto por la mayor potencia que el mundo haya conocido jamás le impide a Cuba adquirir los programas y las licencias necesarias para su uso y/o producción o comercialización. Entre estos programas se pueden mencionar las herramientas ofimáticas, software especializado en sistemas de bases de datos y suites de programación, así como las herramientas para la virtualización necesarias para trabajar con algunos lenguajes de programación. Es por esto que en el año 2005 el Ministerio de Informática y Comunicaciones orienta realizar una migración hacia el software libre. [2]

Es en este marco en el que surge el proyecto Nova en este mismo año con el propósito de crear una plataforma libre para sustituir a las privativas de la familia Microsoft Windows principalmente. Este proyecto se ha desarrollado de manera vertiginosa, afianzándose en el desarrollo de personalizaciones adecuadas a las exigencias de cada cliente específico. El ciclo de desarrollo de este proyecto comienza con la elaboración del sistema operativo base a partir del cual se le añaden las aplicaciones que el cliente solicita además de adecuar su rendimiento para los equipos a los que estaría destinado [3]. Una de estas

versiones se realizó para el Concurso ACM-ICPC (Competición Internacional Universitaria de Programación, organizado por la Association for Computing Machinery ACM), aunque se han realizado personalizaciones para varios clientes nacionales y foráneos.

El concepto de calidad en la distribución Nova se comienza a manejar luego de que la versión Baire fuera liberada en el año 2009 sin obtener resultados satisfactorios, debido entre otras cosas a que fue probada empíricamente por los propios desarrolladores. A partir de este momento es creado el grupo Qalit, el cual tiene la tarea de asegurar la calidad de cada producto elaborado en Nova. Con este fin se comenzaron investigaciones para lograr reunir los elementos necesarios para realizar las pruebas efectivamente y preparar a los probadores, inexistentes en ese momento.

En el año 2011 el trabajo de tesis titulado “Proceso de pruebas de la distribución GNU/Linux Nova”, dio los primeros pasos en el ordenamiento del proceso de pruebas, definiendo un proceso de acuerdo a las mejores prácticas de CMMI unidas al modelo de desarrollo W. Este trabajo de tesis sentó las bases de un proceso de pruebas coherente, pues lo definió de forma estructurada, delimitando cuando era el momento oportuno para realizar las pruebas e introdujo el uso de una herramienta para la realización de las pruebas automáticas.

Durante la utilización de este proceso los líderes de las líneas de desarrollo de Nova comprobaron la utilidad de esta herramienta pero a su vez plantearon la necesidad de que fuesen probados otros aspectos del sistema de manera automática. Todo esto se debe, en parte, a que el actual subproceso de pruebas automáticas de Nova es una generalización y carece de una especificación de las pruebas automáticas que deben ser aplicadas en el desarrollo del proyecto.

De esta situación se deriva el **Problema a Resolver**: Las pruebas automáticas de la distribución GNU/Linux Nova se realizan solamente en función de la herramienta Phoronix Test Suite, lo cual dificulta la evaluación de los indicadores que definen la calidad, medibles automáticamente.

Como **Objeto de Estudio**: Los indicadores de calidad asociados a las pruebas automáticas.

Como **Campo de Acción**: El subproceso de pruebas automáticas de la distribución cubana de GNU/Linux Nova.

Esta investigación tiene por **Objetivo General**: Definir un subproceso de pruebas automáticas para la distribución GNU/Linux Nova que evalúe los indicadores de calidad definidos, que sean medibles automáticamente.

Para cumplir con este propósito se plantean los siguientes **Objetivos Específicos**:

- Precisar el desarrollo y las tendencias de las herramientas para la realización de las pruebas automáticas a sistemas operativos.
- Determinar los indicadores de calidad de software definido por los principales estándares.
- Definir los indicadores de calidad que se deben medir en un sistema operativo para evaluar su calidad y cuáles de estos se pueden evaluar de forma automática.
- Definir el subproceso de pruebas automáticas de la distribución GNU/Linux Nova en función de indicadores de calidad de sistemas operativos.
- Evaluar el subproceso y los manuales propuestos para determinar las ventajas y desventajas de los mismos.

Con el presente trabajo de diploma se plantea que: Definiendo el subproceso de pruebas automáticas de la distribución GNU/Linux Nova basándose en indicadores de calidad se podría mejorar la evaluación de aquellos que sean medibles automáticamente y por consecuencia la evaluación de la calidad del producto.

Los métodos Teóricos que se utilizan para la realización de esta investigación son:

- Analítico – Sintético: Durante el desarrollo de esta investigación se analizan documentos, normas de calidad, páginas de Internet y otros artículos relacionados con las pruebas automáticas de software, para extraer los elementos más significativos relacionados con el tema y componer la fundamentación de la investigación.
- Inductivo – Deductivo: En el desarrollo de esta investigación se estudian distintos estándares internacionales para procesos de pruebas y se adaptan a las necesidades particulares del proyecto Nova.

Métodos Empíricos utilizados en esta investigación:

- Observación: La observación científica como método consiste en la percepción directa del objeto de investigación. La observación investigativa es el instrumento universal del científico. La observación permite conocer la realidad mediante la percepción directa de los objetos y fenómenos.
- Experimental: Es el más complejo y eficaz de los métodos empíricos. En este método el investigador interviene sobre el objeto de estudio modificando a este directa o indirectamente para

crear las condiciones necesarias que permitan revelar sus características fundamentales y sus relaciones esenciales

➤ Método de recopilación de información:

- Encuesta: Se utiliza para recopilar información a través de diferentes encuestas, a las personas que tienen que ver de una forma u otra con el subproceso de pruebas automáticas.

El presente trabajo de diploma se encuentra distribuido de la siguiente forma:

Capítulo #1 Fundamentación teórica de las pruebas automáticas de software: En este capítulo se hace un estudio de las normas referentes a la calidad en vistas de definir cuáles son los indicadores de la calidad medibles de forma automática. Además se hace un estudio de las principales herramientas utilizadas para la realización de las pruebas automáticas de software.

Capítulo #2 Subproceso de pruebas automáticas de la Distribución GNU/Linux Nova: Se realiza un estudio profundo y consciente del actual subproceso de pruebas automáticas y se define el nuevo subproceso de pruebas automáticas en función de métricas de calidad que determinen la calidad del producto de forma cuantitativa.

Capítulo #3 Evaluación del Subproceso de Pruebas Automáticas: Se evalúa la efectividad de este subproceso, así como su impacto en la calidad de la Distribución GNU/Linux Nova.

Fundamentación teórica de las pruebas de Software.

Cumplir con las expectativas de un cliente en la industria del software requiere de un proceso de desarrollo organizado. Este proceso debe coordinar los esfuerzos de las personas involucradas, explicando los pasos a seguir para realizar la empresa en el tiempo y costos previstos, y con la calidad que se requiere. El desarrollo de todo sistema informático se encuentra dividido en una serie de etapas que incluyen: Levantamiento de requisitos, Diseño, Implementación, Pruebas y Mantenimiento. [4]

Esta investigación se centra en la fase de Pruebas en la cual se desarrollan toda una serie de actividades con el objetivo de comprobar que los requerimientos del sistema fueron satisfechos. Entre estas actividades se encuentran las pruebas manuales y las pruebas automáticas del sistema. En estas últimas se utilizan herramientas informáticas que comprueban determinadas funcionalidades que pueden ser especificaciones del usuario o especificaciones de calidad contempladas en estándares internacionales. A continuación se describen los indicadores de calidad de un producto de software de acuerdo a los estándares de calidad. Además de mencionar las aplicaciones más importantes para la realización de las pruebas automáticas así como sus principales características.

Pruebas de Software

Un software o programa informático durante su ciclo de desarrollo requiere de un orden entre todos los aspectos ingenieriles para garantizar que el producto sea elaborado con la mayor calidad posible. Como son las personas las que desarrollan el software, es inherente entonces que la aplicación contenga errores, por lo cual se hace necesario realizarle pruebas al software antes de entregarlo al cliente.

Las pruebas de software consisten en ejercitar el sistema utilizando datos similares a los datos reales, observar los resultados y deducir la existencia de errores o insuficiencias del programa a partir de las anomalías de los resultados. [5]

En la actualidad existen varios estándares de pruebas de software que han sido elaborados por organizaciones como The Testing Standards Working Party (Grupo de Trabajo de Estándares de Prueba)

entre los que se encuentran:

✓ **BS 7925-1 Vocabulary of Terms in Software Testing [6]**

(Vocabulario de Términos en Pruebas de Software). Este estándar es complemento del BS 7925-2 y es un esfuerzo por llegar a un consenso en cuanto a definiciones utilizadas a diario por los especialistas de la calidad y a los cuales en ocasiones se les altera su significado. Contiene más de 200 definiciones de términos informáticos, permitiendo un mejor entendimiento de estos términos, así como esclareciendo cualquier duda que pudiera surgir a la hora de utilizar alguna de estas palabras durante la realización de las pruebas.

Entre los principales términos de interés para esta investigación se encuentran definidos [7]:

- Pruebas: El proceso de realizar la evaluación de un programa, producto o sistema, por medios manuales o automatizados, para verificar que cumple los requisitos especificados y para identificar las diferencias entre los resultados esperados y los reales.
- Pruebas Automáticas: Pruebas individuales creadas para ser ejecutadas sin la intervención directa del probador.
- Métricas: una métrica debe ser una medida real y objetiva de algo como la cantidad de errores por líneas de código.

✓ **BS 7925-2 Software Component Testing Standard [8]**

(Estándar de Pruebas a Componentes de Software). El objetivo de esta norma es permitir la medición y comparación de las pruebas realizadas sobre los componentes de software. Esto permitirá a los usuarios de este estándar mejorar directamente la calidad de sus pruebas de software, y mejorar la calidad de sus productos de software. Esta norma prescribe características del proceso de pruebas, describiendo un número de técnicas para el diseño de casos de pruebas y medición.

Existen algunos estándares de pruebas de software creados por la IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) como:

✓ **IEEE 829 Test Documentation [9]**

(Documentación de Pruebas). Esta norma determina como deben conformarse un conjunto de documentos para ser usados en ocho etapas definidas de pruebas de software. La norma no especifica cómo deben ser llenados, solo la estructura que deben seguir.

- **Plan de pruebas:** Documento rector de todo el proceso en el cual debe ser especificado como serán realizadas las pruebas, quien las va a llevar a cabo, que es lo que va a ser probado, cuánto tiempo va a tomar la realización de estas pruebas y cuan profunda va a ser la evaluación del producto.
 - **Especificaciones del diseño de las pruebas:** En este documento deben quedar detalladas las condiciones de pruebas, los resultados esperados de cada prueba, así como el criterio de aprobación de la prueba.
 - **Especificación de los Casos de Pruebas:** Los datos que serán usados en las pruebas definidas en el documento "Especificaciones del diseño de las pruebas" son reflejados en este documento.
 - **Especificación del procedimiento de pruebas:** Debe ser detallado en este documento como debe ser realizada cada prueba, los pasos que debe seguir lo cual incluye las precondiciones que pudieran ser necesarias para realizar una prueba en particular.
 - **informe de transmisión de un componente de pruebas:** Este documento recoge cuando un componente de software asciende de un nivel de pruebas hacia otro, luego de superarlo exitosamente.
 - **Registro de pruebas:** En este documento queda recogido que prueba se realizó, quien la llevo a cabo, cual fue el resultado y si el software superó la prueba o no.
 - **Informe de incidentes de prueba:** En caso de que una prueba tenga un resultado insatisfactorio, en este documento queda detallada toda la información referente a la prueba lo cual incluye también cualquier evidencia que pueda haber influido en el error.
 - **Informe sumario de las pruebas:** Este informe provee toda la información importante revelada por las pruebas realizadas lo cual incluye evaluaciones de la calidad de las mismas, que tiempo tomo la realización de cada prueba, así como el apego de la aplicación de acuerdo al criterio de aceptación definido por las partes interesadas.
- ✓ **IEEE 1008 Unit Testing [10]**

(Pruebas Unitarias). Metodología que establece requisitos de calidad y propone la definición, ejecución, análisis y validación de los procesos, además de definir las métricas de calidad para productos de software. La metodología se extiende por todo el ciclo de vida del software y tiene por objetivo principal brindar herramientas a todos los roles relacionados con la realización de pruebas unitarias para hacer más

fácil su tarea.

✓ **IEEE 730 Standard for Software Quality Assurance Plans(SQAP) [11]**

(Estándar para Planes de Aseguramiento de la Calidad de Software). La norma especifica el formato y el contenido de los planes de aseguramiento de la calidad de software. Esta norma se encuentra compuesta por 15 secciones. Estas secciones incluyen la descripción de las acciones que se deben llevar a cabo para asegurar la calidad de los procesos fundamentales y que estos contribuyan al aseguramiento de la calidad del producto.

En mayo del año 2007 dio comienzo un proyecto llamado **ISO/IEC 29119** que tiene como propósito crear un estándar para la realización de las pruebas de software. Este proyecto aunque aún en desarrollo promete proporcionar una guía acertada para la realización de las pruebas de software y está previsto que sustituya a los estándares BS 7925-1, BS 7925-2, IEEE 829 y el estándar IEEE 1008.

De los conceptos de pruebas analizados se pudo apreciar que existe uniformidad de opinión, por tanto en esta investigación se consideran como pruebas de software el concepto dado por el estándar BS 7925-1. Las pruebas de acuerdo a la forma en que son realizadas se separan en dos grandes grupos [12]:

- **Manuales:** son las realizadas por los especialistas en calidad de software y tienen por objetivo detectar aquellas imperfecciones que afectan el normal desempeño de la aplicación, o que podrían afectarlo en un futuro.
- **Automáticas:** son aquellas en las que una herramienta realiza un conjunto de operaciones sobre el software especificado de forma automática. Los resultados pueden variar desde una respuesta visual, hasta un informe detallado con las estadísticas de la aplicación.

A pesar de que las pruebas manuales revisten una especial importancia, para realizarlas se hace necesario contar con un personal altamente preparado en ingeniería de software, además de otras disciplinas dependiendo del producto que se desarrolle. Por otra parte las pruebas automáticas, tal como se menciona en el estándar BS 7925-1, no son realizadas por personas sino por aplicaciones especializadas. Esto asegura el resultado de las mismas y permite que este tipo de pruebas puedan ser realizadas por un personal que no necesariamente tiene que tener avanzados conocimientos de ingeniería de software. Este tipo de pruebas comprueban aspectos, que de tratar de ser probados de forma manual, sería un proceso tedioso y extenso.

Estándares de calidad

El software como producto debe cumplir con una serie de requisitos del sistema. Adicionalmente para poder ser comercializado a nivel internacional debe cumplir con una serie de indicadores de calidad. Estos han sido definidos en estándares elaborados por organizaciones reconocidas a nivel internacional que dedican su esfuerzo al estudio y conformación de normas que definan la calidad del producto de software.

NC-ISO/IEC 9126-1: 2005 [13]

Es la versión o adaptación cubana de la ISO/IEC 9126-1:2005. Este estándar establece una serie de características para determinar el grado de calidad de un producto de software. En esta norma se definen seis aspectos fundamentales de la calidad:

1. Funcionabilidad: “Es la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuando el software se usa bajo las condiciones especificadas.” De acuerdo a la norma esta característica describe que hace el software para satisfacer las necesidades, sin importarle el cuándo ni el cómo.

- Idoneidad: Es la capacidad del software para mantener un conjunto apropiado de funciones para las tareas y objetivos especificados por los usuarios.
- Precisión: Capacidad del software para proporcionar efectos o resultados correctos o convenidos con el grado de exactitud necesario.
- Seguridad (informática): Capacidad del producto de software para proteger la información y los datos, para que personas o sistemas desautorizados no puedan leer o modificar los mismos, y las personas o sistemas autorizados tenga el acceso a ellos.
- Interoperabilidad: Capacidad del producto de software para interactuar recíprocamente con uno o más sistemas especificados.
- Conformidad con la funcionalidad: Capacidad del software de apegarse a la normas, leyes, regulaciones y similares que se le apliquen en términos relativos a la funcionalidad.

