

*Universidad de las Ciencias Informáticas*

*Facultad 1*



*Título: “Proceso de pruebas de la distribución GNU/Linux Nova”*

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

*Autores:*

*Lorena Monteagudo Suárez*

*Tutores:*

*Mónica María Albo Castro*

*Ciudad de La Habana*

*Mayo 2011*

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 \_\_\_\_\_

---

Firma del Autor

---

Firma del Autor

---

Firma del Tutor

---

Firma del Tutor

*Frase*

*“La calidad nunca es un accidente;  
Siempre es el resultado de un esfuerzo de la inteligencia.”*

*John Ruskin*

## ***Agradecimientos:***

*A todos los profesores que me enseñaron lo que sé por su amor y dedicación.*

*A todas las personas que de una forma u otra colaboraron en el desarrollo de esta investigación y en mi preparación como futura Ingeniera en Ciencias Informáticas.*

*Le agradezco a mi familia por tanto amor y cariño. En especial a mi mamita, la mejor del mundo, sin su apoyo incondicional no hubiese logrado seguir adelante, sin su sacrificio y abnegación, sus sabios consejos, por compartir alegrías y tristezas, por sus llamadas que me daban fuerzas para continuar cuando lo creía todo perdido.*

*Agradezco a la Revolución y a Fidel por haberme brindado la posibilidad de estudiar en esta Universidad, donde me formé profesionalmente y donde encontré los mejores amigos, que me han dado su amor y cariño, su apoyo incondicional y dedicación.*

*A mis padres Sonia y Julio por estar tan orgullosos de mí, por admirarme y apoyarme en cada paso que doy. A la memoria de mis queridos abuelos y mi primo Franklyn, que estoy segura estaría muy orgulloso y feliz si me pudiera ver hoy.*

*Agradecerles a mis tías, principalmente a mi tía Estela por impulsarme hacia el camino correcto para poder graduarme como Ingeniera. A mi abuelita, por quererme tanto y estar siempre pendiente de mis estudios y mi tesis.*

*A mi hermanito, que lo quiero con la vida y a mis primas y primos por todo el apoyo, la atención y preocupación por mis estudios.*

*A todas mis amistades durante estos 5 años principalmente a mi mama de la UCI Susana y mi papa Alfredo y a mi tía Mylen, que tanto me han ayudado y apoyado para salir adelante.*

*A mis tutoras, oponente y tribunal que nos han ayudado durante la realización de la tesis, dedicando parte de su tiempo a pesar de contar con una agenda bastante apretada.*

*A todas aquellas personas que de una forma u otra han colaborado con el desarrollo de esta investigación.*

## *Dedicatoria*

### *A mis padres*

*por su esfuerzo incansable y paciencia infinita en darme lo mejor de ellos:  
educación, experiencias, regaño y sobre todo, un amor incondicional*

### *A mi familia*

*por sus mimos y amor, por esa preocupación y desvelo impregnados  
de un cariño sin límites*

### *A mis Amigos*

*Por quererme así de grande, por enseñarme a ser mejor y a confiar siempre en  
mi, nunca los defraudaré*

### *A mi hermano*

*que lo quiero con todo mi corazón*

## *Resumen*

En el proceso de desarrollo de software, las probabilidades de cometer diferentes errores son bastante elevadas en cada una de sus etapas. En la actualidad es de gran importancia verificar que el software que va a liberarse tenga la mayor calidad posible, lo que implica que debe cumplir con todas las exigencias y perspectivas del cliente, así como el producto final debe poseer la menor cantidad de errores. Para llevar a cabo esta verificación, el software pasará por un proceso de pruebas, pues las mismas son el elemento principal para medir la calidad de un producto software que se entregará a un cliente específico o se lanzará al mercado internacional.

El presente trabajo de diploma se ha encaminado a asegurar en gran medida la calidad de software verificando que cumple con los requerimientos funcionales a través de la realización de un proceso de pruebas de software para la distribución Cubana de software libre GNU/Linux Nova, con el objetivo de obtener la mayor calidad del producto final y la satisfacción del cliente.

En el transcurso del mismo se realiza un estudio de los procesos de pruebas de las diferentes distribuciones de software libre del mundo y los distintos tipos de pruebas y modelos de desarrollo que existen actualmente, del que se hizo una selección para una mejor estructura y organización del proceso de pruebas desarrollado en Nova. También se propone la utilización de la herramienta de pruebas automáticas Phoronix Test Suite que es de gran ayuda para los desarrolladores pues les permite comparar el rendimiento de distintas versiones de Nova, otros sistemas operativos y distribuciones GNU/Linux permitiéndoles conocer las mejoras en cuanto a rendimiento que esta versión brinda y aportando datos que sirvan como base para estudiar el comportamiento de futuras versiones.

**Palabras claves:** Phoronix Test Suite, Nova, software, rendimiento.

ÍNDICE

ÍNDICE .....	7
INTRODUCCIÓN .....	8
<b>CAPÍTULO 1: “FUNDAMENTACIÓN TEÓRICA DE LAS PRUEBAS DE SOFTWARE” .....</b>	<b>12</b>
1.1 PRUEBAS DE SOFTWARE.....	12
1.1.1 <i>Propiedades de las pruebas</i> .....	13
1.1.2 <i>Niveles de prueba del software</i> .....	15
1.1.3 <i>Métodos de Prueba</i> .....	17
1.2 ENFOQUE DE LOS MODELOS DE DESARROLLO DE SOFTWARE.....	18
1.3 PROCESOS DE PRUEBAS DE DISTRIBUCIONES GNU/LINUX .....	20
1.3.1 <i>Proceso de pruebas de la distribución GNU/Linux, Ubuntu [15].</i> .....	20
1.3.2 <i>Proceso de pruebas de la distribución GNU/Linux Nova [14].</i> .....	22
1.4 HERRAMIENTAS PARA UTILIZAR LAS PRUEBAS AUTOMÁTICAS .....	22
1.4.1 <i>Dogtail [16].</i> .....	23
1.4.2 <i>Phoronix Test Suite [17].</i> .....	23
1.4.3 <i>Mago [18].</i> .....	24
<b>CAPÍTULO 2: “PROPUESTA DE PROCESO DE PRUEBAS PARA LA DISTRIBUCIÓN CUBANA DE SOFTWARE LIBRE GNU/LINUX NOVA” .....</b>	<b>25</b>
2.1 PROCESO DE PRUEBAS PARA LA DISTRIBUCIÓN GNU/LINUX NOVA.....	25
2.2 INSTALACIÓN Y PUESTA EN MARCHA DEL PHORONIX TEST SUITE .....	32
<b>CAPÍTULO 3: “VALIDACIÓN DEL PROCESO DE PRUEBAS DE LA DISTRIBUCIÓN CUBANA DE SOFTWARE LIBRE GNU/LINUX NOVA” .....</b>	<b>36</b>
3.1 MÉTODOS DE VALIDACIÓN .....	36
3.2 SELECCIÓN DEL MÉTODO DE EVALUACIÓN .....	37
3.2.1 <i>Método de validación experimental.</i> .....	37
3.2.2 <i>Método de Recopilación de la Información [21].</i> .....	44
<b>CONCLUSIONES GENERALES .....</b>	<b>48</b>
<b>RECOMENDACIONES.....</b>	<b>49</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>50</b>
<b>BIBLIOGRAFÍA CONSULTADA.....</b>	<b>52</b>
<b>ANEXOS .....</b>	<b>54</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>62</b>

## Introducción

El creciente desarrollo de las tecnologías y las ciencias informáticas se inserta cada vez más en los procesos de la vida diaria de la sociedad actual. Esto es posible en gran medida debido al desarrollo de los sistemas operativos que son "el software que permite la interacción de las personas con las tecnologías".

Cuba, a pesar de la situación geopolítica, ha ido incluyendo en la medida de las posibilidades estos avances tecnológicos en la sociedad. Durante los primeros pasos en esta nueva era de la informática se utilizaron sistemas operativos privativos, principalmente los pertenecientes a la familia Microsoft Windows. Debido a que la industria del software depende del sistema operativo sobre el que se desarrolle, la industria cubana ha tenido que incurrir en gastos adicionales, al tener que pagar patentes de software para poder comercializar los productos desarrollados sobre las plataformas privativas.

El bloqueo económico que mantiene EEUU a Cuba le impide comprar o actualizar diferentes productos como Windows (Software Propietario), uno de los sistemas operativos más populares del planeta y por esto se ha utilizado sin licencia y por tanto sin recibir ningún soporte, ayuda o beneficio legal. Como consecuencia no es posible exportar libremente software cubano desarrollado sobre plataforma Windows, por esta razón se comenzó a valorar la migración hacia el software libre.

En el año 2005 se define en Cuba una política de migración hacia el Software Libre con el propósito de lograr la independencia tecnológica y como parte de esta se comenzaron a introducir los sistemas operativos libres en Cuba.