2. Confiabilidad: La capacidad del producto de software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas.

- Madurez: Capacidad del producto de software de evitar un fallo total como resultado de haberse

producido un fallo del software.

- Tolerancia ante fallos: Capacidad del software de mantener un nivel de ejecución o desempeño especificado en caso de fallos del software.
- Recuperabilidad: Capacidad del producto de software de restablecer un nivel de ejecución especificado y recuperar los datos directamente afectados en caso de fallo total.
- Conformidad con la confiabilidad: capacidad del producto de software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares relativas a la confiabilidad.

3. Usabilidad: capacidad del producto de software de ser comprendido, aprendido, utilizado y de ser atractivo para el usuario, cuando se utilice bajo las condiciones especificadas.

- Comprensibilidad: capacidad del producto de software para permitirle al usuario entender si el software es idóneo, y cómo puede usarse para las tareas y condiciones de uso particulares.
- Cognoscibilidad: capacidad del producto de software para permitirle al usuario aprender su aplicación.
- Operabilidad: capacidad del producto de software para permitirle al usuario operarlo y controlarlo.
- Atracción: capacidad del producto de software de ser atractivo o amigable para el usuario.
- Conformidad con la usabilidad: capacidad del producto de software para adherirse a las normas, convenciones, guías de estilo o regulaciones relativas a la usabilidad.

4. Eficiencia: capacidad del producto de software para proporcionar una ejecución o desempeño apropiado, aprovechando eficientemente los recursos y brindando una experiencia agradable al usuario bajo condiciones determinadas.

- Rendimiento: capacidad del producto de software para proporcionar apropiados tiempos de respuesta y procesamiento, así como tasas de producción de resultados al realizar su función bajo condiciones establecidas.
- Utilización de recursos: capacidad del producto de software para utilizar la cantidad y el tipo apropiado de recursos cuando realiza su función bajo las condiciones establecidas.
- Conformidad de la eficiencia: capacidad del producto de software de adherirse a las normas o convenciones que se relacionan con la eficiencia.

5. **Mantenibilidad:** capacidad del producto de software de ser modificado. Las modificaciones pueden incluir las correcciones, mejoras o adaptaciones del software a cambios en el ambiente, así como en los requisitos y las especificaciones funcionales.

- **Diagnosticabilidad:** capacidad del producto de software de ser objeto de un diagnóstico para detectar deficiencias o causas de los fallos totales en el software, o para identificar las partes que van a ser modificadas.
- **Flexibilidad:** capacidad del producto de software para permitir la aplicación de una modificación especificada.
- **Estabilidad:** capacidad del producto de software para minimizar los efectos inesperados de las modificaciones realizadas.
- **Contrastabilidad:** capacidad del producto del software para permitir la validación de modificaciones realizadas a su código.
- **Conformidad de la mantenibilidad:** capacidad del producto de software para adherirse a las normas o convenciones que se relacionan con la mantenibilidad.

6. **Portabilidad:** capacidad de producto de software de ser transferido de un ambiente a otro.

- **Adaptabilidad:** capacidad del producto de software de ser adaptado a los ambientes especificados sin aplicar acciones o medios de otra manera que aquellos suministrados con el propósito de que el software cumpla sus fines.
- **Instalabilidad:** capacidad del producto de software de ser instalado en un ambiente especificado.
- **Coexistencia:** capacidad del producto de software de coexistir con otro software independiente en un ambiente común y compartir los recursos comunes.
- **Remplazabilidad:** capacidad del producto de software de ser usado en lugar de otro producto de software especificado para los mismos fines y en el mismo ambiente.
- **Conformidad con la portabilidad:** capacidad del producto de software de adherirse a las normas o convenciones relativas a la portabilidad.

Dentro de esta familia de normas existen tres regulaciones más que incluyen:

✓ **ISO/IEC 9126-2 [14]**

Esta norma define las métricas externas para la determinación de la calidad de un producto de software. Las métricas externas son aquellas que miden el comportamiento del sistema operativo donde el software a evaluar se ejecuta. Esta norma está compuesta por 6 grupos de métricas agrupadas como sigue:

- Funcionalidad: Este grupo incluye las métricas de: idoneidad, precisión, interoperabilidad, seguridad y funcionabilidad.
- Confiabilidad: Las métricas madurez, tolerancia ante fallos, recuperabilidad y confiabilidad están ubicadas en este grupo.
- Usabilidad: Comprendidas en este grupo se encuentran: comprensibilidad, capacidad de aprendizaje, operatividad, atractivo y el cumplimiento de la usabilidad.
- Eficiencia: Incluidas en este grupo se encuentran las métricas: comportamiento en el tiempo, utilización de recursos y cumplimiento de la eficiencia.
- Mantenimiento: Incluye las métricas: analizabilidad, mutabilidad, estabilidad, capacidad de prueba y cumplimiento con el mantenimiento.
- Portabilidad: Las métricas de adaptabilidad, instalabilidad, coexistencia, reemplazabilidad y cumplimiento de la portabilidad se encuentran en este grupo.

Cada métrica contenida en esta norma se encuentra descrita de forma tal que es fácil su entendimiento y aplicación. Dada la característica de esta norma de determinar el desempeño del sistema operativo donde se ejecutan las aplicaciones a evaluar, se torna muy útil al tratar de medir el desempeño de un sistema operativo como la distribución Nova.

✓ **ISO/IEC 9126-3 [15]**

Esta norma hace referencia a las métricas internas de la calidad del producto de software, las cuales se concentran en el producto de software en sí mismo. Se encuentra estructurada de forma similar a la ISO 9126-2 a pesar de que las métricas que ofrece tienen un enfoque distinto. Esta norma está diseñada para proporcionarles herramientas a los usuarios de la métrica para determinar, a partir de un software no ejecutable o que esté en fase de desarrollo aun, la calidad del producto final.

✓ **ISO/IEC 9126-4 [16]**

Este estándar proporciona métricas de calidad en el uso para la medición de los atributos definidos en la ISO/IEC 9126-1. Las métricas contenidas en esta norma no fueron creadas con la intención de hacerlas cumplir estrictamente. Desarrolladores, evaluadores y administradores de la calidad podrían seleccionar métricas de esta norma para evaluar productos, definir requerimientos, medir aspectos de la calidad y otros propósitos.

ISO/IEC 14598 [17]

Este grupo de normas (seis) están diseñadas para complementar al grupo de las ISO/IEC 9126, de forma que abarquen completamente el proceso de calidad de software.

- La primera parte de esta norma es simplemente una introducción a las demás, explica la relación entre esta y el modelo de calidad propuesto por la ISO/IEC 9126 en su conjunto.
- La segunda parte se hace cargo de planificar y gestionar todo el proceso de evaluación. [18]
- La selección de los atributos que representan los requisitos de calidad se encuentra reflejado en la tercera parte. Esta se encuentra especialmente enfocada en definir y reportar los requerimientos que podrían ser útiles para predecir la calidad del producto final, aún cuando se encuentre en fases intermedias de desarrollo. [19]
- En la cuarta parte de esta norma se describe el proceso de calidad y se muestran ejemplos de métodos de evaluación. [20]
- Las herramientas para lograr la adaptación del proceso general de evaluación a un entorno específico se encuentran descritas en la quinta parte de esta norma. [21]
- En la sexta parte de esta norma se describe la documentación que puede llevarse en un proceso de calidad de software. [22]

ISO/IEC 25000 [23]

Conocida también como SQuaRE (Evaluación y Requerimientos de Calidad de un Producto de Software, por sus siglas en inglés). Este grupo de normas se basan en la revisión y unificación de las normas ISO/IEC 9126 y la ISO/IEC 14598.

Esta subdividida en:

- ISO/IEC 2500n: División de dirección de calidad.
- ISO/IEC 2501n: División del modelo de calidad.
- ISO/IEC 2502n: División de medida de calidad.
- ISO/IEC 2503n: División de requisitos de calidad.
- ISO/IEC 2504n: División de evaluación de calidad.

Esta familia de normas nace de la necesidad de actualizar a las decadentes 9126 y 14598, pues estas últimas pertenecen a la primera generación de estándares de calidad de un producto de software. Por su parte la norma 25000 al pertenecer a la segunda generación de estándares de calidad de productos de software cuenta con una serie de nuevas características:

- Este estándar se centra en el lado del producto.
- Por ser una revisión de la 9126 hereda las mismas características de calidad de software.
- El ciclo de vida de la calidad del producto software se basa en tres fases principales: producto bajo desarrollo, producto en operación y producto en uso.
- Incorpora modificaciones estructurales respecto a la clasificación de las características y subcaracterísticas que ofrece la 9126. Para tener una mejor idea de cómo han sido modificadas estas características veamos la siguiente tabla:

Tabla 1: Comparación entre NC ISO/IEC 9126-1 y ISO/IEC 25000

NC ISO/IEC 9126-1	ISO/IEC 25000
Funcionabilidad	Mantiene su nombre.
Idoneidad	Desaparece este concepto.
Precisión	Mantiene su nombre.
Interoperabilidad	Mantiene su nombre.
Conformidad con la Funcionabilidad	Mantiene su nombre.
	Se añade la subcaracterística Consistencia. La consistencia se refiere a la ausencia de

Fundamentación teórica de las pruebas de Software.

	contradicciones aparentes en los datos.
	Validez es una nueva característica añadida. Hace referencia a información volátil.
	Integridad es el grado en que todos los valores necesarios se han asignado y almacenado en el sistema informático. Esta nueva subcaracterística es añadida.
	La Exactitud se define como el grado en que un valor de datos se ajusta a su valor actual o especificado. Se añade esta nueva característica.
	Se añade la subcaracterística Seguridad. Esta hace referencia a la capacidad de permitir el acceso a los datos solo a los usuarios o aplicaciones autorizados.
Confiabilidad	Mantiene su nombre.
Madurez	Desaparece esta subcaracterística.
Tolerancia al defecto	Desaparece esta subcaracterística.
Recuperabilidad	Mantiene su nombre.
Fiabilidad	Desaparece esta subcaracterística.
Conformidad con la confiabilidad	Mantiene su nombre.
	Se añade la subcaracterística

Fundamentación teórica de las pruebas de Software.

	Disponibilidad. Esta hace referencia a la capacidad de mantener siempre los datos disponibles.
Usabilidad	Mantiene su nombre.
Comprensibilidad	Mantiene su nombre.
Instructibilidad	Desaparece esta subcaracterística.
Atractivo	Mantiene su nombre.
Conformidad con la usabilidad	Mantiene su nombre.
	Se añade la característica Manejabilidad . Esta hace referencia a la capacidad de almacenar los datos adecuadamente desde un punto de vista funcional.
Eficiencia	Mantiene su nombre.
Comportamiento en el tiempo	Desaparece esta subcaracterística.
Utilización de los recursos	Mantiene su nombre.
Eficacia	Desaparece esta subcaracterística.
Conformidad con la eficiencia	Mantiene su nombre.
Mantenibilidad	Mantiene su nombre.
Analizabilidad	Mantiene su nombre.
Confiabilidad	Desaparece esta subcaracterística.
Estabilidad	Desaparece esta subcaracterística.
Ensayabilidad	Desaparece esta subcaracterística.
Conformidad con la mantenibilidad	Mantiene su nombre.
	Se añade la característica Mutabilidad .

	Esta hace referencia a la capacidad de que los datos sean cambiados en su tipo, longitud o el valor asignado.
Portabilidad	Mantiene su nombre.
Adaptabilidad	Mantiene su nombre.
Instalabilidad	Desaparece esta subcaracterística.
Coexistencia	Desaparece esta subcaracterística.
Reemplazabilidad	Desaparece esta subcaracterística.
Conformidad con la portabilidad	Mantiene su nombre.

A pesar de que la norma ISO/IEC 25000 incluye nuevas subcaracterística de calidad y en algunos casos define mejor otras, esta norma es aún un reporte técnico y no ha sido aprobado su uso en nuestro país.

Herramientas para Pruebas Automáticas

Nessus [24]



Nessus es un programa privativo de escaneo de vulnerabilidades para diversos sistemas operativos. Consiste en **nessusd**, el demonio¹ Nessus, que realiza el escaneo en el sistema objetivo, y **nessus**, el cliente (en modo consola o entorno gráfico) que muestra el avance y el reporte de los escaneos. Desde consola **nessus** puede ser programado para hacer escaneos con Cron.²

En operación normal, **nessus** comienza escaneando los puertos con su

¹ Un demonio es un tipo de programa que se ejecuta discretamente en segundo plano en sistemas operativos de tipo Unix, en lugar de bajo el control directo de un usuario, a la espera de ser activado por la ocurrencia de un evento o condición específica.

² Programa que permite a usuarios Linux/Unix ejecutar automáticamente comandos o scripts (grupos de comandos) a una hora o fecha específica. Es usado normalmente para comandos de tareas administrativas, como respaldos, pero puede ser usado para ejecutar cualquier aplicación o funcionalidad.

propio software o con el programa **nmap**³ para buscar puertos abiertos y después intenta varios **exploits**⁴ para atacarlos. Las pruebas de vulnerabilidad, disponibles como una larga lista de plug-ins, son escritas en NASL (Lenguaje de Scripts de Ataque Nessus por sus siglas en inglés), un lenguaje de script optimizado para interacciones personalizadas en redes.

Opcionalmente, los resultados del escaneo pueden ser exportados en reportes de varios formatos, como texto plano, XML, HTML, y LaTeX. Los resultados también pueden ser guardados en una base de conocimientos para referencia en futuros escaneos de vulnerabilidades. Algunas de las pruebas de vulnerabilidades de Nessus pueden causar que los servicios o sistemas operativos se corrompan y caigan. El usuario puede evitar esto desactivando "unsafe test" (pruebas no seguras) antes de escanear.

Dogtail [25]



Dogtail es un framework⁵ de pruebas que contiene un marco de automatización escrito en Python, uno de los lenguajes de programación más fácil de aprender y de usar en la actualidad. Es una aplicación de software libre, por tanto se puede usar y modificar su código. Actualmente soporta pruebas para aplicaciones del entorno de escritorio

Gnome, así como muchas aplicaciones Gtk. Utiliza tecnologías de acceso AT-SPI (Interfaz de Proveedores de Servicio de Tecnologías de Accesibilidad por sus siglas en inglés) la cual permite la comunicación con aplicaciones de escritorio mediante una interfaz de proveedor de servicios para las tecnologías auxiliares disponibles en la plataforma Gnome.

Esta herramienta está orientada a objetos por lo cual un programador avanzado puede desarrollar clases y librerías de ayuda e incluirlas fácilmente en la aplicación. Para escribir los scripts de las pruebas de esta herramienta se puede utilizar un editor llamado Sniff (/usr/bin/sniff o "AT-SPI Browser" en el menú), esta herramienta facilita la creación de scrips de prueba proporcionando ayuda a la hora de escribirlos.

3 Nmap ("Mapeador de la Red") es un programa libre y de código abierto con la finalidad de la exploración de la red o la auditoría de seguridad.

4 Del inglés to exploit, explotar, aprovechar) es el nombre con el que se identifica un programa informático malicioso, o parte del programa, que trata de forzar alguna deficiencia o vulnerabilidad de otro programa.

5 Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.

Phoronix Test Suite [26]



Phoronix Test Suite es una de las aplicaciones más desarrolladas en la realización de pruebas automáticas, está compuesto de herramientas diseñadas por Phoronix desde el año 2004 en colaboración con los principales productores de hardware y software. Es de código abierto y difundido bajo la licencia GNU GPL v3.

Originalmente desarrollado para Linux, en la actualidad soporta a los sistemas operativos Open Solaris, Apple Mac OS X y Microsoft Windows.

Contiene más de 160 perfiles de prueba que incluyen el chequeo del consumo de la batería, la CPU, gráficos, memoria del sistema, almacenamiento en disco y el chequeo de los componentes de la placa base. El Phoronix Test Suite se compone de un núcleo de procesamiento ligero (**pts-core**) con puntos de referencia, que consiste en un perfil basado en XML y secuencias de comandos de recursos relacionados. La herramienta no permite que le sean adicionadas pruebas desarrolladas por terceros, solo las proporcionadas por los colaboradores reconocidos. Esta herramienta permite además mostrar en un visor los resultados, basados en estilos Web con soporte opcional para subirlos a “<http://global.phoronix-test-suite.com>” y realizar comparaciones con distribuciones o sistemas operativos que corran en un hardware semejante.