Entre las ventajas que ofrece el software libre se encuentra que muchas de estas distribuciones poseen bajo costo de adquisición y muchas veces no se tienen que pagar costosas licencias para un determinado software ni pagar el propio software. Otra ventaja es que se pueden mejorar aplicaciones hechas por otros desarrolladores; lo cual lleva a que se creen aplicaciones superiores porque no hay que comenzar a desarrollarlas desde cero.

El movimiento del software libre se comenzó a desarrollar a partir de que en 1991 el proyecto GNU libera su primer sistema operativo libre denominado GNU/Linux conformado por un kernel llamado Linux desarrollado por Linus Torvalds y las herramientas propias que el proyecto GNU había implementado hasta el momento. Estos sistemas operativos se construyen a partir de un conjunto de aplicaciones, bibliotecas y herramientas de programación, además de un programa para alojar recursos e interactuar con el hardware, denominado núcleo [1].

El desarrollo y evolución propia de este sistema operativo libre y de código abierto llevó al surgimiento de las conocidas distribuciones de Linux o GNU/Linux. Una distribución de Linux no es más que un

conjunto de programas seleccionados específicamente y que corren sobre un kernel Linux [2].

La Universidad de las Ciencias Informáticas (UCI) ha desempeñado un papel protagónico en la migración al software libre, especialmente el proyecto Nova de la actual Facultad 1 creado en el año 2005. El mismo se encuentra enfocado en el desarrollo de tres líneas fundamentales: la encargada de generar el sistema base de la distribución en general y de la versión para escritorio, la implantación de clientes ligeros y la distribución para servidores. Además incluye otros desarrollos que aporten nuevas funcionalidades a la distribución y la creación de un entorno de escritorio propio.

Los fundadores de este proyecto se centraron en crear una distribución de software libre que pudiera sustituir los sistemas propietarios que se usan en las empresas cubanas. Este sistema operativo se conoce en la actualidad como: distribución cubana de software libre GNU/Linux Nova, desarrollada para entornos de escritorio orientado a usuarios inexpertos que hayan experimentado un proceso de migración a software libre. Se especializa en implementar personalizaciones, en el marco del software libre y en la informática una personalización no es más que modificar un software de acuerdo a los gustos y necesidades del usuario o cliente específico.

A nivel internacional cada distribución de software libre, antes de ser liberada, debe pasar por un proceso de pruebas que validen la calidad de la misma y que le permitan a sus desarrolladores controlar distintos aspectos técnicos como rendimiento, seguridad, y otros. Generalmente estos procesos son, en su mayoría, tediosos por naturaleza y complicados, necesiéndose para realizarlos de un personal ampliamente calificado y con una vasta experiencia en este campo.

Dentro del mencionado proceso de pruebas se incluyen un conjunto importante de pruebas al sistema, así como también a las aplicaciones que correrán sobre él. Debido a que Nova es un proyecto joven, en lo que a calidad respecta solo se le ha realizado un ciclo de pruebas en el año 2009, el cual se hizo de forma manual y por un personal totalmente inexperto. Esta situación provocó que las pruebas no se realizaran con la calidad requerida ni abarcaran todos los aspectos a comprobar, lo que trajo como consecuencia que la calidad de los productos liberados no haya sido la esperada.

Por lo anterior esta investigación se plantea como **problema a resolver**: La distribución GNU/Linux Nova se libera con poca calidad debido a que el proceso de pruebas no abarca todos los aspectos requeridos.

El **objeto estudio** del presente trabajo de diploma son los procesos de pruebas de las distribuciones GNU/Linux, donde se define como **campo de acción**: el proceso de pruebas de la distribución cubana de software libre GNU/Linux Nova.

Partiendo de esta temática el **objetivo general** es definir un proceso de pruebas que permita evaluar

todos los aspectos de la distribución GNU/Linux Nova e incluya pruebas automáticas para incrementar su calidad.

Y para alcanzarlo se han propuesto como tareas las siguientes:

- Investigar la evolución y tendencias de los procesos de pruebas de las distribuciones libres para conocer el comportamiento histórico de las mismas.
- Investigar las herramientas que automatizan las pruebas de los sistemas operativos para conocer las potencialidades de las mismas y determinar la más ventajosa.
- Redefinir el proceso de pruebas de la distribución GNU/Linux Nova para ajustarlo al uso de la herramienta seleccionada.
- Construir un manual que contenga las indicaciones para el uso de la herramienta dentro del proceso de pruebas de la distribución GNU/Linux.
- Validar el proceso y manual propuestos para determinar las ventajas y desventajas del mismo.

Con la realización del presente trabajo investigativo, se defiende la idea que con la implantación de un proceso de pruebas que evalúe todos los aspectos de una distribución GNU/Linux se puede elevar la calidad de los productos que se obtengan en el proyecto Nova.

Como Métodos Teóricos para esta investigación se utilizaron:

1. Analítico - Sintético: Durante la investigación se analizaron documentos, páginas de Internet y otros artículos de donde se extraen los elementos más importantes relacionados con las pruebas automáticas de software para los sistemas operativos libres. Además se realizaron resúmenes y valoraciones de las características más relevantes relacionadas con las aplicaciones que automatizan estas pruebas.
2. Inductivo - Deductivo: Según el estudio general de diferentes pruebas automáticas se propuso una aplicación para hacer más efectivas dichas pruebas.

Como Métodos Empíricos para esta investigación se utilizaron:

- Experimental: Se han realizado las correspondientes pruebas automáticas al sistema operativo Nova experimentando con una determinada aplicación.

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

Capítulo #1: Fundamentación teórica de pruebas de software: aquí se realiza un estudio de las aplicaciones y los procesos que existen para realizar pruebas a sistemas operativos libres, enunciándose los conceptos básicos necesarios para comprender el funcionamiento de dichas

aplicaciones y procesos.

Capítulo #2: Proceso de pruebas para la distribución GNU/Linux Nova: Realizar un proceso de pruebas para la distribución GNU/Linux, además de explicar la forma de utilizar la herramienta incluida en esta propuesta.

Capítulo #3: Validación de proceso de pruebas: Validación de la efectividad de este proceso, así como, su impacto en la calidad de los productos de la distribución cubana GNU/Linux Nova.

### **Capítulo 1: “Fundamentación teórica de las pruebas de software”**

El proceso de desarrollo de software incluye varias etapas entre ellas la de pruebas, la cual permite medir la calidad de los productos obtenidos previo a su liberación. En los últimos años, las pruebas de software han ido tomando un lugar importante en el denominado ciclo de desarrollo de software, principalmente debido a los altos costos que conlleva corregir errores en un software que ha completado su desarrollo. Además de las posibles pérdidas de los clientes porque el producto liberado no contiene la calidad esperada. Estas son las razones fundamentales que evidencian la importancia de manejar de manera cuidadosa el proceso de pruebas de todo desarrollo de software.

A continuación se detallan las características e importancia de las pruebas dentro del desarrollo de software, así como una breve descripción y comparación de varios modelos de desarrollo de software en cuanto al enfoque que le dan a las pruebas dentro del proceso en general. Se analizan y comparan los procesos de pruebas de otras distribuciones libres y por último se valoran algunas herramientas para la realización de pruebas automáticas. Estos últimos aspectos con el fin de encontrar un punto de referencia para el proceso de pruebas de la distribución GNU/Linux Nova y una herramienta que eleve la calidad y eficiencia del mismo.

#### **1.1 Pruebas de Software**

Todo software requiere un proceso de desarrollo con el propósito de realizar una implementación eficaz y eficiente de un producto que reúna los requisitos pedidos por el cliente. Un proceso de desarrollo de software es un conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto. Las mencionadas actividades se agrupan en varias etapas que generalmente son: Análisis, Diseño, Implementación, Pruebas y Mantenimiento, en las cuales, la salida de cada una representa la entrada de la siguiente [3].

En el caso de la etapa de pruebas, que es la que está relacionada con esta investigación, se hacen principalmente un conjunto de comprobaciones en función de los requerimientos definidos. Aunque también se pueden incluir pruebas técnicas no visibles para el usuario final, que una vez que son superadas pueden permitir la entrega del software finalizado. En fin, consiste en verificar que el software realiza correctamente las tareas indicadas en las especificaciones.

En esta etapa es necesario aplicar técnicas de prueba las cuales consisten en probar por separado cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo. Se considera una buena práctica que las pruebas sean efectuadas por alguien distinto al desarrollador que la programó, idealmente un equipo de pruebas; sin perjuicio de lo anterior el programador debe hacer sus

propias pruebas.

Las pruebas de software suelen realizarse durante el desarrollo del sistema y después de que éste ha sido terminado. Estas 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 [4].

Por tal motivo la prueba es una actividad fundamental en muchos procesos de desarrollo, incluyendo el del software. De manera general, se puede decir que la prueba de software permite al desarrollador determinar si el producto generado satisface las necesidades y especificaciones establecidas. Así mismo, una prueba de software permite detectar la presencia de errores que pudieran generar salidas o comportamientos inapropiados durante su ejecución.