Mago [27]



Mago es una aplicación basada principalmente en LDTP (Linux Desktop Testing Project), un framework para pruebas de escritorio, programado en C⁶. Esta herramienta desarrollada por el equipo de Gnome contiene un corredor de pruebas el cual permite ejecutarlas constantemente y es capaz de presentar informes agradables con los resultados de las mismas. Las pruebas están organizadas por distintas carpetas de archivos XML, donde cada cual contiene uno o más casos de pruebas. Trabajar con Mago requiere

⁶ Es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

tener conocimientos del lenguaje Python para poder diseñar las pruebas pues se deben crear de forma manual en este lenguaje.

Esta herramienta actualmente soporta los entornos de escritorio Gnome 2.0 de forma estable y Gnome 3.0 de forma inestable.

Snort [28]



Snort es una herramienta de código abierto para la detección de intrusiones de red (para el sistema). Es capaz de llevar a cabo análisis de tráfico en tiempo real y registro de paquetes en redes. Puede realizar análisis de protocolos, búsqueda o identificación de contenido y puede ser utilizado para detectar una gran variedad de ataques y pruebas, como por ejemplo: buffer overflows (desbordamiento de búffer), escaneos indetectables de puertos ("stealth port scans"), ataques a CGI⁷, pruebas de SMB⁸ ("SMB Probes"), intentos de reconocimientos de sistema operativos ("OS fingerprinting") y muchos más. Snort utiliza un lenguaje flexible basado en reglas para describir el tráfico que debería recolectar o dejar pasar, y un motor de detección modular. Especialistas en seguridad han sugerido que la Consola de Análisis para Bases de Datos de Intrusiones (Analysis Console for Intrusion Databases, ACID) sea utilizada con Snort.

Antes de instalar Snort se debe comprobar que se tienen una serie de paquetes de software instalados. Estos son: libpcap, PCRE, Libnet y Barnyard.

Snort tiene dos componentes principales:

- un motor de detección que utiliza una arquitectura de plug-in modular (el "motor Snort"),
- un lenguaje de reglas flexible para describir el tráfico que se recoge (las "reglas de Snort").

El motor de Snort se distribuye en forma de código fuente y los binarios para las distribuciones populares de Linux y Windows. Es importante señalar que el motor de Snort y las reglas Snort se distribuyen por separado.

⁷ CGI (Interfaz de entrada común) es la capa o interfaz que se encuentra entre un formulario de una página Web y el servidor permite la interacción con el servidor de forma indirecta para una mayor seguridad.

⁸ Protocolo de red que se utiliza para compartir archivos e impresoras.

Las reglas del Equipo de Investigación de Vulnerabilidades Sourcefire (VRT) son las reglas oficiales de Snort. Cada regla es desarrollada y probada siguiendo los mismos estándares que sigue la VRT para el desarrollo de Sourcefire (su versión comercial). Estas reglas son distribuidas bajo la licencia de reglas VRT. Esta licencia permite la modificación de las reglas pero impide su comercialización.

Ethereal [29]



Ethereal es todavía técnicamente software beta, pero tiene un amplio conjunto de funciones y es adecuado para su uso de forma profesional. A continuación se describen una lista de características, vigente en la versión 0.9.14, mencionadas sin ningún orden en particular:

- Los datos pueden ser capturados en tiempo real a partir de una conexión de red activa, o leer desde un archivo de captura.
- Puede leer archivos de captura de varias aplicaciones de escaneo de red, sniffers fundamentalmente.
- Puede leer datos en tiempo real sobre redes Ethernet, FDDI, PPP, Token Ring, IEEE 802.11, y el clásico IP sobre ATM, y las interfaces de loopback⁹ (al menos en algunas plataformas, no todos los tipos son compatibles con todas las plataformas).
- Los datos capturados de la red pueden ser consultados a través de una interfaz gráfica de usuario, o por medio del modo TTY (consola o terminal) del programa, conocido por “tethereal”.
- Actualmente la aplicación puede diseccionar 759 protocolos de transmisión de datos.
- La salida puede ser guardada o impresa en forma de texto plano o PostScript®.
- Todas, o una parte de las trazas capturadas, pueden ser guardadas en el disco.

Proceso de mejoras de la calidad en la UCI

La calidad de los productos desarrollados en la UCI es controlada por el grupo Calisoft. Este grupo utiliza normas internacionales en sus procesos para garantizar la máxima calidad de cada producto inspeccionado, la norma NC ISO/IEC 9126-1 es una de ellas.

⁹ Interfaz de red virtual.

Para la realización de pruebas automáticas actualmente cuentan con varias herramientas que facilitan la realización de estas actividades, entre ellas se cuentan herramientas de carga y estrés, funcionales, y varias de seguridad como:

- Jmeter
- Selenium
- Meld
- Brutus
- Dirbuster
- SQLmap
- SQL Brute
- Webcarab
- Nikto
- Wapiti
- OpenVas
- Nmap
- Metasploit
- DBMonster

A pesar del número tan grande de herramientas que utilizan estas se encuentran enfocadas hacia aplicaciones de escritorio por lo cual se hace inapropiada su utilización para evaluar el desempeño de una distribución GNU/Linux tal como Nova.

En este capítulo han sido analizados los principales estándares de calidad que definen la utilización de métricas para determinar la calidad de un producto de software. En este marco se determinó la utilización de la norma NC ISO/IEC 9126-1 por varias razones:

- Está normalizada en Cuba, lo cual indica que ha sido adaptada a las condiciones de la industria de software de la isla.
- Es la utilizada por Calisoft para definir la calidad de los productos que evalúa en su quehacer diario.
- Incluye subcaracterísticas que coinciden con los aportes de la ISO/IEC 25000:2005.

Luego de ser realizado un estudio profundo de las principales herramientas para la ejecución de pruebas automáticas a sistemas operativos, se determinó que la herramienta más completa para llevar a cabo esta tarea es el Phoronix Test Suite. Esta herramienta contiene alrededor de 160 perfiles de prueba agrupados en 6 categorías que permiten comprobar distintos aspectos de la plataforma instalada. Además es código abierto, lo cual permite modificarlo de acuerdo a las necesidades del proyecto. Los resultados obtenidos por esta herramienta pueden ser visualizados en el ordenador o subidos a la página oficial donde pueden ser comparados con otras plataformas (privativas o no).

Fundamentación teórica de las pruebas de Software.

A pesar de todas estas ventajas la herramienta no contiene perfiles de pruebas que evalúen la seguridad del sistema frente a diversos tipos de ataques, por lo cual se sugiere el estudio y utilización de la herramienta Snort debido a que es una de las herramientas para pruebas de seguridad más completas que existen y es de código abierto.

Subproceso de pruebas automáticas de la Distribución GNU/Linux Nova.

En este capítulo se describe el subproceso de pruebas automáticas diseñado para la distribución GNU/Linux Nova, obtenido a partir de la norma NC ISO/IEC 9126 y de la utilización de la herramienta para pruebas automáticas Phoronix Test Suite. Este subproceso se define a partir de métricas para evaluar un grupo de características de calidad necesarias en esta distribución. Además es necesario definir algunos elementos para ampliar y generalizar el uso de la herramienta Phoronix Test Suite.

Indicadores de Calidad para Pruebas Automáticas.

Para comercializar los productos de software se hace necesario identificar cuando está apto para ser utilizado, por lo que los fabricantes deben estar muy atentos a los requerimientos del sistema. Es por esto que la génesis de todo proceso de calidad de software comienza con la elaboración y aprobación de los requisitos de calidad del producto, elaborados a partir de los requerimientos del sistema y la competencia del mercado.

Cuando los requisitos de calidad de software se definen, se relacionan con las características de calidad de software o sub-características que contribuyen al cumplimiento de dichos requisitos. Luego se especifican las métricas externas apropiadas y los intervalos aceptables para cuantificar los criterios de calidad que validan que el software satisface las necesidades del usuario.

Un requisito de calidad es la función de medición que provee una interpretación de una propiedad de la calidad del software y un valor objetivo de la medida de la calidad [30]. Los indicadores constituyen medidas indirectas de las características del software y son la herramienta principal para lograr la medición, constituyen la regla que define cuando el producto cumple con la calidad respecto a una determinada característica y cuando no [31]. Los requisitos de calidad determinan que características debe tener cuantitativamente el producto para ser de calidad, pero sin los indicadores asociados sería imposible determinar cuando un requisito de calidad es cumplido o no.

Requisitos de calidad de la Distribución GNU/Linux Nova.

La definición de los requisitos de calidad de Nova está determinada por la experiencia de los miembros más experimentados de Qalit, en consulta con el equipo de desarrollo. Estos requisitos serán utilizados para asociar a los indicadores de calidad y a las métricas correspondientes y así poder evaluarlos cuantitativamente. Los requisitos de calidad determinados son los siguientes:

1. Las aplicaciones instaladas deben ser funcionales.
2. Las aplicaciones instaladas deben cumplir con los requisitos.
3. Las opciones del sistema (menú principal, opciones del panel, opciones de energía, etc.) deben de devolver el resultado esperado.
4. El sistema no debe permitir la corrupción de los datos.
5. El sistema no debe permitir que usuarios no autorizados accedan a la información contenida.
6. El EE (Entorno de Escritorio) debe estar diseñado he implementado de acuerdo a los requerimientos.
7. El sistema debe de tener un mecanismo de recuperación ante fallos.
8. Obtener el menor tiempo posible de recuperación del sistema ante cualquier fallo ocurrido.
9. Obtener el menor tiempo posible de inactividad del sistema ante cualquier fallo ocurrido.
10. Obtener posibles puntos de restauración ante cualquier fallo ocurrido.
11. Confeccionar demos, manuales de usuarios y ayudas tanto online como en formato duro.
12. Buena descripción de los demos, manuales de usuarios y ayudas confeccionados.
13. Permitir la mayor personalización del sistema por el usuario.
14. Permitir un fácil acceso a todas las aplicaciones instaladas y a las opciones del sistema.
15. Cumplimiento de los requisitos de eficiencia definidos.
16. Obtener el menor tiempo posible de respuesta del sistema.
17. El sistema debe funcionar correctamente en las distintas condiciones especificadas en los requerimientos.
18. El sistema debe permitir el uso de varios dispositivos de hardware y protocolos de red además de los especificados.
19. Evitar que un fallo ocurrido se convierta en un fallo total del sistema.
20. Todas las aplicaciones instaladas, y las opciones del sistema deben ser probadas.

Subproceso de pruebas automáticas de la Distribución GNU/Linux Nova.

21. El sistema debe ser capaz de actualizarse sin la ocurrencia de fallos

Para determinar cuáles requisitos de los anteriormente descritos podrían ser evaluados automáticamente se procedió a realizar el siguiente análisis. Se asoció cada uno de los requisitos de calidad a la característica que más acorde este, de acuerdo a la NC ISO/IEC 9126-1 y posteriormente se comprobaron cuales de estas características podrían ser evaluadas automáticamente. (Ver imagen)

Subproceso de pruebas automáticas de la Distribución GNU/Linux Nova.

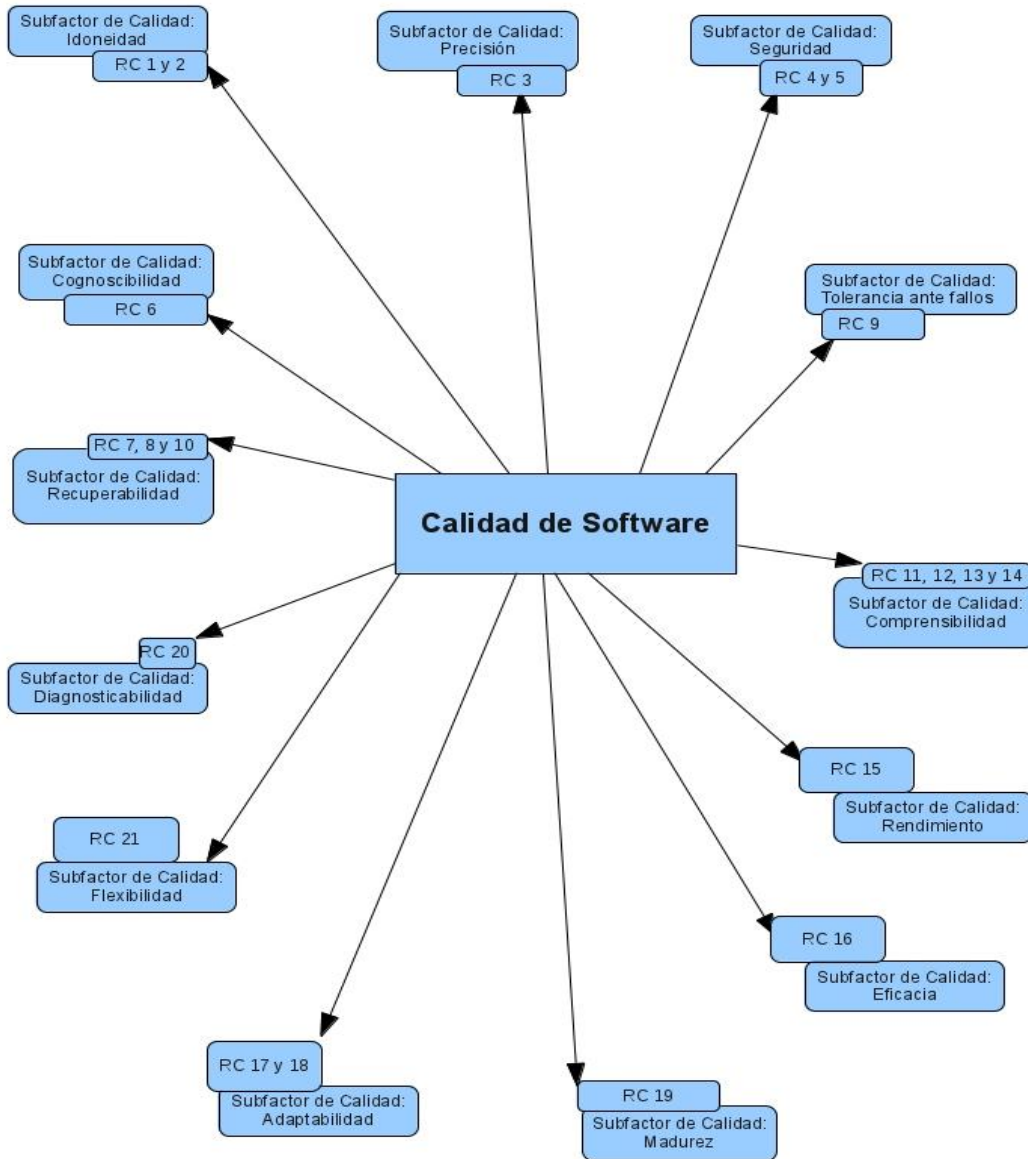


Imagen 1: Relación de requisitos de calidad con características de la NC ISO/IEC 9126-1

Subproceso de pruebas automáticas de la Distribución GNU/Linux Nova.

Una vez asociados los requisitos de calidad a las subcaracterísticas de acuerdo a la NC ISO/IEC 9126-1 fueron identificadas tres subcaracterísticas que podrían ser evaluadas automáticamente, basados en el estudio de las herramientas para la realización de pruebas automáticas:

- Seguridad: Esta subcaracterística puede ser evaluada de forma automática utilizando pruebas que intenten acceder de forma no autorizada a la información o al sistema.
- Precisión: Debido a que dentro de las opciones del sistema hay un gran número de funcionalidades, estas no pueden ser probadas de forma automática en su totalidad. No obstante pueden ser realizadas pruebas que midan la respuesta de determinadas opciones de forma automática, por ejemplo las funciones de energía (Apagar, Reiniciar, Hibernar, Suspender).
- Rendimiento: Esta subcaracterística puede ser medida en su gran mayoría por pruebas automáticas, aquí se incluyen pruebas al procesador, disco duro, tarjeta aceleradora de gráficos, entre otras.