### **Objetivos de las pruebas de software [5]:**

- Encontrar y documentar los defectos que puedan afectar la calidad del software.
- Validar que el software trabaje como fue diseñado.
- Validar y probar los requisitos que debe cumplir el software.
- Validar que los requisitos fueron implementados correctamente.

### **1.1.1 Propiedades de las pruebas**

Las propiedades de las pruebas se definen de acuerdo a la forma de aplicación y se agrupan en tres grupos: estructural o funcional, dinámica o estática y manual o automatizada.

Tabla 1.1 Propiedades de las pruebas

Estructural / Funcional
-------------------------

<p>En el análisis estructural de las pruebas la tendencia es encontrar errores que ocurren durante la codificación del sistema [4].</p> <p>Las pruebas estructurales aseguran la comprobación de la implementación de una función. Este tipo de pruebas se utiliza generalmente durante la fase de codificación. Se recomienda utilizar el análisis estructural en todas las fases del ciclo de vida en donde el sistema se representa formalmente como un algoritmo, diseño o lenguaje [6].</p>	<p>Las pruebas funcionales aseguran que el sistema cubre los requerimientos y las funciones para el cual fue diseñado. Estas únicamente proporcionan información sobre el resultado del proceso [6].</p> <p>En el análisis funcional de las pruebas la tendencia es encontrar errores que ocurren en la implementación de requerimientos o especificaciones del plan [4].</p>
<p>Dinámica / Estática</p>	
<p>El análisis dinámico requiere que el sistema se ejecute. Es decir, el programa se corre en algún caso de prueba y se examinan los resultados del desempeño del mismo para verificar si el programa opera como se espera. La prueba dinámica se utiliza generalmente en la fase de prueba del sistema [4].</p>	<p>El análisis estático normalmente no involucra la ejecución real del programa. Las técnicas comunes del análisis estático incluyen tareas como verificación de la sintaxis. La prueba estática se realiza sin ejecutar los programas operacionales, esto requiere hacer un seguimiento en papel para determinar que los sistemas funcionan apropiadamente cuando se ejecutan. La prueba estática se utiliza generalmente en fase de requerimientos y de diseño [4].</p>
<p>Manual / Automatizada</p>	
<p>Las pruebas manuales son ejecutadas por las personas, y las pruebas automatizadas por una computadora, mediante herramientas que realizan esta tarea. En los procesos</p>	

de desarrollo se recomienda utilizar procesos de pruebas automatizados por su simplicidad. Cuando se cuenta con personal capacitado para analizar, documentar y desarrollar sistemas a mano, se recomienda utilizar la prueba manual [6].

### **1.1.2 Niveles de prueba del software**

Un nivel de prueba permite especificar el alcance de la prueba de software que se realiza. Las pruebas se pueden clasificar en pruebas de bajo nivel (pruebas de unidad, de integración) y de pruebas de alto nivel (pruebas de usabilidad, de sistema, de funcionalidad y de aceptación) [4].

#### **Bajo nivel**

En este nivel están consideradas todas aquellas pruebas que se realizan a componentes o módulos individuales de un programa uno por uno o combinados. Requieren un conocimiento de la estructura interior del programa. Las pruebas que se pueden realizar en este nivel son:

#### **Prueba de unidad**

Este tipo de prueba se aplica a elementos de software individualmente, excluyendo todos aquellos casos en los que se considere la interacción con otras unidades. La prueba de unidad se define como el proceso de probar los módulos individuales de un programa. Su propósito es descubrir las diferencias entre la especificación de la interfaz del módulo y su conducta real [7]. La prueba de unidad es una combinación de varias pruebas. Lo que facilita la detección de errores y permite probar varios módulos simultáneamente y en forma independiente.

#### **Prueba de integración**

Las pruebas de integración se realizan con el propósito de ejercitar la arquitectura de un sistema, es decir, es el proceso de combinar y probar varios módulos unidos. Una vez que ya se han probado que las unidades funcionan de forma correcta de forma aislada, se procede a probar cómo funcionan al integrarlas con otras de forma que se llegue a probar el comportamiento de un sistema final. El objetivo principal es descubrir los errores en las interfaces entre los módulos. Existen varias alternativas en las pruebas de integración, una de ellas es la integración incremental. En la integración incremental se prueba cada módulo agregándose al conjunto de componentes previamente probados [4].

#### **Alto nivel**

En éste nivel las pruebas se orientan a un producto completo es decir se refiere al conjunto de pruebas que se aplican a un producto en la fase de liberación; para ello se proponen las siguientes alternativas:

### **Prueba de usabilidad**

La prueba de usabilidad es el proceso de identificar las diferencias entre las interfaces del usuario de un producto y los requerimientos del cliente [7], la prueba de usabilidad implica tener a los usuarios trabajando con el sistema y observando su reacción ante él, recopilando información de los problemas específicos que reportan los usuarios, a menudo involucra la evaluación de la presentación de un producto en lugar de su funcionalidad [4].

### **Prueba de función**

Cada función en el sistema tiene una sola especificación definida con claridad. Las pruebas de función pretenden descubrir las diferencias entre la especificación funcional de un sistema y su conducta real, [7]. Por tal motivo las pruebas de función se realizan antes de que el producto esté disponible al cliente y pueden empezar a aplicarse siempre que el producto tenga la funcionalidad suficiente para ejecutar algunas de las pruebas, o después de que las pruebas de unidad e integración hayan terminado [4].

### **Pruebas de sistema**

Las pruebas de sistema intenta demostrar que el sistema cumple sus requerimientos y objetivos originales como se declaró en el documento de requerimientos. Los requerimientos deben ser lo más específicos posible para probarse pero conservando la generalidad para permitir la libertad en el diseño funcional. En este tipo de pruebas los probadores tratan a menudo de “romper” el sistema introduciendo datos o ejecutando acciones erróneas [7].

### **Pruebas de aceptación.**

Las pruebas de aceptación Comparan el producto final con las necesidades actuales del cliente. Normalmente se realizan una vez que se han completado las pruebas de usabilidad, función, y de sistema satisfactoriamente. Las mismas involucran el funcionamiento y operación del software en un período específico de tiempo que puede durar hasta un año y se efectúan con datos reales [7].

Si el software es desarrollado bajo contrato, las pruebas de aceptación son realizadas por el cliente. Esto obedece al hecho de que el criterio de aceptación se define en el contrato. Pero si el producto no se desarrolla bajo contrato, la organización puede disponer de formas alternativas de prueba de aceptación como son: alfa y beta. Entre estas primero se le realizan las pruebas al alfa y luego al beta. Las dos involucran el funcionamiento y operación del sistema en un período especificado [7].

En el caso de las pruebas alfa normalmente son realizadas dentro de la compañía desarrolladora, se utiliza el sistema en forma natural con el desarrollador como observador del usuario y registrando los errores y problemas de uso. Las pruebas beta se llevan fuera de la compañía que desarrollo el software.

A diferencia de la prueba alfa el desarrollador no está presente. El primer paso para la implementación alfa y beta es definir objetivos primarios de la prueba: pruebas progresivas y/o regresivas. Las pruebas progresivas son el proceso de probar código nuevo para determinar si contiene errores y las pruebas regresiva es el proceso de probar el sistema para determinar si un cambio ha introducido errores en el código que se ha modificado [4].

### 1.1.3 Métodos de Prueba

Los métodos de pruebas no son más que formas de realizar pruebas a un sistema informático para comprobar el cumplimiento de los requerimientos y de las funcionalidades. De acuerdo a la forma de realizar esta comprobación se clasifican en: pruebas de caja blanca y pruebas de caja negra.

#### Caja blanca

La prueba de caja blanca se basa en el examinar todos los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones. Se puede ver el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado [8].

Mediante las pruebas de caja blanca, el ingeniero de pruebas puede obtener casos en donde se [4]:

- Garantice que se ejercita por lo menos una vez todos los caminos independientes de cada módulo.
- Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Ejecuten todos los lazos en sus límites y con sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

#### Caja Negra

La prueba de caja negra son las pruebas que se llevan a cabo sobre la interfaz del sistema y se centran en los requerimientos funcionales del mismo. Permiten al ingeniero de sistemas obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales de un programa [8].

La prueba de caja negra pretende demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, y se mantiene la integridad de la información externa. Una prueba de caja negra examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software [4].

Según Pressman (1999), los tipos de errores que intenta descubrir esta prueba son: funciones

incorrectas o ausentes, errores de interfaz, errores en las estructuras de datos o en accesos a bases de datos externas, errores de rendimiento, errores de inicialización y de terminación [9].

Las pruebas de software son un elemento crítico dentro del aseguramiento de la calidad de un producto, de ahí la importancia de que estos salgan libres o con la menor cantidad de errores posibles. Para que un proceso de pruebas sea efectivo es recomendable llevar a cabo una planificación en tiempo donde queden bien documentados todos los elementos a probar, los recursos y responsabilidades como los resultados de las pruebas a realizar. Se debe además diseñar métodos y llevar a cabo niveles de pruebas que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo. Además que deben realizarse siguiendo un plan, si no se desperdicia tiempo, se dedica un esfuerzo innecesario y aun peor, es posible que no se detecten errores.

### 1.2 Enfoque de los modelos de desarrollo de software

Con respecto a los modelos de pruebas software, es preciso indicar que son un factor fundamental para lograr el éxito de los proyectos. Para su aplicación es necesario adaptar sus prescripciones a la naturaleza del proyecto. Para mejorar los procesos de desarrollo se crearon los modelos de desarrollo entre los cuales se encuentran el modelo de desarrollo en Cascada y en Espiral los cuales fueron la base para los modelos más actuales de desarrollo como son:

**El modelo TMMI (Modelo de Madurez Integrado de Prueba):** Este modelo es un marco de referencia desarrollado por la fundación TMMI, como guía para el mejoramiento de procesos de pruebas. TMMI podría ser utilizado en organizaciones que cuentan con una estructura organizativa de pruebas basada en equipos independientes. Sin embargo, no se encuentra totalmente definido; únicamente se ha desarrollado para los procesos situados en el nivel 2 de madurez. En el caso de que los desarrolladores sean los testers o equipos integrados de pruebas, este modelo de referencia resulta inapropiado, debido a la dimensión reducida de estos equipos la carga de trabajo sería excesiva [11]. TMMI (al igual que CMMI) define 5 áreas de madurez o jerarquía evolutiva donde cada nivel de madurez contiene un conjunto de áreas de proceso.

**El modelo CMMI [11]:** Es un modelo cuya finalidad es ayudar a las organizaciones a la mejora de la capacidad y madurez de sus procesos tanto para el desarrollo y mantenimiento de sus productos y servicios, mediante la colección de mejores prácticas. Además cuenta con 5 niveles de madurez.

Inicial o Nivel 1 CMMI. Este es el nivel en donde están todas las empresas que no tienen procesos.  
Nivel 2 CMMI. El proyecto es gestionado y controlado durante el desarrollo del mismo. El desarrollo no es opaco y se puede saber el estado del proyecto en todo momento.

Nivel 3 CMMI. Alcanzar este nivel significa que la forma de desarrollar proyectos (gestión e ingeniería)

está establecida, documentada y que existen métricas (obtención de datos objetivos) para la consecución de objetivos concretos. La mayoría de las empresas que llegan al nivel 3 paran aquí, ya que es un nivel que proporciona muchos beneficios y no ven la necesidad de ir más allá porque tienen cubiertas la mayoría de sus necesidades.

Nivel 4 CMMI. Los proyectos usan objetivos medibles para alcanzar las necesidades de los clientes y la organización. Se usan métricas para gestionar la organización.

Nivel 5 CMMI. Los procesos de los proyectos y de la organización están orientados a la mejora de las actividades.

**Modelo de desarrollo V [12]:** Definido inicialmente por Paúl Rook al final de los años 80, el modelo-V, tiene como objetivo mejorar la eficacia y la efectividad del desarrollo de software. Este emergió del modelo en cascada que trataba las pruebas como una fase simple dentro del proceso de desarrollo tradicional. Dentro de este modelo la fase de pruebas tiene la misma importancia que las otras fases del desarrollo. La parte izquierda del Modelo-V representa el proceso donde se definen las especificaciones del sistema, lo que representa la definición conceptual del sistema a partir de los requerimientos de usuario, las especificaciones de estos y el diseño conjunto con la definición de la arquitectura. La parte derecha representa el proceso donde se prueba el sistema (contra las especificaciones definidas en la parte izquierda), donde se muestran los pasos para realizar las pruebas unitarias, de integración y verificación, así como la validación del sistema por el usuario y posteriormente las operaciones de mantenimiento.

Este modelo es indicado para aquellos proyectos que requieren diferentes tipos de pruebas antes de ponerlos en funcionamiento debido a su complejidad, y en donde las actualizaciones son difíciles y costosas. El modelo puede ser flexible, especialmente para procesos de desarrollo diferentes a los de un software estándar, como es el caso de los sistemas operativos o las distribuciones GNU/Linux. [anexo I]

**Modelo W [13]:** Está formado por las empresas Evolutif, es una mejora del modelo V. Este está formado por dos modelos en forma de V, lo que representa la prueba y el proceso de desarrollo. En este modelo las pruebas y el desarrollo del software se llevan a cabo de forma simultánea, es decir, este modelo ayuda a identificar problemas a tiempo completo. Por ejemplo, el análisis de necesidades se realiza completo, los probadores deben participar en la verificación y validación del proceso con el fin de localizar el origen del problema y solucionarlo lo más pronto posible. Mientras que la realización de las pruebas de forma continuada es beneficiosa pues disminuye la duración de las pruebas así como su complejidad, desarrollar tempranamente estrategias de respuesta podría reducir dramáticamente el

tiempo total de respuesta y aceleraría el proyecto.

Durante la fase de prueba, el desarrollador es responsable de la eliminación de defectos y la corrección de la aplicación. Esto subraya claramente el hecho de que la prueba es más que la construcción, ejecución y evaluación de casos de prueba. [anexo II]

### **1.3 Procesos de Pruebas de Distribuciones GNU/Linux**

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

En el caso de las distribuciones GNU/Linux la existencia de las comunidades de desarrollo como parte del equipo de diseño, implementación y pruebas ha generado una tendencia a la liberación de versiones mínimamente funcionales, con el objetivo de someterlas a prueba por la comunidad. Aquí se introducen los conceptos de imagen Alpha ( $\alpha$ ): la cual es una primera versión donde se le añaden nuevas funcionalidades a este, beta ( $\beta$ ): donde se corrigen nuevos errores encontrados en la etapa anterior, testing: es la versión que se libera para que la comunidad lo pruebe y vea se está acorde con sus necesidades: donde todos los errores han sido eliminados y esta lista para ser distribuida. Las distribuciones GNU/Linux pasan por un exhaustivo ciclo de pruebas que puede incluir en su equipo de probadores a usuarios finales con pocos conocimientos sobre informática [14].

“Entre las diversas distribuciones GNU/Linux existen un conjunto que basan su desarrollo en las comunidades, como es el caso de GNU/Linux Debían. Otras son desarrolladas por empresas, con el apoyo de la comunidad, entre estos casos podemos encontrar a Ubuntu. Por último se encuentran las distribuciones propietarias que poseen generalmente mayor robustez a cambio de un costo monetario por obtener soporte del sistema, caso típico de la distribución Red Hat” [14].

Por ello se hace referencia solamente al proceso de pruebas de la distribución Ubuntu Linux que está publicada en su sitio oficial y al actual proceso de Nova, pero no ocurre lo mismo con las otras distribuciones de GNU/Linux que, en la mayoría de los casos mencionan el uso de algún proceso de pruebas pero no lo comparten.

#### **1.3.1 Proceso de pruebas de la distribución GNU/Linux, Ubuntu [15].**

La distribución GNU/Linux, Ubuntu ha creado un conjunto determinado de pruebas con el objetivo fundamental de tratar de mejorar el software a través de pruebas estructuradas y exploratorias de los paquetes y las imágenes ISO del sistema. Además de detectar la mayor cantidad de errores como sea

posible antes de que los CDs se liberen al público en general.

Las pruebas que se realizan a través de un conjunto de actividades documentadas en la “lista de actividades”, las cuales son días antes de la liberación de una imagen ISO (Alpha, Beta o Release candidate), descargar la última versión hecha, grabarla en un CD y probarla. Esta ayuda a detectar varios errores y no conformidades no vistas por los desarrolladores, principalmente en los CD de instalación. Este tipo de pruebas se denominan pruebas de ISO las que se centran en probar el funcionamiento correcto del LiveCD; las variantes de instalación, entre ellas la que permite redimensionar una de las particiones antes de instalar y la de particionado personalizado.

Otro tipo de pruebas son las llamadas pruebas de humo diarias, se realizan sobre una base diaria normalmente en una máquina virtual. Estas se basan en que las regresiones, los errores y fallas generales sean detectados lo más temprano posible antes de que el producto sea liberado. Las pruebas de humo diarias están dadas principalmente a los distintos tipos de instalación, aunque en dependencia del tiempo se prueban también las aplicaciones en los sistemas instalados. Una de las pruebas más importantes son las pruebas de actualizaciones a versiones estables (SRU), debido a que los usuarios esperan un alto grado de estabilidad de estas versiones liberadas. Estas actualizaciones tienen como objetivo fundamental, corregir los errores de alto impacto, tales como: vulnerabilidades de seguridad, errores de integridad de los datos, nuevas versiones de software comercial, entre otros. Realizan además pruebas para las características principales del Release, las pruebas generales y las de aplicaciones.