Luego de definir cuáles de las subcaracterísticas pueden ser evaluadas automáticamente se determinó que el requisito de calidad número tres hace referencia a la correcta respuesta que debe devolver el sistema al seleccionar alguna de sus funcionalidades, lo cual puede ser probado automáticamente de forma parcial.

Los requisitos cuatro y cinco hacen referencia a pruebas de seguridad. Las cuales son difíciles de probar de forma manual, pues se necesita a un experto en seguridad con avanzados conocimientos para realizar un hackeo ético al sistema.

Por otro lado el requisito número quince hace referencia al rendimiento del sistema como producto, lo cual es factible probar de forma automática, pues realizadas de forma manual estas pruebas son absolutamente tediosas e inexactas.

La eficacia del sistema puede ser medida automáticamente, lo cual evitaría utilizar tiempo y personal calificado para realizar estas pruebas de forma manual. Existen pruebas que calculan la eficacia del sistema y pueden ser realizadas por un personal sin grandes conocimientos de este tipo de pruebas.

Métricas del Subproceso de Pruebas Automáticas

Conocidos los requisitos de calidad del producto a evaluar se hace necesario elaborar las métricas que serán utilizadas para medir cada una de las características que se encuentran reflejadas en estos. En esta investigación solo se definen métricas para los requisitos medibles automáticamente.

El requisito número tres describe la precisión de la respuesta de las opciones del sistema. Esta característica es importante en un sistema que debe cumplir con requerimientos de energía, por ejemplo el sistema debe ser capaz de ser suspendido, en este modo de energía el consumo se reduce a más de la mitad. Para ello el sistema debe ofrecer esta opción al usuario y responder de forma correcta.

Los requisitos de calidad cuatro y cinco hacen referencia a la seguridad del sistema. Esta subcaracterística es importante pues se hace indispensable desarrollar una distribución segura para su utilización tanto en estaciones de trabajo como en el hogar.

El requisito de calidad quince hace alusión al rendimiento, una característica de calidad muy importante en el contexto cubano. Debido a que son utilizados ordenadores antiguos y con bajas prestaciones no se puede desarrollar una distribución altamente consumidora de recursos.

Subproceso de pruebas automáticas de la Distribución GNU/Linux Nova.

Para evaluar estas características se plantean las siguientes métricas:

No	Nombre de la Métrica	La Métrica a medir	Método de Aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de Medida
1	Rendimiento	¿Cuántas tareas pueden realizarse con éxito en un período determinado de tiempo?	Realizar varias tareas concurrentes. Llevar un registro electrónico de cada intento.	$X=A/T$ A=Número de tareas completadas. T=Período de tiempo de observación.	$0 < X$ El más grande es el mejor	X= Contable / Tiempo A=Contable T=Tiempo
2	Utilización de los dispositivos de E/S	¿A qué velocidad el sistema puede realizar operaciones de E/S?	Con el sistema funcionando ejecutar pruebas de utilización de recursos. Registrar cada intento	$X=A/T$. A=Volumen de datos. T=Período de tiempo de E/S.	$0 < X$ El más grande es el mejor	X= Contable / Tiempo A=Contable T=Tiempo
3	Opciones de energía del sistema	¿Qué funcionalidades de energía se encuentran presentes en el sistema?	Con el sistema funcionando intentar llevar a cabo las operaciones de apagado, reseteado, hibernado y suspendido. Llevar un registro electrónico de cada intento	$X=1-(Papagado + Preseteado + Phibernado + Psuspendido) / (Nintentos)$ Papagado = Prueba de apagado. Preseteado = Prueba de reseteado. Phibernado = Prueba de	$0 < X < 1$ Mientras más pequeño mejor	X=Contable Papagado= Contable. Preseteado= Contable. Phibernado= Contable. Psuspendido=

Subproceso de pruebas automáticas de la Distribución GNU/Linux Nova.

				<p>hibernado.</p> <p>Psuspendido = Prueba de suspendido.</p> <p>Nintentos = cantidad de intentos realizados.</p> <p>En todas las pruebas el éxito es 1, el fracaso 0.</p>		<p>Contable.</p> <p>Nintentos=</p> <p>Contable.</p>
4	Capacidad de control de acceso	¿Puede ser controlado el acceso al sistema?	<p>Intentar acceder al sistema de forma ilícita.</p> <p>Llevar un registro electrónico de cada intento</p>	<p>$X=A/B$</p> <p>A=accesos ilícitos logrados.</p> <p>B=Total de intentos</p>	<p>$0 \leq X \leq 1$ Cerca de 0 mejor.</p>	<p>X=Contable.</p> <p>A=Contable.</p> <p>B=Contable.</p>
5	Prevención de la corrupción de datos	¿El sistema protege los datos eficazmente?	<p>Intentar corromper datos del ordenador.</p> <p>Llevar un registro electrónico de cada intento</p>	<p>$X= 1 - A/N$</p> <p>A=Número de veces en que son corrompidos los datos.</p> <p>N=Número de intentos de corromper los datos</p>	<p>$0 \leq X \leq 1$</p> <p>Cerca de 1 es mejor</p>	<p>X=Contable.</p> <p>A=Contable.</p> <p>N=Contable.</p>

Pruebas Automáticas del Phoronix Test Suite

La herramienta para pruebas automáticas Phoronix Test Suite agrupa a más de 160 pruebas individuales en seis categorías diferentes. La clasificación original de las pruebas del Phoronix Test Suite incluye: Sistema, Procesador, Disco (almacenamiento), Gráficos, Memoria RAM y Red. Esta clasificación no es factible para el probador porque las pruebas se encuentran repetidas en varias categorías lo cual entorpece el proceso de selección de las que se van a aplicar. Para evitar esto fueron agrupadas de acuerdo a la funcionalidad que evalúan, para una mayor comprensión:

- **Pruebas de compilación:** En este subgrupo se encuentran aquellas pruebas en las cuales se mide cuanto tiempo demora el sistema en construir un determinado software. En este se incluyen: build-imagemagick, build-apache, build-linux-kernel, build-mplayer, build-mysql y build-php.
- **Pruebas de compresión:** A este subgrupo pertenecen aquellas pruebas que miden el rendimiento del sistema al comprimir o descomprimir un determinado fichero utilizando variados algoritmos de compresión. Aquí se encuentran las pruebas: compress-7zip, compress-gzip, compress-lzma, compress-pbzip2 y unpack-linux.
- **Pruebas de codificación:** El tiempo que demora el sistema en codificar un archivo multimedia en un formato determinado, es la característica que define a las pruebas incluidas en este subgrupo. A este pertenecen: encode-ape, encode-flac, encode-mp3, encode-ogg, encode-wavpack, espeak, ffmpeg, mencoder y x.264.
- **Cálculo matemático:** El rendimiento del ordenador al procesar algoritmos matemáticos complejos es el punto de unión de estas pruebas de rendimiento. Entre estas se encuentran: bullet, byte, crafty, fhourstones, gmpbench, himeno, hmmer, java-scimark2, mafft, minion, mrbayes, n-queens, nero2d, scimark2, sudokut y systester.
- **Seguridad y criptografía:** Aquí se encuentra las pruebas que miden la capacidad del sistema de encriptar archivos así como de generar y romper claves. Entre estas pruebas se encuentran: bork, bcrypt, gnugp, jhon-the-reaper, Openssl.
- **Generación de gráficos:** Este subgrupo está compuesto por pruebas que miden la capacidad del sistema para generar gráficos en dos y tres dimensiones, así como recrear entornos que contengan iluminación y sombras. Aquí se encuentran: bwfirt, c-ray, dcraw, graphics-magick,

gluxmark, jgxbat, povray, smallpt, tachyon, ttsiod-renderer, specviewperf10, specviewperf9, trislam, wine-cloth, wine-domino, wine-fire2, wine-hdr, wine-metaballs, wine-vf2, wine-water, juliagpu, MandelbulbGPU, Mandelgpu, smallPT-gpu, sunflow.

- **Rendimiento y stress del procesador:** Estas pruebas recrean condiciones extremas de cálculo para el procesador, forzándolo al máximo. En este subgrupo están: npb, opstone-svd, opstone-vsp, stresscpu2, tscp, Apache, geekbench, pgbench, phpbench, pybench.
- **Utilización de recursos:** En esta categoría se encuentran las pruebas que miden la utilización de los recursos de almacenamiento tanto temporal como persistente del sistema, por ejemplo la memoria RAM, la cache, así como el disco duro. Dentro de este subgrupo se encuentran: cachebench, Aio-stress, blogbench, compilebench, dbench, fio, fs-mark, hdparm-read, iozone, postmark, sqlite, tiobench, ramspeed y stream.
- **Juegos de computadora:** En esta categoría se encuentran las pruebas que miden el rendimiento de los gráficos del ordenador al ejecutar juegos con alto nivel de detalle en sus interfaces. Aquí se encuentran: Doom3, et, etqw, etqw-demos, nexuiz, nexuiz-iqc, openarena, padman, prey, smoking-guns, supertuxcart, tremulous, unigine-heaven, unigine-sanctuary, Unigine-tropics, Urbanterror, ut2004-demo, vdrift, vdrift-fps-monitor, warsow, pprayer, xplane9 y xplane9-icq.
- **Rendimiento de servidor X.Org y varios motores gráficos:** Las pruebas que miden el rendimiento del servidor X.Org de las aplicaciones generadas con los motores gtk y qt4. En este subgrupo de pruebas se pueden encontrar: gtkperf, j2dbench, jxrendermark, Lightsmark, qgears2, render-bench.
- **Rendimiento del adaptador de red:** Aquí se encuentra la prueba que mide el rendimiento del adaptador de red del ordenador, network-loopback.
- **Opciones del sistema y energía:** Aquí se encuentran las pruebas que miden el comportamiento de algunas opciones de energía del sistema además de comprobar otras funcionalidades de la plataforma. En esta categoría están: battery-power-usage, compilance-ACPI, compilance-sensors e idle.

Esta forma de organización permite que el proceso de selección sea agilizado pues para seleccionar una prueba no es necesario estudiar las más de 160 pruebas individuales que realiza la herramienta. Solo se deben analizar algunas de las pocas que se encuentren dentro de la subcategoría que se ajuste al

Subproceso de pruebas automáticas de la Distribución GNU/Linux Nova.

requerimiento. De esta forma el proceso de selección de las pruebas del Phoronix Test Suite ha sido mejorado permitiendo un mayor aprovechamiento del tiempo destinado a la preparación de las pruebas.

La versión anterior del proceso de pruebas incluye pruebas automáticas en su desarrollo, con el propósito de evaluar de forma automática algunas características del producto desarrollado. Pero, al no utilizar métricas para determinar cuantitativamente la calidad del producto los resultados de las pruebas podían ser malentendidos. Es por esto que de acuerdo a lo indicado en la norma NC ISO/IEC 9126-1 se asocian métricas a las pruebas a realizar.

A la métrica Rendimiento por la característica de medir el aprovechamiento del sistema de la capacidad del procesador se le pueden asociar los grupos de pruebas: compresión, codificación, compilación, cálculo matemático, rendimiento y estrés del procesador, y juegos de computadora. Es preciso señalar que a pesar de que estos grupos contienen numerosas pruebas no existe la necesidad de realizar cada una de las pruebas que los componen, basta con realizar las pruebas recomendadas para los grupos especificados a continuación:

- Pruebas de compilación: build-imagemagick.
- Pruebas de cálculo matemático: java-scimark2.
- Pruebas de rendimiento y estrés del procesador: stresscpu2.
- Pruebas de juegos de computadoras: nexuiz.

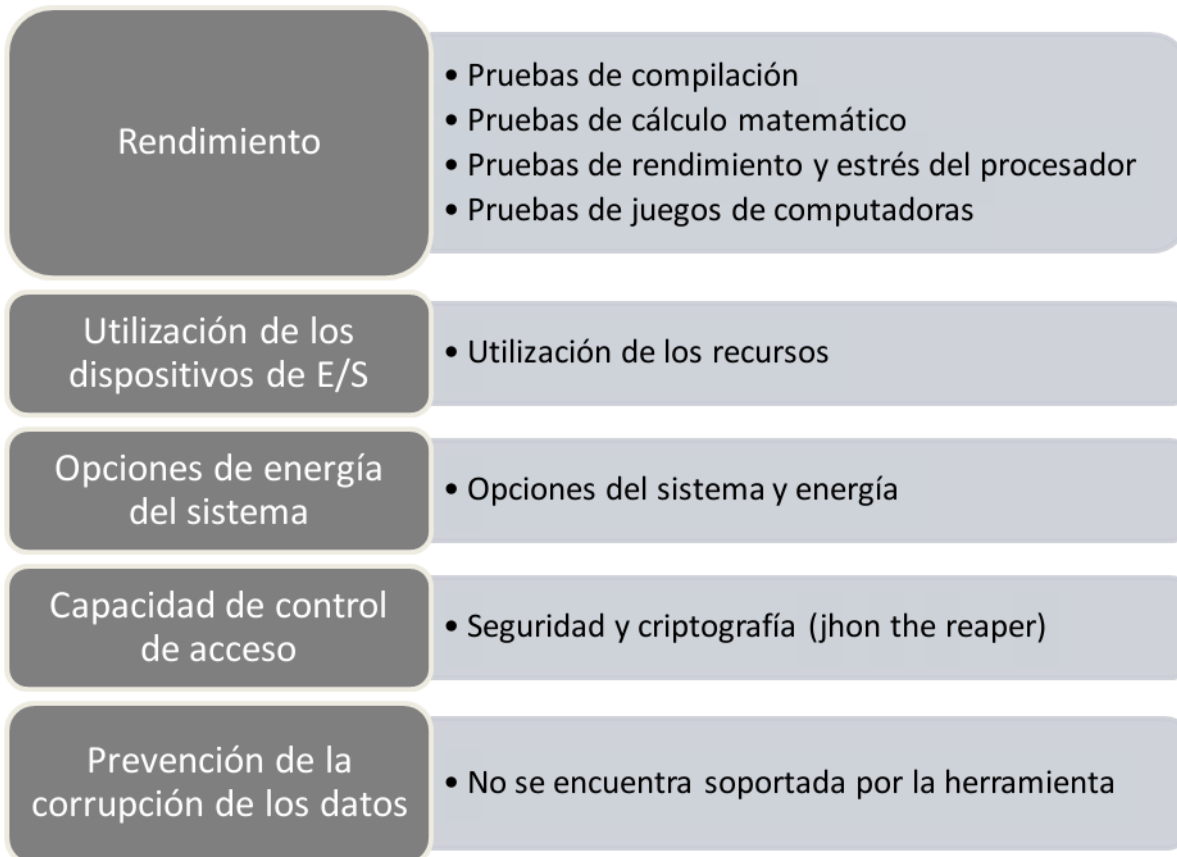
La métrica Utilización de los dispositivos de E/S tiene relacionadas las pruebas que componen el grupo de pruebas utilización de recursos. En el momento de seleccionar las pruebas hay que tener en cuenta que existen variados dispositivos de E/S en la computadora, razón por la cual se deben correr al menos una prueba por cada uno de ellos.

La métrica Opciones de energía del sistema tiene asociadas las pruebas que pertenecen al grupo Opciones del sistema y energía en el cual se encuentran pruebas que permiten evaluar las opciones del sistema de forma parcial, comprobando principalmente opciones del sistema relacionadas con el uso de la energía.

A la métrica Capacidad de control de acceso se le puede asociar la prueba "jhon the reaper" del grupo de pruebas Seguridad y criptografía, aunque en la descripción de esta prueba se especifica que solo funciona para determinar claves de acceso simples.

A la métrica de prevención de la corrupción de los datos no se le asocia ningún grupo de pruebas porque la herramienta propuesta no contiene pruebas que evalúen la misma. Además se puede apreciar que existen grupos de pruebas que no están asociados a ninguna métrica en específico debido a que no comprueban ninguno de los requisitos de calidad definidos, pero no se excluye su utilización en procesos de pruebas de productos específicos dentro de la distribución.