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

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

### 1.3.2 Proceso de pruebas de la distribución GNU/Linux Nova [14].

El proceso de pruebas de la distribución cubana GNU/Linux Nova incluye un plan de pruebas donde se contemplan los requerimientos a probar, un cronograma de las tareas o pruebas a ejecutar acompañadas del responsable y la fecha en que deben efectuarse, por último se incluye como y donde se reportan las no conformidades o errores encontrados para su posterior corrección. También cuenta con una serie de casos de pruebas que estarán igualmente clasificados en pruebas de aplicaciones, sistema, soporte de hardware e instalación del sistema desde el LiveCD. Este proceso de pruebas a Nova esta ajustado al modelo V, en el cual se definen los componentes fundamentales de la distribución, lo cual permite una mejor comprensión de este proceso.

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

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

De este proceso de pruebas hay que señalar que a pesar de no ser oficial, hace referencia a las pruebas de unidad, integración, sistema y aceptación pero descuida dentro de las pruebas de aceptación las de rendimiento que deberían ser un factor determinante en los procesos de calidad de la distribución Nova pues uno de sus objetivos como Sistema operativo es funcionar en ordenadores de bajas prestaciones y para ello se hace indispensable conocer el rendimiento de cada nueva versión.

### 1.4 Herramientas para utilizar las pruebas automáticas

Las pruebas de automatización al software utilizan aplicaciones para controlar la ejecución de las mismas, comparar los resultados reales con los resultados previstos, la creación de las condiciones previas de la prueba, y el control de la prueba. Con la automatización de las pruebas se reduce el tiempo necesario para cada prueba así como el costo, permitiendo probar más funcionalidades con menor cantidad de personal y obteniendo una calidad superior.

En esta investigación se consideraron una variedad de sistemas automáticos, pero solo se referencia las herramientas más desarrolladas pues muchas se encuentran en las primeras versiones y son altamente inestables. Se encontraron sitios de Internet que realizan algunas pruebas on-line pero esto no es rentable debido a las políticas de uso de la universidad.

### 1.4.1 Dogtail [16]

Dogtail es una herramienta GUI (Interfaz Gráfica de Usuario, por sus siglas en inglés) y Framework de pruebas que contiene un marco de automatización escrito en Python, uno de los lenguajes de programación más fácil de aprender y de usar. Es una aplicación para Software Libre por tanto se puede usar y modificar su código. Actualmente soporta aplicaciones de 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.

### 1.4.2 Phoronix Test Suite [17]

Phoronix Test Suite es una de las aplicaciones más desarrolladas en la realización de pruebas automáticas, usa herramientas diseñadas por Phoronix (Empresa de medios de Internet orientada a mejorar la experiencia del hardware en Linux) desde el 2004 en colaboración con los principales productores de hardware y software. Este software es de código abierto y 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 130 perfiles de prueba como son el chequeo del consumo de la batería, la CPU, gráficos, memoria de sistema de almacenamiento en disco y los componentes de la placa base. El Phoronix Test Suite se compone de un núcleo de procesamiento ligero (**pts-core**) con cada punto de referencia que consiste en un perfil basado en XML y secuencias de comandos de recursos relacionados.

#### **Características:**

- **Fácil de usar:** Esta herramienta hace que el proceso de realización de pruebas automatizadas sea sencillo.
- **Arquitectura Extensible:** Una prueba es capaz de extender y potenciar un perfil ya existente.
- **Resultados de la grabación:** Los resultados de esta herramienta muestran en un visor de resultados basados en estilos Web con soporte opcional para subirlos a "<http://global.phoronix-test-suite.com>".

- **Multiplataforma:** Esta aplicación, fue diseñada principalmente para la realización de pruebas en Linux, además soporta a OpenSolaris, Mac OS X, Windows y sistemas operativos BSD.

### 1.4.3 Mago [18]

Mago es una aplicación basada principalmente en LDTP, un Framework de pruebas de escritorio, escrita en C. Esta herramienta contiene un corredor de pruebas el cual permite ejecutar las pruebas constantemente y ser capaz de presentar informes agradables de sus pruebas. Las pruebas están organizadas por distintas carpetas de aplicaciones XML, donde cada una de los archivos XML contiene uno o más casos de pruebas. Para poder trabajar con Mago se debe tener conocimiento del lenguaje Python para poder diseñar las pruebas pues se deben crear de forma manual.

Según el análisis realizado en el presente capítulo de los modelos se puede concluir que el modelo TMMI no es conveniente pues desciende del modelo CMMI y se encuentra en una fase inicial. El modelo CMMI describe explícitamente todo el proceso de preparación de las pruebas, que incluye la selección del producto a probar, la selección del entorno donde se va a probar y los criterios de validación y prueba por los cuales se registrará el proceso. El proceso V se centra en validar el producto a partir de la relación entre las actividades de la ingeniería y las pruebas, pero no contempla relación entre los procesos de prueba y la corrección de errores de forma simultánea. El modelo W corrige al modelo V adicionando una segunda V al proceso permitiendo una relación directa entre las pruebas y la corrección de errores.

De las herramientas analizadas se puede decir que en el caso de Dogtail y de Mago se deben tener conocimientos de lenguaje Python para poder diseñar las pruebas pues se deben crear de forma manual. Se propone la utilización de la herramienta Phoronix Test Suite, pues al contar con toda una serie de pruebas prediseñadas se garantiza un nivel de exactitud muy superior en las pruebas a la hora de comparar su software con otros similares en el mercado proporcionando información fidedigna acerca de las pruebas realizadas, lo que permite trabajar en las estadísticas y manejar el rendimiento de nuestro sistema en diferentes arquitecturas de hardware.

Se recomienda la utilización del modelo de desarrollo de software W combinado con CMMI debido a que CMMI recomienda buenas prácticas para organizar un ciclo de pruebas, desde la selección del producto a probar, pasando por la definición del ambiente de pruebas hasta la selección de los criterios de prueba pero no especifica la forma en que se debe realizar dicho proceso, es en este punto donde el modelo W presta una gran ayuda definiendo de forma explícita cómo realizar el proceso de pruebas.

## **Capítulo 2: “Propuesta de proceso de pruebas para la distribución cubana de Software libre GNU/Linux Nova”**

En este capítulo se describe la propuesta de proceso de pruebas para la distribución de software libre GNU/Linux Nova, obtenida luego de realizar un análisis del modelo de desarrollo de software W y de las mejores prácticas que CMMI recomienda y que incluye la utilización de la herramienta de pruebas automáticas Phoronix Test Suite. Este proceso viene acompañado de las instrucciones necesarias para instalar la herramienta y usarla, de forma tal que el proceso sea lo más rápido y entendible posible para el equipo de aseguramiento de la calidad. De la herramienta se describen las características más significativas, entre las que destacan: las dependencias necesarias para su instalación, el proceso de instalación paso a paso y por último se describe la mejor forma de realizar las pruebas y analizar sus resultados.

### **2.1 Proceso de Pruebas para la distribución GNU/Linux Nova.**

El proceso de pruebas para Nova comienza con la captura de los requerimientos del sistema pues a partir de ellos se formulan los casos de pruebas que se utilizarán. En este proceso debe existir una excelente coordinación de los Aseguradores de la Calidad y los Desarrolladores, pues de la buena comunicación entre estos equipos depende la rápida respuesta y solución de los errores de software (bugs) y no conformidades encontradas.

#### **Propósito**

El propósito fundamental es encontrar la mayor cantidad de errores y no conformidades en las versiones Alpha y Beta, lo que permitiría corregirlos antes de liberar la distribución, incidiendo directamente en la calidad y el prestigio de Nova.

#### **Alcance**

Se considera necesario realizar una evaluación que permita conocer el estado de la distribución GNU/Linux Nova en un momento en el que la Revolución necesita liberarse tecnológicamente del software propietario y en el que el proyecto Nova necesita dar un salto cualitativo en cuanto a calidad para convertirse en uno de los instrumentos principales de esta batalla. Con este objetivo se necesitan encontrar la mayor cantidad de errores posibles en las distintas versiones de nova de forma tal que puedan ser corregidos lo más rápido posible a fin de evitar consumir una mayor cantidad de recursos y

## CAPÍTULO 2: PROPUESTA DE PROCESO DE PRUEBA PARA LA DISTRIBUCIÓN CUBANA DE SOFTWARE LIBRE GNU/LINUX NOVA

---

tiempo en arreglarlos. Por esto se considera que el alcance de este proceso de pruebas es la mejora de la calidad de los productos liberados por el proyecto Nova mediante la realización de un proceso de pruebas adaptado a las necesidades Nova y el uso de una herramienta de pruebas automáticas que permite abarcar otros puntos de vista que puedan incrementar la calidad del producto liberado

Tabla 2.1 Roles

<b>Nova</b>	
<b>Rol</b>	<b>Responsabilidades específicas</b>
Administrador de la calidad	Dirigen el proceso, definiendo las pautas a seguir en las pruebas al software y son los encargados de transmitir los resultados de las pruebas al grupo de desarrollo.
Asegurador de la calidad	Garantiza la calidad de la documentación generada en el proyecto, realizando revisiones periódicas.
Probador	Es el encargado de ayudar al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
Gestor de las configuraciones	Controla los cambios de las versiones, evitando futuros errores.

### **Especificación de software o hardware necesario**

#### **Requerimientos de hardware de pruebas**

1. Varios ordenadores con distinta arquitectura y prestaciones como mínimo 512 Mbyte de RAM y procesador de 1.0 Ghz en adelante.
2. Cámara fotográfica digital
3. Scanner

4. Impresora
5. Audífonos con micrófono
6. Webcam

### **Requerimientos de software**

- Versión 2.8.1 del software Phoronix Test Suite y los paquetes adicionales necesarios (pecl.deb y php-gtk.deb).
- Versión correspondiente del producto a probar. (ISO de Nova)

### **Diseño de las pruebas**

El diseño de las pruebas comienza con la especificación de los requerimientos, así como el listado oficial de aplicaciones, los cuales definen las funcionalidades del sistema a implementar así como las aplicaciones que facilitan la realización de estas, donde el proceso de definición de las pruebas y la especificación de los requisitos son procesos simultáneos. Cuando estos requisitos quedan registrados entonces se procede a escoger qué aplicaciones son las óptimas para desarrollar estas funcionalidades y se procede a conformar el sistema. En este proceso las pruebas se separan en dos grupos fundamentales: las pruebas que se realizarán de forma manual a la versión Alpha y las pruebas que se realizarán de forma manual y automatizada a la versión Beta.

Las pruebas manuales se diseñan en base a las aplicaciones que darán cumplimiento a los requisitos descritos. El diseño de las pruebas se basa en el método de caja negra, las mismas se encuentran distribuidas según la aplicación a la que van destinadas y en ellas se describen un flujo de eventos que debe seguir el probador así como el resultado esperado. Dentro de las pruebas manuales se encuentran las pruebas dirigidas al liveCD, el cual contiene una versión del sistema que funciona sin necesidad de instalarse en el ordenador huésped y que incluyen muchas de las mismas pruebas dirigidas a las aplicaciones que correrán sobre el sistema instalado; las pruebas de aplicaciones, que están dirigidas a comprobar que las aplicaciones del sistema cumplen con los requisitos ya definidos y las pruebas de instalación que buscan comprobar que el software diseñado se instala correctamente en el ordenador.

El diseño de las pruebas automáticas está determinado en su mayoría por una serie de perfiles de prueba a aplicar con objetivos específicos que contiene la herramienta y estos se ejecutan según los parámetros de la distribución a probar.

### **Planificación**

El proceso de pruebas comienza con la primera versión alpha a la cual se le realiza un primer ciclo de prueba, una vez concluido este se envían a los desarrolladores las no conformidades encontradas y se les da un tiempo no mayor a una semana para corregirlos. Luego de este período se libera la segunda versión alpha a la cual se le realiza un segundo ciclo de prueba y así sucesivamente hasta que no se encuentren errores. Cuando sean superadas estas pruebas esta imagen Alpha se convierte en la versión Beta y se realizan las pruebas correspondientes a la versión actual de la misma forma que se le realizó a la versión Alpha. Al final de estos dos procesos iterativos se obtiene una versión Release o liberación que es la que se entrega al cliente. El proceso de pruebas de la distribución GNU/Linux Nova debe estar registrado en un plan de pruebas donde se contemplan los requisitos a probar, un cronograma de las tareas o pruebas a probar acompañado de un responsable y la fecha en que deben ejecutarse, además de incluir como y donde se reportan las no conformidades o errores encontrados para su posterior corrección.

### **Procedimiento de las pruebas**

Como se explica en el epígrafe anterior luego de finalizada la conformación del sistema basándose en los requerimientos del cliente se obtiene una versión Alpha con lo cual comienzan las pruebas que incluyen las pruebas al LiveCD, pruebas de instalación y pruebas al gestor de paquetes, lo que garantiza que los requisitos primarios del sistema sean cumplidos. El procedimiento de pruebas para el liveCD comienza con la introducción del CD en la unidad se reinicia el ordenador, seleccionando en las opciones de booteo el arranque desde el CD y luego se realizan las pruebas previamente diseñadas. Las pruebas de instalación verifican que este permita instalar el sistema en el ordenador cumpliendo una serie de parámetros, por ejemplo:

1. Instalar el sistema usando el disco completo.
2. Instalar utilizando el espacio libre en el disco duro.
3. Instalar de forma manual el sistema, personalizando el uso del espacio en el disco duro.

Luego de los sucesivos ciclos de pruebas y corrección dirigidos a perfeccionar la versión Alpha se libera una versión Beta a la cual se le realizan varios tipos de pruebas:

- Pruebas de actualización en las cuales se comprueba que la versión Alpha instalada detecta de forma automática las actualizaciones al indicarle un cambio en los repositorios de paquetes y solicita la actualización del sistema permitiendo adicionar al sistema las mejoras sin reinstalar el sistema.

- Las pruebas de rendimiento son pruebas que se realizan de forma automática, que tienen como objetivo conocer de forma cuantificable el rendimiento del sistema en el ordenador instalado, comprobando aspectos tales como el rendimiento del procesador, del disco duro, aceleradores gráficos, entre otros.
- El uso diario de un sistema instalado permite detectar errores que un proceso de pruebas simple no encontraría pues fomenta el uso de funcionalidades poco reconocidas pero importantes, constituyendo una importante retroalimentación.

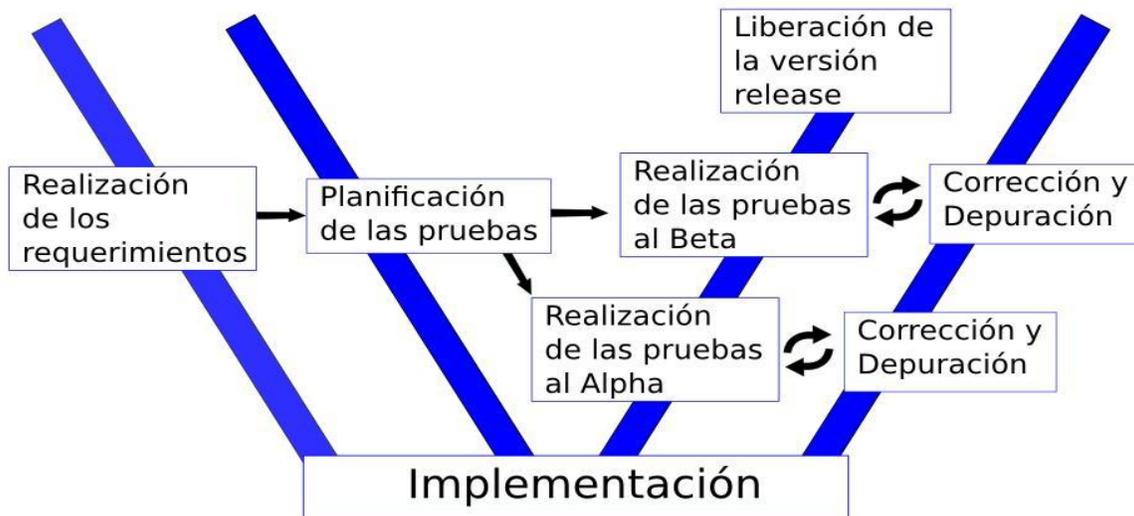


Figura 2.1 Proceso de Pruebas de la distribución GNU/Linux Nova.

### Definición de entregables

Los elementos entregables para este proceso son:

- Plan de pruebas: Donde se contemplan los requerimientos a probar, un cronograma de las tareas o pruebas a ejecutar acompañadas del responsable, la fecha en que deben efectuarse y los recursos asignados para cada prueba.
- Casos de prueba: Documento que contiene los casos de pruebas que permiten definir el grado

## CAPÍTULO 2: PROPUESTA DE PROCESO DE PRUEBA PARA LA DISTRIBUCIÓN CUBANA DE SOFTWARE LIBRE GNU/LINUX NOVA

---

de aceptación del producto y el cumplimiento de las funcionalidades acordadas con el cliente.

- Reportes de las pruebas: Se reportan las no conformidades y errores encontrados para su posterior corrección.

### Recursos requeridos

Tabla 2.2 Recursos humanos

<b>Nova</b>	
<b>Rol</b>	<b>Cantidad (recursos mínimos necesarios)</b>
Administrador de la calidad	1
Asegurador de la calidad	1
Probador	5
Gestor de las configuraciones	1

Tabla 2.3 Recursos de trabajo

<b>Recursos</b>	<b>Cantidad (recursos mínimos necesarios)</b>	<b>Nombre/Tipo</b>
Ordenadores	5	Probar LiveCD o Instalación del software.
Escáner	1	Probar compatibilidad del hardware.
Impresora	1	Probar compatibilidad del hardware.
Cámara Digital	1	Probar compatibilidad del

## CAPÍTULO 2: PROPUESTA DE PROCESO DE PRUEBA PARA LA DISTRIBUCIÓN CUBANA DE SOFTWARE LIBRE GNU/LINUX NOVA

---

		hardware.
Webcam	1	Probar compatibilidad del hardware.
Micrófono con Audífono	2	Probar aplicaciones de audio y vídeo.