Asociación entre métricas y pruebas



Subproceso de Pruebas Automáticas

El subproceso de pruebas automáticas comienza a partir de la actividad de diseño de las pruebas, el alcance de este se reduce a los requerimientos que sean evaluables de forma automática. Las entradas de este subproceso están conformadas por los requisitos de la versión de la distribución en desarrollo y

los requisitos de calidad de la misma, para determinar en ambos casos cuales se pueden probar de forma automática. Además se considera una entrada el manual donde se caracterice la herramienta para pruebas automáticas a usar, debido a que en este es donde se describen las pruebas que permite hacer.

Las actividades principales del proceso de pruebas de la distribución donde se refleja este subproceso son:

- El diseño de las pruebas.
- Selección e implementación de pruebas automáticas.
- Ejecución de las pruebas.
- Evaluación de los resultados de las pruebas.

¿Cómo diseñar casos de prueba que se correspondan con los requerimientos de una prueba automática?

Para diseñar casos de pruebas que puedan ser realizados de forma automática se hace necesario conocer los requerimientos de calidad y los requerimientos del sistema asociados al producto de software que va a ser probado. A partir de la definición de estas listas de requerimientos se deben identificar aquellos que pudieran ser probados automáticamente, para lo cual solo deben tenerse en cuenta aquellos que no dependan de la interacción del sistema con un usuario. Generalmente todos los requerimientos asociados con rendimiento y utilización de los medios de almacenamiento pueden ser probados automáticamente.

En caso de que el sistema tenga algún requerimiento específico, depende del conocimiento del probador determinar que pruebas podrían realizarse de forma automática. En este aspecto influye el dominio de qué tipo de pruebas pueden realizar las herramientas para pruebas automáticas utilizadas en este proceso. Si se determina que el requerimiento puede ser evaluado por la herramienta, entonces se procede a confeccionar el caso de prueba. Para esto se deben seguir los siguientes pasos:

1. Se definen las condiciones de ejecución de la prueba, determinando el contexto necesario para la correcta ejecución de la misma. Por ejemplo: las dependencias de la aplicación a probar deben estar instaladas, el software para las pruebas automáticas debe estar instalado y las pruebas deben estar instaladas.

2. Se debe realizar una descripción del caso de prueba detallando que aspecto será medido por este. Ejemplo: En este caso de prueba se medirá el aprovechamiento del procesador por la distribución.
3. Se procede a elaborar los escenarios del caso de prueba los cuales contienen un identificador, una descripción y la respuesta del sistema. Ejemplo: El aprovechamiento del procesador se puede ver desde distintos puntos de vista, cada escenario correspondería a uno de estos. Uno de los elementos distintivos dentro de las pruebas de rendimiento del procesador son las pruebas de estrés, que implican la ejecución de forma concurrente de varias tareas que requieran el procesamiento de un gran volumen de datos. La respuesta del sistema para el ejemplo sería X volumen de datos / Y tiempo.

El diseño de las pruebas no incluye un flujo de operaciones, ni variables debido que las pruebas automáticas no requieren de interacción del usuario con el sistema. Una vez diseñadas las pruebas se debe obtener el documento del diseño de casos de pruebas que servirá como guía para el proceso. Debe actualizarse el cronograma en el documento Plan de pruebas para especificar cuando se realizaran las pruebas, quien será el responsable y cuáles son los recursos necesarios.

Selección e implementación de pruebas automáticas

A partir del documento de diseño de casos de prueba y del manual de la herramienta para pruebas automáticas a utilizar se procede a la selección de las pruebas de la herramienta que satisfacen las diseñadas. Para realizar esta selección se comparan los escenarios con las pruebas que ofrece la herramienta. En caso de que un escenario de pruebas no se ajuste a ninguna prueba específica de las ofrecidas por la herramienta existen dos opciones:

- si la herramienta permite añadirle nuevas pruebas, se procederá a implementar la misma de acuerdo a las características de la herramienta;
- en caso contrario se estudiará la posibilidad de utilizar otra herramienta que permita realizar este tipo de pruebas.

Esta investigación recomienda el uso de la herramienta Phoronix, debido a que es la más completa, aunque las pruebas de seguridad que ofrece no abarcan la totalidad de las características requeridas por Nova. Para este caso, en un primer momento se comparan las funcionalidades a probar con la clasificación mostrada anteriormente y luego se escogerían las pruebas recomendadas dentro de esa

categoría. La implementación de las pruebas en el caso de la herramienta Phoronix no es posible, pues esta solo acepta pruebas desarrolladas por los colaboradores reconocidos y no brinda facilidad alguna para desarrollarlas.

Debido a que la herramienta propuesta no logra comprobar todos los requisitos de calidad de seguridad se podría utilizar la herramienta Snort la cual permite el diseño de pruebas personalizadas y brinda facilidades para ello. La seguridad es una característica de la calidad que abarca muchas subcaracterísticas, por lo que en esta investigación no se ofrece detalles de la misma dejándose este tema para investigaciones posteriores.

Ejecución de las pruebas.

La ejecución parte de la selección de las pruebas automáticas de la(s) herramienta(s) y la disponibilidad de la(s) prueba(s), posteriormente se deben instalar todas las aplicaciones necesarias. Se recomienda, para obtener resultados fidedignos, que cada prueba se realice en ordenadores con una configuración de hardware similar. Se sugiere también mantener ejecutándose solo las aplicaciones necesarias en el momento de realizar la prueba, a menos que la prueba determine lo contrario. Luego de ejecutar las pruebas seleccionadas se obtiene un informe con los resultados, cuyo formato responde a cada herramienta.

En los casos de las pruebas de Phoronix la ejecución se reduce a ejecutar el Phoronix Test Suite, seleccionar la prueba a ejecutar y ordenar su ejecución. El informe resultante muestra un resumen de las principales características de hardware y de software del equipo donde se realizaron las pruebas, además de una descripción detallada de los resultados que incluye gráficos que permiten visualizarlos. Adjunto al informe resultante se encuentran las no conformidades de cada prueba realizada.

Evaluación de las pruebas

A partir de los informes de resultados de las pruebas realizadas es necesario entonces evaluar las pruebas. Es en este momento en el que las métricas adquieren un papel protagónico. Para hacerlo se tienen dos alternativas:

1. Comparando los resultados de las pruebas realizadas a la versión actual contra los resultados

Subproceso de pruebas automáticas de la Distribución GNU/Linux Nova.

obtenidos de otras plataformas (privativas o no).

2. Comparando los resultados obtenidos con un registro histórico de versiones anteriores de la misma plataforma.

Se recomienda que de ser posible se realicen ambas comparaciones porque al hacerlo se obtiene una idea del trabajo realizado respecto a otras distribuciones a nivel mundial. No obstante la comparación principal debe ser con el histórico de versiones anteriores por lo cual se debe llevar un registro electrónico del resultado de cada proceso de prueba realizado.

Al concluir el cálculo de las métricas se obtienen los elementos necesarios para el informe final del proceso de pruebas, que serían los siguientes:

- Resultados cuantitativos de las pruebas: Resumen del informe obtenido de la herramienta ajustado a los requisitos evaluados.
- Calidad de la versión actual de la distribución: Resultados del análisis comparativo con los resultados históricos de otras versiones del producto.
- Calidad del sistema operativo GNU/Linux Nova: Resultados del análisis comparativo con los resultados históricos de otros sistemas.
- Evaluación Final: Esta evaluación es cualitativa y hace alusión a si el sistema está listo para ser liberado.

En este capítulo han sido definidos los indicadores de calidad para evaluar una distribución y se determinaron cuales podrían ser medibles automáticamente. A partir de estos indicadores fue conformado el subproceso de pruebas automáticas para la distribución GNU/Linux Nova, el cual contempla la incorporación de los indicadores de calidad al proceso de evaluación y establece el orden en que debe ser desarrollado este subproceso.

Evaluación del Subproceso de Pruebas Automáticas

Obtener la máxima calidad de un producto de software es el objetivo primordial de todo proceso de pruebas. Para obtenerla se hace necesario que el proceso cuente con una buena organización y calidad, de forma tal que el producto pueda ser evaluado totalmente durante todo su desarrollo. Con la finalidad de entregar un producto de que logre satisfacer a los clientes se deben cumplir con los requerimientos convenidos de antemano. Esto garantiza que el producto sea liberado con la mínima cantidad de errores y que cumpla con las normas de calidad internacionales.

Es necesario evaluar de alguna manera la propuesta descrita en el capítulo anterior para comprobar que este subproceso contiene los elementos necesarios para mejorar la calidad de los productos que libera. En este capítulo se hace un análisis de los métodos que permiten hacer este tipo de evaluaciones, para seleccionar al menos uno y aplicarlo al subproceso de pruebas automáticas de la distribución GNU/Linux Nova.

Métodos de Evaluación.

Método de consulta a expertos (Delphi): Este trata de lograr un consenso de opiniones expresadas individualmente por un grupo de personas seleccionadas cuidadosamente como expertos calificados en torno al tema, por medio de la iteración sucesiva de un cuestionario retroalimentado de los resultados promedio de la ronda anterior, aplicando cálculos estadísticos [32].

Método de validación práctica: Consiste en la obtención, comparación y análisis de resultados obtenidos en el transcurso de la investigación al aplicar prácticamente el procedimiento en varios proyectos [33].

Método de grupo focal: Este método se basa en la selección de un conjunto de personas con abundantes conocimientos del tema. Deben ser expertos o especialistas, de distintos niveles y categorías. Estos se reúnen en un lugar a una hora determinada, donde se discute en forma de grupo-debate dirigido por los autores, lo que se quiere conocer sobre el procedimiento [34].

A partir de los métodos descritos anteriormente se selecciona el método Delphi, el método de validación práctica no es viable porque para utilizarlo sería necesario probar la propuesta en varios proyectos, lo cual

en este caso solo se aplica a Nova. El método de grupo focal se desestima debido al corto tiempo disponible para llevarlo a término.

Evaluación de la viabilidad de la propuesta sobre la base del criterio de expertos. El método Delphi.

Esta técnica proporciona el conocimiento del grupo de expertos sobre el tema que se ha escogido, permitiendo conocer la información de consenso en un grupo y es útil como herramienta exploratoria para el pronóstico tecnológico.

Consiste en la aplicación de un cuestionario a cada uno de los posibles expertos donde se mide su coeficiente de competencia obteniendo los que presentan aptitudes para dar respuesta a cada una de las interrogantes. Las conclusiones del análisis de las respuestas se traducen en un segundo cuestionario, que de nuevo se remite al grupo de expertos. [35]

El método suele dividirse en tres etapas o fases fundamentales:

1. Fase inicial: es donde ocurre la formulación del problema. Las preguntas deben ser precisas, cuantificables e independientes.
2. Fase exploratoria: es en esta fase donde ocurre la selección de los expertos con toda la serie de parámetros que la misma requiere.
3. Fase final: es donde ocurre el desarrollo práctico y la explotación de los resultados.

El método Delphi es caracterizado por:

- Anonimato: No es necesario que los expertos se conozcan, esto garantiza que puedan dar sus respuestas sin llegar a confrontaciones y además:
 - impide que el criterio de un experto del grupo sea influenciado por la reputación de otro de los miembros o por el peso de oponerse a la mayoría;
 - el experto puede defender sus argumentos con la tranquilidad de que en caso de estar equivocado, su equivocación no será conocida por los otros expertos.
- Iterativo: Pueden ser realizadas tantas iteraciones como sean necesarias para obtener datos concisos.
- Retroalimentación controlada: luego de cada ronda de preguntas son tabuladas las respuestas y son procesadas de forma tal que antes de la siguiente ronda pueden ser evaluados los resultados

de la anterior, así como las razones dadas por cada respuesta y su dispersión del promedio.

- Respuesta estadística del grupo: entre rondas de preguntas la información obtenida se procesa por medio de técnicas estadístico-matemáticas, que dotan al investigador de un instrumento objetivo y concreto en el cual pueden apoyarse para tomar una decisión final.

Selección de los Expertos

Los expertos son un factor fundamental al aplicar el método Delphi pues de su opinión depende la validez de la propuesta. Para seleccionar los expertos se tomó en cuenta las siguientes condiciones:

- Debe ser graduado de nivel superior.
- Debe estar vinculado a la investigación y al desarrollo de sistemas operativos.
- Debe poseer conocimiento acerca de la gestión de la calidad de software.
- Debe contar con tres años de experiencia como mínimo en el tema.
- No necesariamente un mismo especialista tiene que dominar todos los temas.

Luego a partir de estos criterios fueron seleccionados 5 expertos, de ellos:

- 5 son Ingenieros en Ciencias Informáticas.
- 4 de ellos cuentan con más de 5 años en el desarrollo de distribuciones de Software libre
- 2 de ellos tiene experiencia en pruebas de software.

Para determinar la competencia de los expertos seleccionados se elaboró un cuestionario en el cual cada experto valora sus conocimientos. Los resultados de esta encuesta se encuentran reflejados en la siguiente tabla.

Criterio/Evaluación	E1	E2	E3	E4	E5
Pruebas de Software	5	4	3	4	3
Calidad de Software	4	5	3	3	3
Desarrollo de Distribuciones de Software Libre	3	3	5	5	5
Pruebas Automáticas	4	3	3	3	4

Todos los expertos son especialistas en uno de los criterios definidos y dominan al menos un segundo criterio por lo cual se consideran aptos para llevar a cabo la validación de esta propuesta.

Conformación del cuestionario.

Para la conformación del cuestionario se utilizaron los requisitos de calidad definidos para su evaluación de forma automática. A partir de estos fueron elaborados los siguientes aspectos:

1. Evaluación de todas las características posibles.
2. Idoneidad de las métricas según los requisitos de calidad definidos.
3. Definición correcta de los requisitos de calidad a evaluar.
4. Estructura abarcadora para medir la calidad del producto
5. Descripción clara y concisa del proceso.
6. Idoneidad de la propuesta de acuerdo a las condiciones y proceso de desarrollo de Nova.

Una vez definidos estos aspectos se elaboraron las preguntas que fueron adicionadas a la encuesta y que tienen por objetivo evaluar tanto las métricas definidas, como el subproceso de pruebas automáticas propuesto. (Ver Anexo 3)

Desarrollo práctico y explotación de resultados.

En esta fase es donde se lleva a cabo el análisis estadístico de los datos que fueron obtenidos luego de aplicar la encuesta a los especialistas. Una vez aplicada la encuesta a los especialistas los datos obtenidos son procesados utilizando el factor de Kendall para determinar el grado de concordancia entre sus opiniones. Para obtener este valor fue utilizada la herramienta SSPS.

Los valores del coeficiente de Kendall deben oscilar entre cero y uno ($0 < k < 1$), los valores cercanos a uno representan concordancia entre los expertos. Luego de calculado obtenemos que $k = 0.67$, lo cual representa que los expertos coinciden un 67 por ciento.