### Criterios de validación

La validación se define como el proceso en el cual se ratifica que el producto de software funciona de tal manera que satisface las expectativas razonables del cliente (especificación de requisitos-criterios de validación).

Después de que se ha dirigido cada caso de prueba de validación, existirán dos condiciones posibles:

1. La característica de funcionamiento o desempeño cumple con la especificación y se la acepta.
2. Se descubre una desviación de la especificación y se crea una lista de no conformidades.

Las no conformidades se reenvían al equipo de desarrollo para ser tratadas y verificadas en el próximo ciclo de pruebas.

### Estrategia de pruebas

#### Versión Alpha:

**Pruebas al sistema:** Las pruebas al sistema se realizan teniendo en cuenta básicamente el soporte de hardware así como las especificaciones más distintivas.

- **Pruebas al LiveCD:** Se realizan casi todas las pruebas dirigidas al sistema exceptuando las del gestor de paquetes.
- **Prueba de Instalación:** Se comprueba que el LiveCD instale de forma correcta en el ordenador la versión Alpha de Nova.
- **Pruebas del Gestor de paquetes:** Se realizan una vez instalado el sistema y tienen por objetivo comprobar que el mismo permite instalar, actualizar y desinstalar aplicaciones.

#### Versión Beta:

**Pruebas de aceptación:** Se realizan una vez que el sistema ha superado todos los errores encontrados en la versión anterior y tienen como objetivo comparar el producto con las necesidades actuales del

cliente. La comunidad juega en este punto un papel importante pues su participación en estas pruebas es determinante para validar si sus necesidades se ven reflejadas en el producto obtenido y así apoyar a los aseguradores de calidad del proyecto.

- **Pruebas de actualización:** Una vez corregidos los errores encontrados en la versión Alpha se conforman los parches que solucionarían estos problemas. Una vez hecho esto se debe poder actualizar el sistema completo de forma automática, si la actualización transcurre sin problemas la prueba de actualización ha sido un éxito. Tras la actualización debe estar en presencia de la versión Beta
- **Pruebas de uso diario:** Estas están constituidas por el uso continuo del sistema de forma tal que se puedan probar funcionalidades que a pesar de no ser significativas tienen un peso en la valoración de calidad más exigente (la del cliente).
- **Pruebas de rendimiento:** Estas pruebas se realizan utilizando la herramienta Phoronix Test Suite, que permite obtener datos comparables a otras distribuciones y sistemas operativos del mundo entero. Luego de un periodo de tiempo y de haber sido corregidos los errores del Beta se libera la Versión Release la cual se distribuye a los clientes.

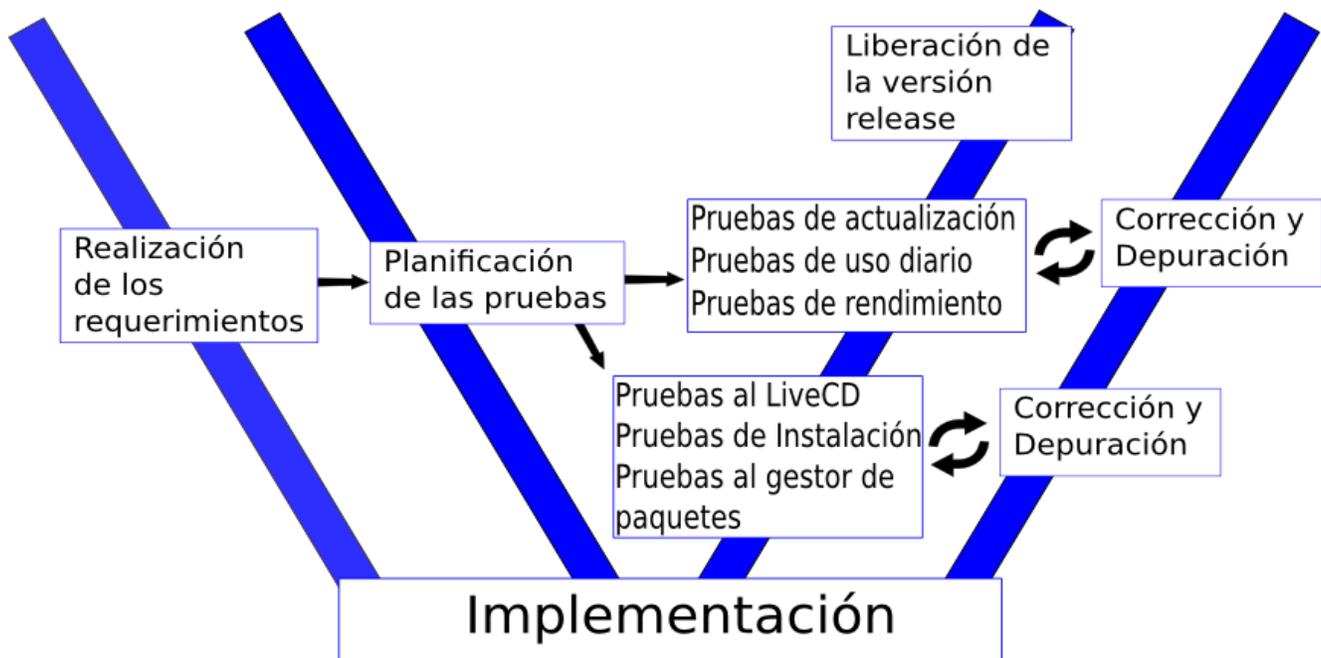


Figura 2.2 Estrategia de pruebas de la distribución GNU/Linux Nova.

## 2.2 Instalación y puesta en marcha del Phoronix Test Suite

### Instalación del Phoronix Test Suite

Como se explicó en el capítulo anterior la herramienta Phoronix es una de las más utilizadas a nivel internacional en cuanto a pruebas automatizadas se refiere, contiene el uso de una interfaz sencilla, que a pesar de estar en inglés se encuentra bien estructurada. En versiones anteriores la aplicación no contaba con una interfaz gráfica y todo el trabajo debía realizarse en consola o en un navegador Web, con el objetivo de mejorar la usabilidad de este programa el grupo Phoronix le añadió una interfaz gráfica disponible en la versión 2.8.1 que depende de las librerías php-gtk y pecl. Estas librerías le permiten a la aplicación tener una interfaz, lo cual es más asequible para la mayoría de los usuarios y mucho más fiable que trabajar en un navegador Web que puede consumir abundantes recursos.

Se decidió instalar esta versión por las ventajas antes mencionadas, además de que ofrece una interfaz amigable así como un mayor número de pruebas disponibles. Para su instalación fue necesario descargar por separado las librerías de las que depende pues no se encontraban entre las dependencias requeridas por la distribución para agregar en los repositorios. Para poder utilizar la interfaz gráfica de la aplicación en la distribución Nova fue necesario recompilar los paquetes de las dependencias debido a incompatibilidad de versiones entre estas librerías y el Phoronix en sí mismo. Con el objetivo de insertar a los usuarios de la comunidad en el proceso de pruebas se incluyó la aplicación en los repositorios de Nova, a partir de esto se podrá instalar el Phoronix desde el gestor de paquetes de Nova.

### **Instalación de las pruebas de Phoronix**

Una vez instalado Phoronix es necesario entonces instalar las pruebas o las suites de pruebas para poder usarlas. Esta aplicación nos brinda un conjunto de 132 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 conocer las necesidades de prueba según las líneas de desarrollo de Nova se realizó una encuesta [Anexo III] a los líderes de proyectos y con esta información se procedió a descargar solamente las pruebas necesarias para cumplir con las exigencias del proyecto.

A partir de los resultados de la encuesta a los líderes de las distintas líneas de desarrollo de Nova, se decidió por parte de la dirección del proyecto incluir algunas pruebas que a pesar de no ser las más requeridas, no dejaban de ser importantes.

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

- Copiar el archivo nombrado "nombre".tar-gz o "nombre".zip en la carpeta /home/usuario/.phoronix-test-suite/download-cache.
- Descompactar el paquete de la prueba.
- 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".

### **Incorporación de la herramienta en el proceso de pruebas**

Posterior a la instalación del Phoronix y de las pruebas necesarias se procede a la ejecución de las pruebas al sistema, el orden de ejecución no es estricto debido a que las pruebas son independientes y no modifican al sistema original. Las pruebas pueden ejecutarse desde la ventana principal de la aplicación seleccionando la pestaña "Installed tests" seleccionando la prueba deseada y presionando el botón "Run", una vez concluida la prueba la aplicación muestra en su interfaz principal muestra un enlace hacia los resultados de la prueba que se muestran en el navegador Web, además de abundante información de los recursos de hardware y software del equipo que se está probando en específico. Si se desea ver resultados de pruebas anteriores se selecciona la pestaña "Test Results", la prueba en específico y se presiona el botón "View Results".