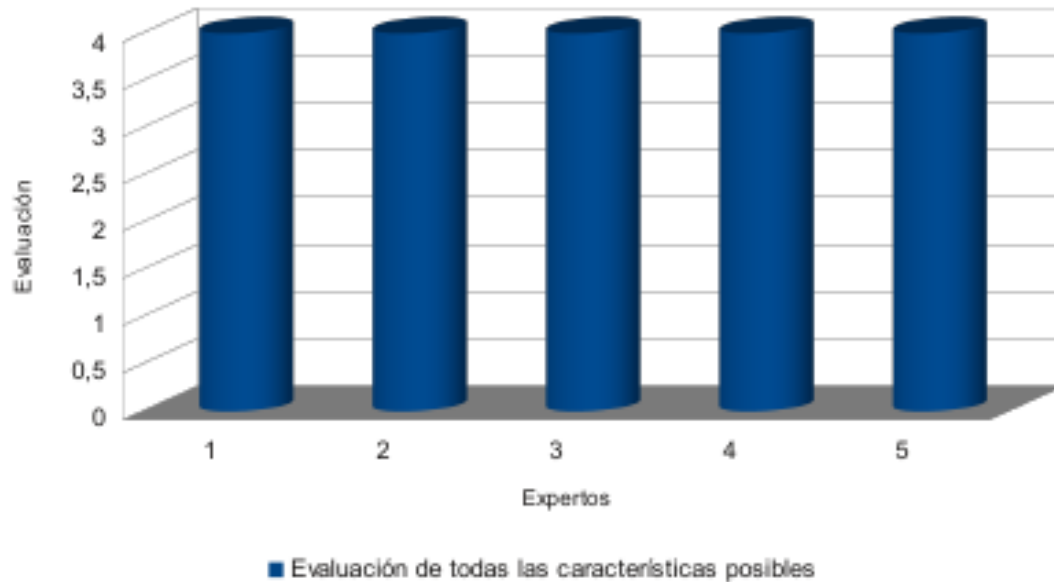
Test Statistics

N	5
Kendall's W(a)	0,668
Chi-Square	16,707
df	5
Asymp. Sig.	0,005

a Kendall's Coefficient of Concordance

Para comprobar la validez de la propuesta se busca el Chi-cuadrado tabulado en la tabla del percentil de la distribución Chi-cuadrado con un nivel de significación $\gamma-1$ grados de libertad, representado

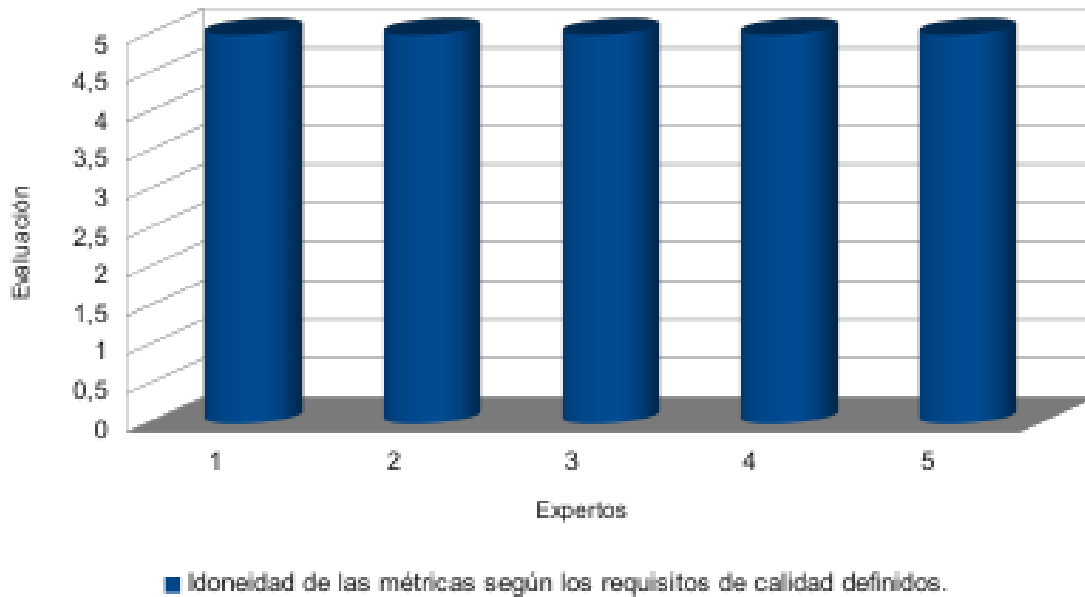
por $\chi_{tab}^2 = \chi_{\alpha; n-1}^2$ y se compara con el Chi-cuadrado calculado por la herramienta, si $\chi_{cal}^2 > \chi_{tab}^2$ entonces la propuesta es válida. Comparando Chi-cuadrado calculado =16.71 con Chi-cuadrado tabulado =1,145 se tiene que $\chi_{cal}^2 > \chi_{tab}^2$ por lo tanto podemos afirmar que los expertos coinciden en que la propuesta es



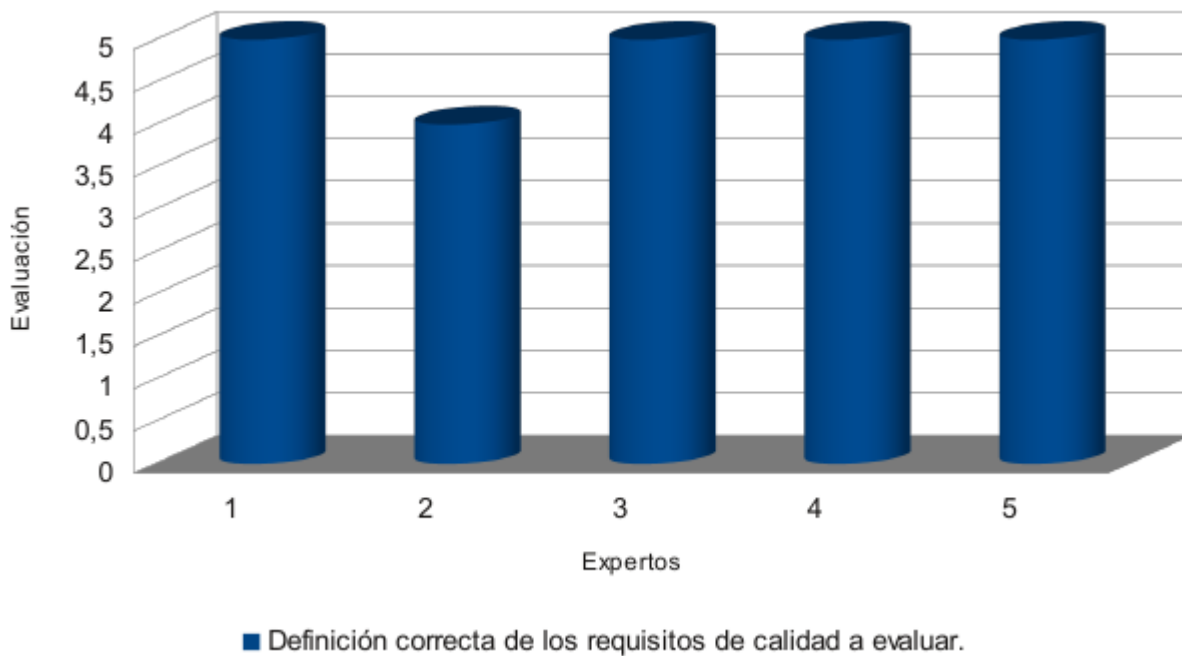
válida. A continuación se muestran gráficos que exponen los resultados de cada uno de los criterios establecidos.

La gráfica muestra que todos los expertos coinciden en que la guía abarca la mayoría de las características de la calidad necesarias para evaluar una distribución de software libre. Se obtuvo un 80% de aceptación por parte de los expertos sobre esta pregunta.

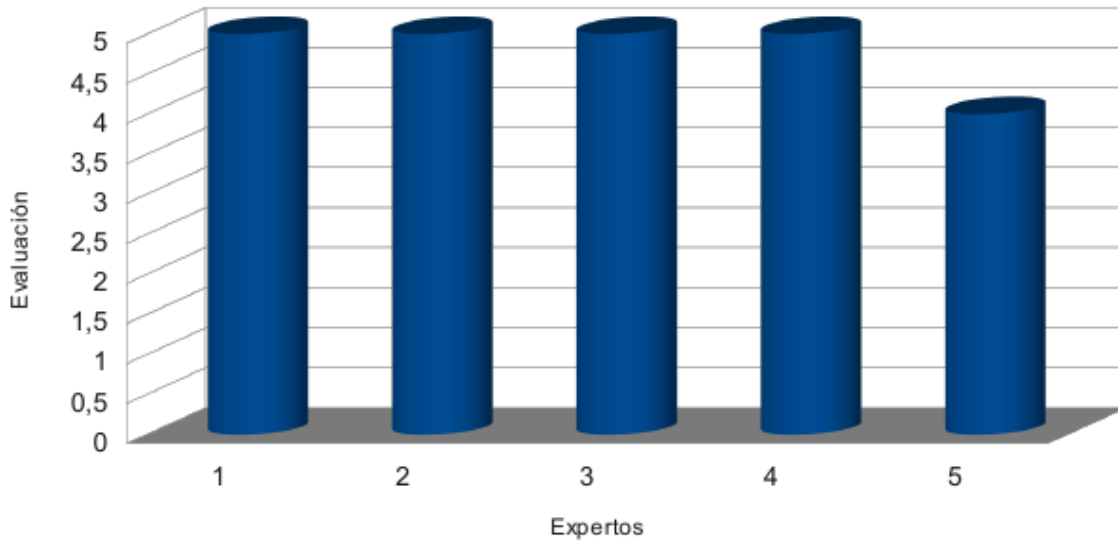
Evaluación del Subproceso de Pruebas Automáticas



En este gráfico muestra la idoneidad de las métricas definidas para evaluar los requisitos de calidad. En este caso los expertos coinciden 100% en que las métricas propuestas son idóneas para evaluarlos.



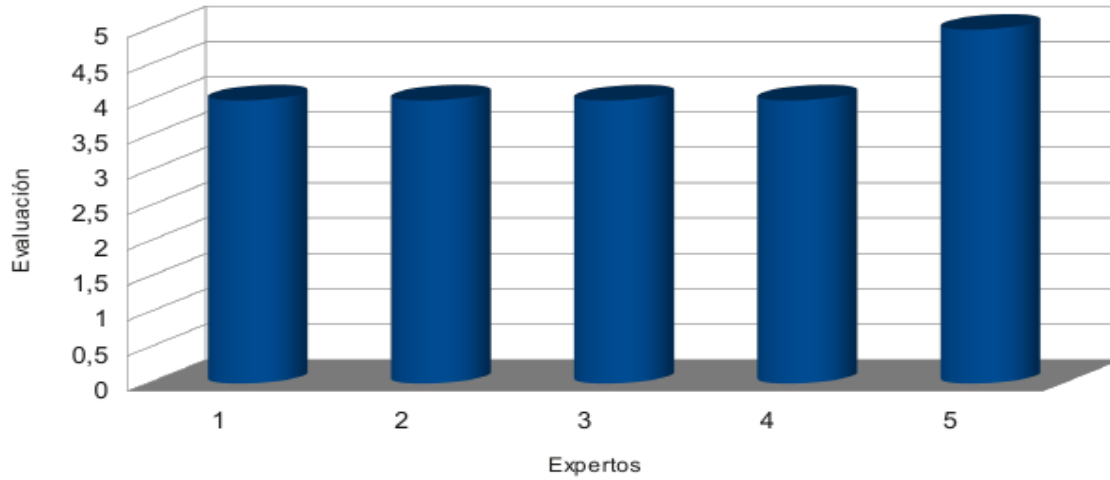
La gráfica anterior muestra la opinión de los expertos sobre la correcta definición de los requisitos de calidad a evaluar, coincidiendo en un 96% sus criterios.



■ Estructura abarcadora para medir la calidad del producto

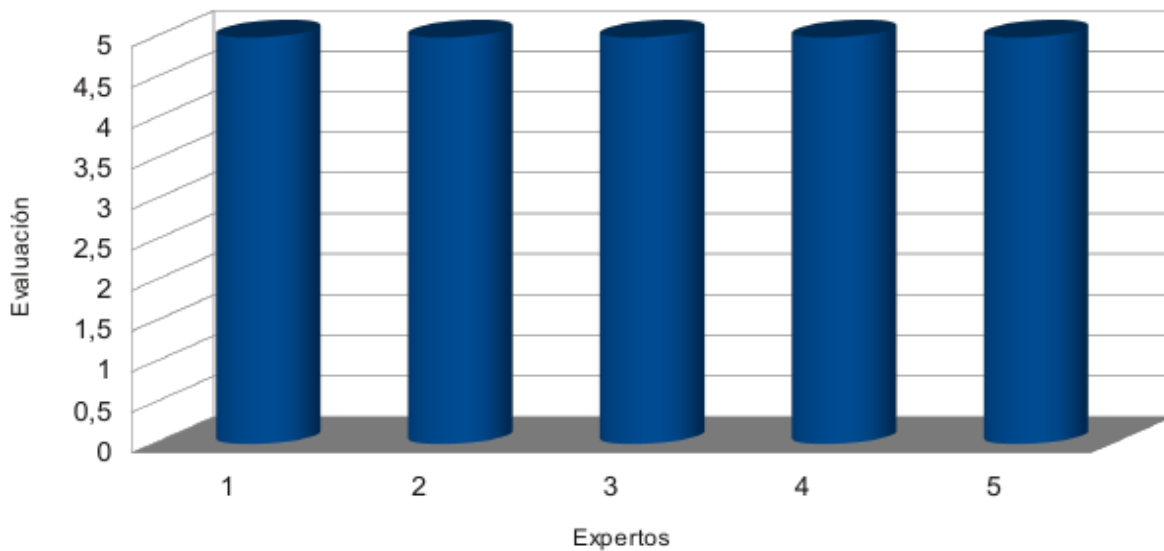
En la figura anterior se aprecia la evaluación asignada por los expertos a la estructura del subproceso descrito, haciendo énfasis en lo abarcadora que es para lograr medir la calidad del producto. En este caso los expertos coinciden en un 96% en sus opiniones.

En la gráfica siguiente se expresa el criterio de los expertos respecto a la claridad y consistencia de la descripción del proceso. Coincidiendo en un 84% sus evaluaciones.



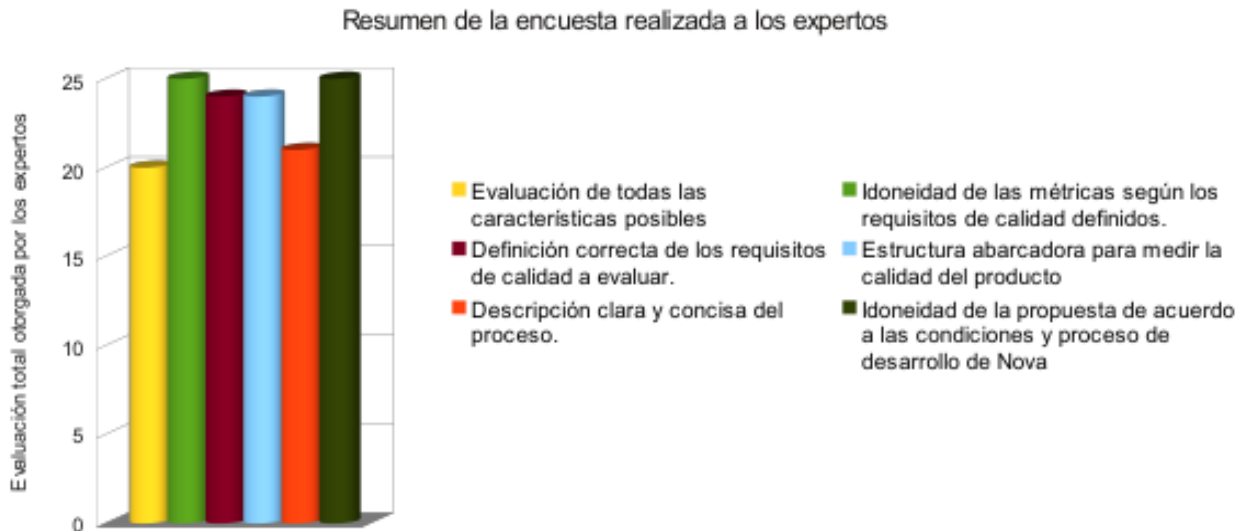
■ Descripción clara y concisa del proceso.

La figura siguiente representa el criterio de los expertos de acuerdo a la idoneidad del proceso propuesto atendiendo a las condiciones y procesos de desarrollo de la distribución de GNU/Linux Nova. En este caso los criterios coinciden en un 100%.



■ Idoneidad de la propuesta de acuerdo a las condiciones y proceso de desarrollo de Nova

El porcentaje de los especialistas tuvo siempre valores muy altos, en todos los casos supero la media, por lo tanto la propuesta ha sido validada utilizando el criterio de expertos. Los resultados generales se pueden apreciar en la siguiente gráfica:



Caso de estudio

Otra evaluación de la propuesta de esta investigación se propone practicarle las pruebas a una personalización de Nova conocida como NovaMedia. Esta personalización surge de la necesidad del proyecto FreeViUX de crear una personalización que contenga las principales aplicaciones para: edición de audio y video, creación de imágenes en dos y tres dimensiones, entre otros requerimientos y que sea lo suficientemente ligera como para correr en ordenadores antiguos.

Para comprender el grado de madurez de la herramienta será comparada en dos sentidos:

1. Realizar la comparación con una versión antigua de Nova para comprobar si esta personalización agrega alguna optimización en su modificación.
2. Comparar con una distribución GNU/Linux actual para tener una perspectiva de cómo se encuentra esta personalización con respecto a otras distribuciones en función del rendimiento.

Ejecución de la propuesta.

Con el propósito de comprobar si la opinión de los expertos es la acertada se procedió a realizar una muestra de algunas pruebas para determinar la efectividad de los indicadores y las métricas definidas para este subproceso.

En aras de cumplir este objetivo se definieron las siguientes actividades:

1. Diseño de los casos de prueba
 - Definición de los requisitos de calidad a evaluar así como de los indicadores de la versión a evaluar.
 - Definición de los criterios de evaluación y las métricas a calcular.
 - Diseño de los casos de prueba atendiendo a los requisitos.
2. Planificación de las pruebas
 - Realizar el cronograma de las pruebas.
3. Ejecución de los casos de pruebas
 - Son publicados los resultados de las pruebas.
4. Evaluación de los casos de pruebas
 - Se calculan las métricas definidas para obtener un resultado cuantitativo del cumplimiento de cada indicador escogido.
 - Se obtiene el informe de evaluación atendiendo a cada indicador de calidad.

Para esta demostración fue seleccionado el requisito de calidad rendimiento y a partir de este se escogieron varias pruebas. El criterio de selección de las pruebas cumple con dos requisitos: disponibilidad física de la prueba e importancia de la misma.

Del grupo de pruebas asociados al rendimiento y estrés del procesador fueron escogidas las pruebas java-scimarck2 y c-ray, siendo ambas adecuadas porque permiten poner a prueba la capacidad de procesamiento del sistema. Pertenecientes al grupo de pruebas utilización de recursos fueron seleccionadas las pruebas iozone, ramspeed y tiobench, pues permiten probar el rendimiento del disco duro, de la memoria RAM y del sistema de ficheros de forma exhaustiva.

No existe un plan de pruebas definido para la realización de pruebas a esta personalización, debido a que es una aplicación parcial de la propuesta a un caso de estudio, para su evaluación. Luego de diseñados los casos de pruebas fue elaborado un cronograma en el cual se detalla quien realiza cada prueba, que recursos están asignados para su realización y en qué momento debe realizarse.

Cronograma de ejecución de las pruebas

No.	Tarea	Fecha	Responsable	Participante	Observaciones
1	Ejecutar los casos de prueba 1 y 2.	08/06/2012	Yusdel P Delgado	Morejón	Ordenador con NovaMedia instalado, PTS ¹⁰ y las pruebas a ejecutar instaladas.
2	Evaluar el resultado obtenido en los casos de prueba.	08/06/2012	Yusdel P Delgado	Morejón	

Para llevar a cabo una comparación abarcadora fueron realizadas pruebas de forma adicional a Nova 2011 y Debian testing, de esta forma se puede formar una opinión basándose en los resultados de estas pruebas. Los casos de pruebas diseñados para esta tarea se pueden apreciar en el Anexo 5.

Luego de realizadas las pruebas utilizando la herramienta se obtiene los siguientes resultados:

Resultados de las pruebas Automáticas

Pruebas/Distribuciones	NovaMedia	Nova 2011	Debian
C-ray	503.13 s	1208.50 s	1369.26 s
Java-scimarck2	426.29 Mflops	280.33 Mflops	147.45 Mflops
lozone (prueba de escritura)	81.53 Mb/s	51.22 Mb/s	55.68 Mb/s
Ramspeed	2701.67 Mb/s	2515.05 Mb/s	1471.56 Mb/s
Tiobench (escritura aleatoria)	3.68 Mb/s	3.86 Mb/s	4.29 Mb/s

Obtenidos estos resultados se hace necesario aplicar las métricas con el propósito de entender estos resultados. En el caso de la primera prueba los resultados están expresados en función del tiempo, razón por la cual se aplica una relación inversa en la fórmula $X=A/T$ donde considerando que la prueba se realiza utilizando la misma cantidad de datos se establece entonces una relación inversamente

¹⁰ PTS, Phoronix Test Suite.

proporcional a la velocidad, a menor tiempo mayor velocidad. En el resto de las pruebas la fórmula de las métricas es similar por lo que el mayor valor es el que determina el éxito en las pruebas.

A partir de los resultados obtenidos en las pruebas se puede concluir que la distribución NovaMedia respecto a la capacidad de procesamiento se encuentra a la vanguardia, pues los resultados de las pruebas C-ray y Java-Scimarck2, luego de procesados utilizando métricas, se aprecia que prácticamente han duplicado los resultados de sus competidores foráneos o no. Respecto a la métrica de utilización de los recursos los resultados muestran que NovaMedia se encuentra en una posición ventajosa pues domina dos de las tres pruebas realizadas evidenciándose su calidad.

A partir de estos resultados se puede concluir que NovaMedia cumple con el requisito de calidad rendimiento de la distribución Nova de forma general. De forma particular se puede destacar que esta personalización muestra avances en la optimización del procesamiento de algoritmos matemáticos y en la utilización de los recursos.

Al concluir la realización de este capítulo se puede apreciar como se ha comprobado el valor de la propuesta a partir de la utilización del método del experto y de la aplicación parcial de la misma en un caso de estudio. Los expertos determinaron que la propuesta es acertada con un 67% de concordancia en sus opiniones, lo cual muestra su aceptación. La aplicación de la propuesta en el caso de estudio proporcionó una valoración extensiva de la personalización evaluada, permitiendo determinar su competencia de acuerdo a los requisitos de calidad definidos con respecto a versiones anteriores de la misma distribución, además de compararlo con respecto a otras distribuciones.

Conclusiones

Al concluir esta investigación se puede afirmar que los objetivos propuestos fueron cumplidos de forma satisfactoria, respondiendo a una problemática real del proyecto Nova.

- Se han definido los indicadores de calidad medibles automáticamente en sistemas operativos los cuales, unidos a la utilización de la herramienta para pruebas automáticas Phoronix Test Suite permiten una evaluación más abarcadora de la calidad del producto Nova.
- El subproceso obtenido como resultado de esta investigación cumple con normas de calidad de reconocimiento internacional. Posibilita la evaluación eficaz de los productos Nova a través de la utilización de indicadores y métricas de calidad.

Recomendaciones

La presente investigación obtiene como resultado un subproceso de pruebas automáticas, el cual se recomienda:

- Sea puesto en práctica dentro de los procesos de aseguramiento de la calidad del proyecto Nova, contribuyendo con el proceso de pruebas para obtener productos con una mejor calidad.
- Profundizar el estudio de las herramientas para pruebas automáticas de seguridad, en aras de aumentar la exigencia de los requisitos de calidad, referentes a la seguridad, en los productos Nova.

Referencia Bibliográfica

1. Vismar Santos Hernández, La Industria del software. Estudio a nivel Global y América Latina. Consultado en mayo 2012. Disponible en [<http://www.eumed.net/coursecon/ecolat/la/09/vsh.htm>]
2. Forodebate desde el Sitio del Minrex, disponible en [http://www.google.com.cu/url?sa=t&rct=j&q=minrex+cuba+software+libre+transcripcion+foro&source=web&cd=1&ved=0CCUQFjAA&url=http%3A%2F%2Fwww.cubaminrex.cu%2FDocumentos%2FTranscripci%25C3%25B3n%2520del%2520Foro%2520Debate.doc&ei=L6t5T6SDIciAgwetreHZBQ&usg=AFQjCNHFki42prP2LSjKtJtruB_ctQyMGg&cad=rja].
3. Machín Castillo, Jorge Luis. *Descripción del proceso de construcción del sistema operativo base de la distribución cubana de GNU/Linux Nova*. Universidad de las Ciencias Informáticas. Cuba. 2010.
4. Pablo Pazos Gutiérrez. Ingeniería en sistemas de software, 2010. Consultado en: abril 2012. Disponible en: [ingenieria_de_Software.blogspot.com].
5. Yuridia Azucena Hernández. *Pruebas: Una guía de valoración de sistemas de software durante el ciclo de vida*. Centro de investigación científica y de educación superior de enseñanza. California, 2002. 236.
6. BS 7925-1 disponible en [http://www.testingstandards.co.uk/bs_7925-1.htm].
7. Software Testing Glossary of Terms, 2008. Consultado en: mayo 2012. Disponible en [http://www.vietnamesetestingboard.org/zbxe/?document_srl=46096].
8. BS 7925-2. Consultado en: mayo 2012. Disponible en [http://www.testingstandards.co.uk/bs_7925-2.htm].
9. IEEE 829:2008. Consultado en: mayo 2012. Disponible en [<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4578383&isnumber=4578382>].
10. IEEE 1008:1987. Consultado en: mayo 2012. Disponible en [<http://standards.ieee.org/findstds/standard/1008-1987.html>].
11. IEEE 730:2002. Consultado en: mayo 2012. Disponible en [<http://standards.ieee.org/findstds/standard/730-2002.html>].

12. Perry, D. y Kaiser, G. Adequate testing y object-oriented programming. Testing Object-Oriented Software. IEEE Computer Society, 1998. pp 11-17.
13. NC ISO/IEC 9126-1:2005. Consultado en: mayo 2012. Disponible en [<http://calisoft.uci.cu/tmp/documentos/normas/iso/NC-ISO-IEC%209126-1.pdf>].
14. ISO/IEC 9126-2:2003. Consultado en: mayo 2012. Disponible en [http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22750].
15. ISO/IEC 9126-3:2003. Consultado en: mayo 2012. Disponible en [http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22891].
16. ISO/IEC 9126-4:2004. Consultado en: mayo 2012. Disponible en [http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39752].
17. ISO/IEC 14598-1:1999. Consultado en: mayo 2012. Disponible en [http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=24902].
18. ISO/IEC 14598-2:2000. Consultado en: mayo 2012. Disponible en [http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=24903].
19. ISO/IEC 14598-3:2000. Consultado en: mayo 2012. Disponible en [http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=24904].
20. ISO/IEC 14598-4:1999. Consultado en: mayo 2012. Disponible en [http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=24905].
21. ISO/IEC 14598-5:1998. Consultado en: mayo 2012. Disponible en [http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=24906].
22. ISO/IEC 14598-6:2001. Consultado en: mayo 2012. Disponible en [http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=24907].
23. ISO/IEC 25000:2005 disponible en [http://www.iso.org/iso/catalogue_detail.htm?csnumber=35683].
24. Tenable.Nessus, 2002. Consultado en: mayo 2012. Disponible en [<http://www.tenable.com/products/nessus>].
25. Trac.Dogtail, 2006. Consultado en: octubre 2011. Disponible en [<https://fedorahosted.org/dogtail/>].
26. Michael Larabel. Phoronix Test Suite, 2008. Consultado en: octubre 2011. Disponible en [<http://www.phoronix-test-suite.com/>].

Bibliografía Consultada

1. ISO/IEC 62:1996 sustituida por ISO/IEC 17021:2011 disponible en [http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=56676].
2. Lorena Monteagudo Suarez. *Proceso de pruebas de la distribución GNU/Linux Nova. La Habana. 2011.* Universidad de las ciencias informáticas. Cuba. 2011. Disponible en [http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_0503/1/TD_0503.pdf].
3. Lic. María Isabel de Arquer. NTP 401: *Fiabilidad humana: métodos de cuantificación, juicio de expertos.* [España]: Centro Nacional de condiciones de trabajo, Enero 2010.
4. Góngora, Asnier. *Catálogo automatizado de métricas de calidad para evaluar los productos de las pruebas.* Universidad de las ciencias informáticas. Cuba. Diciembre, 2011.

Anexos

1.-Manual de usuario de la herramienta Phoronix Test Suite:

Instalación del Phoronix Test Suite

Para realizar la instalación del Phoronix Test-suites se realizan los siguientes pasos:

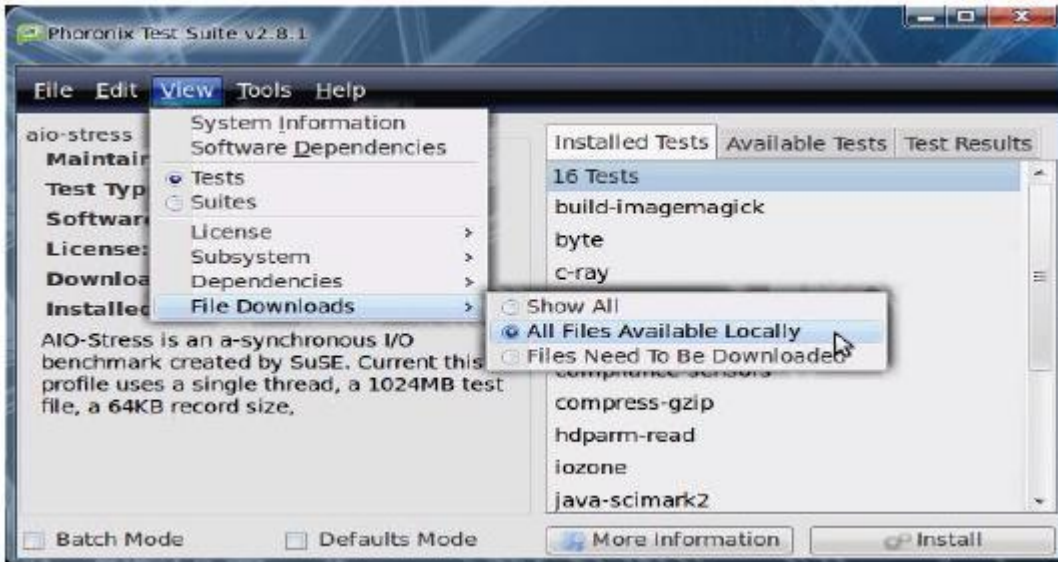
1. Abrir el menú de nova e ir a Añadir/Quitar software
2. Buscar Phoronix Test-suites mediante la pestaña Buscar
3. Instalar los paquetes phoronix-test-suite_2.8.1_all, php-gtk_2.0.0-1nova2_i386 y pecl-cairo_0.2.0-beta-1_i386.

Instalación de las pruebas de Phoronix

Esta aplicación nos brinda un conjunto de más de 160 pruebas individuales así como 57 suites o conjuntos de pruebas especializadas, la aplicación por defecto contiene los enlaces de los sitios oficiales donde se encuentran los paquetes de pruebas, que en la mayoría de los casos han sido elaborados por los propios fabricantes del software a probar.

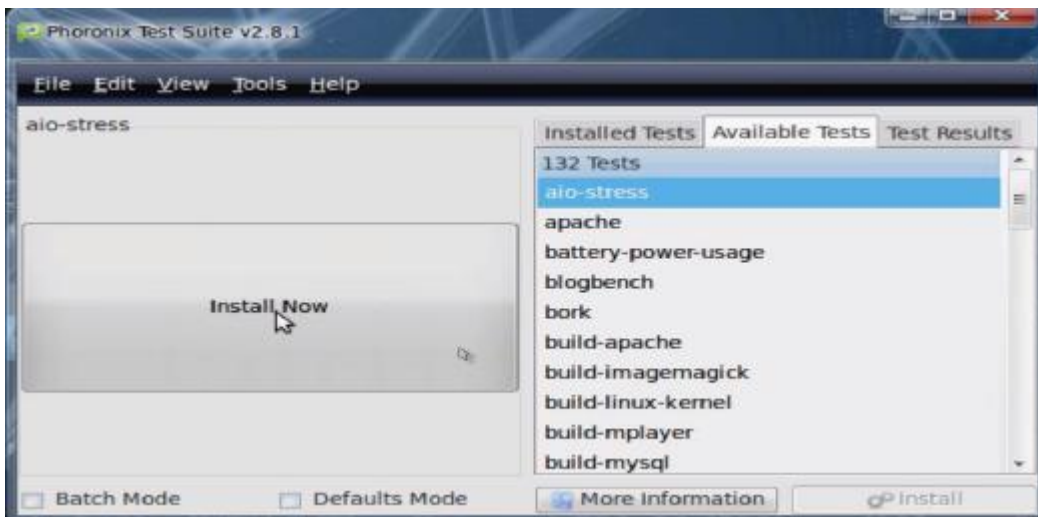
Para instalar estas pruebas una vez que han sido descargadas, se sigue un procedimiento más o menos estándar:

1. Copiar el archivo nombrado “nombre”.tar-gz o “nombre”.zip en la carpeta /home/usuario/.phoronix-test-suite/download-cache.
2. Descompactar el paquete de la prueba.
3. Se marca la opción “All Files Available Locally” en el menú “View” submenú “File Downloads”



Se selecciona la pestaña “available tests” en la interfaz principal.

Se escoge la prueba a instalar y se presiona el botón “Install” y luego el botón “Install Now”.



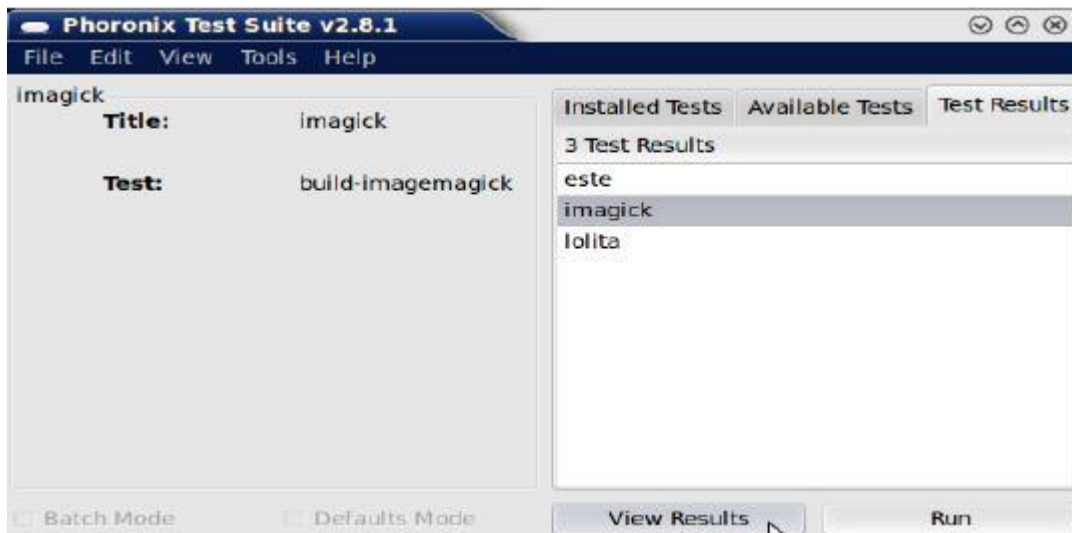
Las pruebas pueden ejecutarse desde la ventana principal de la aplicación

Se selecciona la pestaña “installed tests”.

Se escoge la prueba deseada y se presiona el botón “Run”.



En el cuadro de dialogo que se muestra a continuación se coloca el nombre con el cual se guardarán los resultados de la prueba.



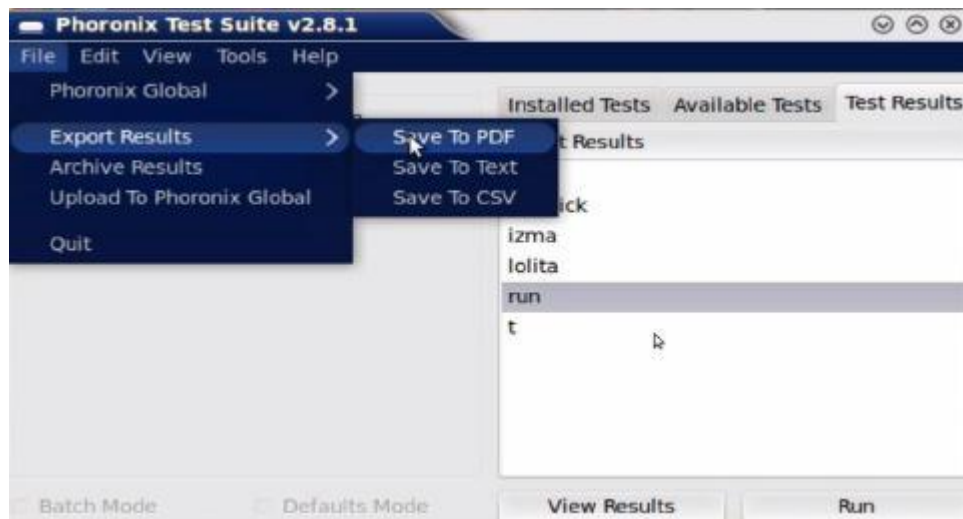
Visualización de los resultados de las pruebas

Seleccionar la pestaña “Test Results”.

Se selecciona la prueba de la cual se desean ver los resultados.

Se presiona el botón “View Results”.

Estos resultados se muestran en el navegador Web por defecto del sistema, pero si el sistema tiene instalado el plugin “PHP-FPDF” una vez seleccionado el resultado de la prueba se puede exportar en formato PDF.



La herramienta para pruebas automáticas Phoronix Test Suite agrupa a más de 160 pruebas individuales en seis categorías diferentes. La clasificación original de las pruebas del Phoronix Test Suite incluye: Sistema, Procesador, Disco (almacenamiento), Gráficos, Memoria RAM y Red. Esta clasificación no es factible para el probador porque las pruebas se encuentran repetidas en varias categorías lo cual entorpece el proceso de selección de las pruebas a aplicar. Para evitar esto fueron agrupadas de acuerdo a la funcionalidad que evalúan, para una mayor comprensión de las pruebas:

- **Pruebas de compilación:** En este subgrupo se encuentran aquellas pruebas en las cuales se mide cuanto tiempo demora el sistema en construir un determinado software. En este grupo se incluyen: build-imagemagick, build-apache, build-linux-kernel, build-mplayer, build-mysql y build-php.
- **Pruebas de compresión:** A este subgrupo pertenecen aquellas pruebas que miden el rendimiento del sistema al comprimir o descomprimir un determinado fichero utilizando variados algoritmos de compresión. Aquí se encuentran las pruebas: compress-7zip, compress-gzip, compress-lzma, compress-pbzip2 y unpack-linux.
- **Pruebas de codificación:** El tiempo que demora el sistema en codificar un archivo multimedia en un formato determinado es la característica que define a las pruebas incluidas en este subgrupo. A este pertenecen: encode-ape, encode-flac, encode-mp3, encode-ogg, encode-wavpack, espeak, ffmpeg, mencoder y x.264.
- **Cálculo matemático:** El rendimiento del ordenador al procesar algoritmos matemáticos complejos es el punto de unión de estas pruebas de rendimiento. Entre estas se encuentran: bullet, byte,

crafty, fhourstones, gmpbench, himeno, hmer, java-scimark2, mafft, minion, mrbayes, n-queens, nero2d, scimark2, sudokut y systester.

- **Seguridad y criptografía:** Aquí se encuentran las pruebas que miden la capacidad del sistema de encriptar archivos así como de generar y romper claves. Entre estas pruebas se encuentran: bork, bcrypt, gnugp, jhon-the-reaper, Openssl.
- **Generación de gráficos:** Este subgrupo está compuesto por pruebas que miden la capacidad del sistema para generar gráficos en dos y tres dimensiones, así como recrear entornos que contengan iluminación y sombras. Aquí se encuentran: bwfirt, c-ray, dcraw, graphics-magick, gluxmark, jgxbat, povray, smallpt, tachyon, ttsiod-renderer, specviewperf10, specviewperf9, trislam, wine-cloth, wine-domino, wine-fire2, wine-hdr, wine-metaballs, wine-vf2, wine-water, juliagpu, MandelbulbGPU, Mandelgpu, smallPT-gpu, sunflow.
- **Rendimiento y stress del procesador:** Estas pruebas recrean condiciones extremas de cálculo para el procesador, forzándolo al máximo. En este subgrupo están: npb, opstone-svd, opstone-vsp, stresscpu2, tscp, Apache, geekbench, pgbench, phpbench, pybench.
- **Utilización de recursos:** En esta categoría se encuentran las pruebas que miden la utilización de los recursos de almacenamiento tanto temporal como persistente del sistema, por ejemplo la memoria RAM, la cache, así como el disco duro. Dentro de este subgrupo se encuentran: cachebench, Aio-stress, blogbench, compilebench, dbench, fio, fs-mark, hdparm-read, iozone, postmark, sqlite, tiobench, ramspeed y stream.
- **Juegos de computadora:** En esta categoría se encuentran las pruebas que miden el rendimiento de los gráficos del ordenador al ejecutar juegos con alto nivel de detalle en sus interfaces. Aquí se encuentran: Doom3, et, etqw, etqw-demos, nexuiz, nexuiz-iqc, openarena, padman, prey, smoking-guns, supertuxcart, tremulous, unigine-heaven, unigine-sanctuary, Unigine-tropics, Urbanterror, ut2004-demo, vdrift, vdrift-fps-monitor, warsow, ppraycer, xplane9 y xplane9-icq.
- **Rendimiento de servidor X.Org y varios motores gráficos:** Las pruebas que miden el rendimiento del servidor X.Org de las aplicaciones generadas con los motores gtk y qt4. En este subgrupo de pruebas se pueden encontrar: gtkperf, j2dbench, jxrendermark, Lightsmark, qgears2, render-bench.
- **Rendimiento del adaptador de red:** Aquí se encuentra la prueba que mide el rendimiento del adaptador de red del ordenador, network-loopback.
- **Opciones del sistema y energía:** Aquí se encuentran las pruebas que miden el comportamiento de algunas opciones de energía del sistema además de comprobar otras funcionalidades de la

plataforma. En esta categoría están: battery-power-usage, compliance-ACPI, compliance-sensors e idle.

2.-Encuesta realizada a los expertos para determinar sus conocimientos referentes a la propuesta.

El propósito de esta encuesta es determinar los conocimientos que usted posee acerca de ciertas materias. Marque con una X de acuerdo a su criterio. **Nota:** C1 es mayor que C5.

No.	Criterio	Autoevaluación del Experto				
		C1	C2	C3	C4	C5
1	¿Cómo considera usted sus conocimientos acerca de las Pruebas de Software?					
2	¿Qué calificación usted le asigna a la Calidad de Software?					
3	¿Cómo evalúa usted su experiencia en el Desarrollo de Distribuciones de Software Libre?					
4	¿En el campo de Pruebas Automáticas de Software cómo se autoevalúa usted?					

3.-Encuesta realizada a los expertos para evaluar la propuesta.

El subproceso de pruebas automáticas de la distribución GNU/Linux Nova propuesto se basa en la utilización de métricas de calidad para ofrecer una valoración cuantitativa de los resultados obtenidos y asegurar la valía de los mismos. Propone la utilización de la herramienta Phoronix aunque para las pruebas de seguridad se necesitaría otra herramienta.

Los requisitos de calidad a medir automáticamente:

1. Las opciones de energía del sistema deben devolver el resultado esperado.
2. La distribución no debe permitir la corrupción de los datos.
3. La distribución no debe permitir que usuarios o aplicaciones no autorizadas accedan a la

información.

4. Se deben cumplir los requisitos de eficiencia establecidos con el usuario final.

Las métricas para medir estos requisitos:

Rendimiento	¿Cuántas tareas pueden realizarse con éxito en un período determinado de tiempo?	Realizar varias tareas concurrentes. Llevar un registro electrónico de cada intento.
Utilización de los dispositivos de E/S	¿A qué velocidad el sistema puede realizar operaciones de E/S?	Con el sistema funcionando ejecutar pruebas de utilización de recursos. Registrar cada intento
Opciones de energía del sistema	¿Qué funcionalidades de energía se encuentran presentes en el sistema?	Con el sistema funcionando intentar llevar a cabo las operaciones de apagado, reseteado, hibernado y suspendido. Llevar un registro electrónico de cada intento
Capacidad de control de acceso	¿Puede ser controlado el acceso al sistema?	Intentar acceder al sistema de forma ilícita. Llevar un registro electrónico de cada intento
Prevención de la corrupción de datos	¿El sistema protege los datos eficazmente?	Intentar corromper datos del ordenador. Llevar un registro electrónico de cada intento

En la siguiente tabla responda las preguntas, asignándole un criterio a cada una según convenga. **Nota:**

C1 es mayor que C5.

No.	Preguntas	Criterio del Experto				
		C1	C2	C3	C4	C5
1	¿Usted considera que las métricas definidas evalúan todos los aspectos de la calidad posibles?					
2	¿Considera usted que las métricas elaboradas son idóneas para medir los requisitos de calidad definidos?					
3	¿De acuerdo a su criterio, la definición de las métricas está acorde a los requerimientos de calidad que evalúa?					
4	¿Cree usted que la estructura del subproceso de pruebas automáticas abarca todos los pasos para medir la calidad del producto?					
5	¿Usted considera que la descripción del proceso es clara y concisa?					
6	¿Considera usted que de acuerdo a las condiciones y al proceso de desarrollo de Nova este subproceso es idóneo?					

¿Usted considera que la propuesta mejora el proceso de pruebas de la distribución GNU/Linux Nova?

4.-Casos de pruebas para la personalización NovaMedia.

CP1. Rendimiento del procesador

Condiciones de ejecución

La herramienta y la prueba seleccionada deben estar instaladas en el ordenador.

Escenario	Descripción	Respuesta del sistema	Flujo central
-----------	-------------	-----------------------	---------------

EC 1.1 Algoritmos matemáticos	Se comprueba la capacidad del sistema de resolver complejos algoritmos matemáticos	<p>1.1 La aplicación muestra las opciones de la prueba.</p> <p>2.1 La aplicación ejecuta la prueba seleccionada.</p> <p>2.2 La aplicación muestra los resultados de la prueba</p>	<p>1. Seleccionar la prueba "java-scimarc2".</p> <p>2. Ejecutar la prueba seleccionada.</p>
EC 1.2 Generación de gráficos 2D.	Se renderiza una imagen 2D de alta definición.	<p>1.1 La aplicación muestra las opciones de la prueba.</p> <p>2.1 La aplicación ejecuta la prueba seleccionada.</p> <p>2.2 La aplicación muestra los resultados de la prueba</p>	<p>1. Seleccionar la prueba "c-ray".</p> <p>2. Ejecutar la prueba seleccionada.</p>

CP2. Utilización de los dispositivos de E/S

Condiciones de ejecución

La herramienta y la prueba seleccionada deben estar instaladas en el ordenador.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 2.1 Rendimiento del disco duro	Se comprueba el rendimiento del disco duro realizando operaciones de E/S combinadas	<p>1.1 La aplicación muestra las opciones de la prueba.</p> <p>2.1 La aplicación ejecuta la prueba seleccionada.</p> <p>2.2 La aplicación muestra los resultados de la prueba</p>	<p>1. Seleccionar la prueba "iozone".</p> <p>2. Ejecutar la prueba seleccionada.</p>
EC 2.2 Rendimiento del disco duro.	Se comprueba el rendimiento de la RAM	<p>1.1 La aplicación muestra las opciones de la prueba.</p> <p>2.1 La aplicación ejecuta la prueba seleccionada.</p>	<p>1. Seleccionar la prueba "ramspeed".</p> <p>2. Ejecutar la prueba seleccionada.</p>

		2.2 La aplicación muestra los resultados de la prueba	
EC 2.3 Rendimiento del disco duro y del sistema de ficheros.	Es comprobado de forma extensiva el rendimiento, tanto del disco duro como del sistema de ficheros	<p>1.1 La aplicación muestra las opciones de la prueba.</p> <p>2.1 La aplicación ejecuta la prueba seleccionada.</p> <p>2.2 La aplicación muestra los resultados de la prueba</p>	<p>1. Seleccionar la prueba "tiobench".</p> <p>2. Ejecutar la prueba seleccionada.</p>