El diseño de las pruebas se rige por el siguiente flujo de eventos:

Seleccionar la prueba a ejecutar → Ejecución de la prueba automática → Obtención de los resultados en un informe detallado que se muestra en el navegador Web por defecto. [anexo V]

Los criterios de validación para estas pruebas se basan en la comparación de los resultados de las pruebas con otros resultados realizados en ordenadores con prestaciones similares y con sistemas operativos diferentes, de forma tal que de acuerdo a la prueba que se realice se puede tener un criterio del rendimiento de nuestro sistema. Las pruebas que miden el tiempo necesario para un resultado esperado tienen como valores óptimos los menores; por otra parte las pruebas que evalúan la capacidad de cálculo, de copia o de lectura de algún dispositivo, encuentran en los más altos valores los óptimos.

A continuación se detallan las pruebas para probar el rendimiento del procesador

1. Build-imagemagick: Esta prueba tiene como objetivo medir cuanto tiempo se necesita para construir imagemagick una aplicación constituida por una poderosa colección de herramientas orientadas a la manipulación y procesamiento de imágenes en línea de comandos. Un menor

## CAPÍTULO 2: PROPUESTA DE PROCESO DE PRUEBA PARA LA DISTRIBUCIÓN CUBANA DE SOFTWARE LIBRE GNU/LINUX NOVA

---

tiempo indica un resultado positivo.

2. C-ray: Esta diseñada para evaluar el rendimiento en el cálculo de números con punto flotante de la CPU.
3. FFmpeg: Esta utiliza FFmpeg para probar el sistema de audio y el desempeño de la codificación de vídeo.
4. Openssl: Es un conjunto de herramientas de código abierto que implementa los protocolos SSL (Secure Sockets Layer) y TLS (Transport Layer Security). Esta prueba mide el rendimiento al generar una clave RSA de 4096 bits de OpenSSL.
5. Pray: Se utiliza para ver el rendimiento a la hora de crear gráficos en 3D usando el trazado de rayos.
6. RamSpeed: Esta prueba comprueba el rendimiento a la hora de utilizar la memoria del sistema (RAM).
7. Tachyon: Esta es una prueba para medir el rendimiento del sistema para el trazado de rayos.
8. Lzma: Esta prueba mide el tiempo necesario para comprimir un archivo.

Este proceso de pruebas permite que los productos finales de Nova sean liberados con la calidad requerida de acuerdo a los requisitos del cliente y de la comunidad, especificándose los pasos a seguir para obtener la máxima calidad y que tipos de pruebas realizar a cada versión. Además se incluye un software de pruebas automáticas que permite conocer el rendimiento del sistema y compararlo con otros sistemas operativos. Con la implementación de este proceso se obtiene una guía para la realización de las pruebas a la distribución Nova mejorando el desempeño del equipo de calidad y asegurando un producto fuerte, versátil y adecuado a las necesidades del usuario.

### Capítulo 3: “Validación del proceso de pruebas de la distribución cubana de software libre GNU/LINUX Nova”

La construcción de un correcto proceso de pruebas tiene como objetivo principal lograr que el producto funcione correctamente y que cumpla con las expectativas trazadas. Precisamente para lograr estos objetivos es indispensable que el producto sea evaluado y comprobado a medida que se va construyendo y de esta manera permitir que la detención de defectos se haga lo antes posible y por lo tanto tenga menos impacto en el tiempo y esfuerzo de desarrollo. Con el fin de entregar a los clientes productos satisfactorios, el software debe alcanzar ciertos niveles de calidad y para ello el número de defectos deben ser mínimos. Es por esta razón que el aseguramiento de la calidad de software se ha convertido hoy día en una necesidad prioritaria para los desarrolladores de software, porque cada vez más los errores repercuten directa e indirectamente en graves consecuencias.

En el presente capítulo se evaluará el procedimiento para validar el proceso de pruebas de la distribución GNU/Linux Nova mediante la selección de un método de validación más adecuado.

#### 3.1 Métodos de Validación

**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 [19].

**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 [19].

**Método de recopilación de información:** Se basa en recopilar diferentes opiniones, encuestas, cuestionarios o entrevistas a los clientes o a las personas que tengan que ver de una forma u otra con lo que se está verificando, o con la puesta en práctica de este de forma general [19].

**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 [19].

**Método Experimental:** Este se usa siguiendo un protocolo para examinar un número limitado de variables. Consiste en una forma sistemática de observar los eventos, coleccionando datos, analizando información y reportando resultados. Además realiza ejemplos reales en los que se presenta una historia positiva sobre los beneficios que un producto o servicio les ha significado a unos determinados usuarios [20].

### 3.2 Selección del método de evaluación

Por lo descrito anteriormente se procede a aplicar como métodos de validación el experimental y el método de recopilación de la información, debido a que el método Delphi y grupo focal no se pueden utilizar porque la universidad no cuenta con expertos en las pruebas de software para distribuciones GNU/Linux. En el caso del método de validación práctica este debe aplicarse a varios proyectos y en este caso el proceso de pruebas es solo para Nova.

#### 3.2.1 Método de validación experimental

El método de validación experimental es un proceso que muestra a grandes rasgos qué tan factible es el proceso de pruebas pues los resultados que en él se obtienen comprueban todos los beneficios y las oportunidades que brinda la propuesta desarrollada. Para llevar a cabo este método se utilizan una serie de variables o procesos que constituyen la solución o una parte de la solución que se va a probar y de estas se seleccionan por cada variable un conjunto de atributos que estos representan requisitos o criterios para la solución. Luego de este procedimiento se realiza un profundo análisis del problema antes de que se pusiera en práctica el proceso de pruebas y después de elaborado este proceso de pruebas para así poder conocer cuán factible es el mismo.

Para realizar la validación de la siguiente investigación se determinó realizar una estimación en niveles del 1 al 5 dónde (1 es mal, 2 más o menos regular, 3 regular, 4 bien y 5 excelente). Donde se evalúan cómo se han comportado cada uno de los requisitos Antes y Después del proceso de pruebas y así conocer cuán beneficioso es el proceso de pruebas.

El caso de estudio en cuestión está dado por la mala calidad del producto, la inexperiencia de los probadores, debido a que estos no tienen conocimiento de cómo medir correctamente la calidad del

## CAPÍTULO 3: VALIDACIÓN DEL PROCESO DE PRUEBA DE LA DISTRIBUCIÓN CUBANA DE SOFTWARE LIBRE GNU/LINUX NOVA.

---

software, y la desorganización del proceso de pruebas. Por tal motivo este proceso de pruebas tiene como propósito liberar el producto con la mayor calidad posible.

Las variables seleccionadas son 3 procesos que forman parte de la solución y cada uno de los requisitos constituye procedimientos que deben cumplir cada una de las variables para evaluar su comportamiento.

Las variables que se tienen en cuenta son:

- Organización: organización del proceso de pruebas.
- Capacitación del personal: Nivel de experiencia del personal especializado.
- Calidad: Calidad del proceso de pruebas

Para la existencia de un buen proceso de pruebas y de esta forma permitir una mayor calidad, prestigio y organización del proyecto Nova, se necesitan diferentes procedimientos que le permitan al personal ser capaces de encontrar la mayor cantidad de errores que posea el software y así corregirlos para una mayor calidad en el software. Estos procedimientos constituyen requisitos a tener en cuenta, entre ellos se encuentran:

- Diseñar las pruebas
- Determinar los recursos necesarios
- Planificar las pruebas
- Capacitar al personal
- Realizar todas las iteraciones de pruebas previstas
- Analizar resultados de la pruebas

A continuación se muestran cómo se evidencian tales parámetros Antes y Después de existir un proceso de pruebas que aumente la calidad del producto.

### **Antes**

La distribución cubana de software libre GNU/Linux Nova es un producto con poca calidad a la hora de realizar las pruebas debido a la desorganización, la poca experiencia del personal y además por tener un proceso de pruebas poco organizado y estructurado. Las pruebas que se le realizan a Nova generalmente las llevan a cabo los propios desarrolladores del proyecto. Sólo se realizan dos tipos de pruebas: de integración, hechas por los desarrolladores y pruebas de aceptación y de sistemas, realizadas por los aseguradores de calidad de forma muy superficial y desorganizada.

**CAPÍTULO 3: VALIDACIÓN DEL PROCESO DE PRUEBA DE LA DISTRIBUCIÓN CUBANA DE SOFTWARE LIBRE GNU/LINUX NOVA.**

---

Tabla 3.1 organización antes del proceso pruebas

<b>Organización</b>					
	Diseñar las pruebas	Determinar los recursos necesarios	Planificar las pruebas	Realizar todas las iteraciones de pruebas previstas	Analizar resultados de la pruebas
Antes	2	3	1	2	2.5

**Organización**

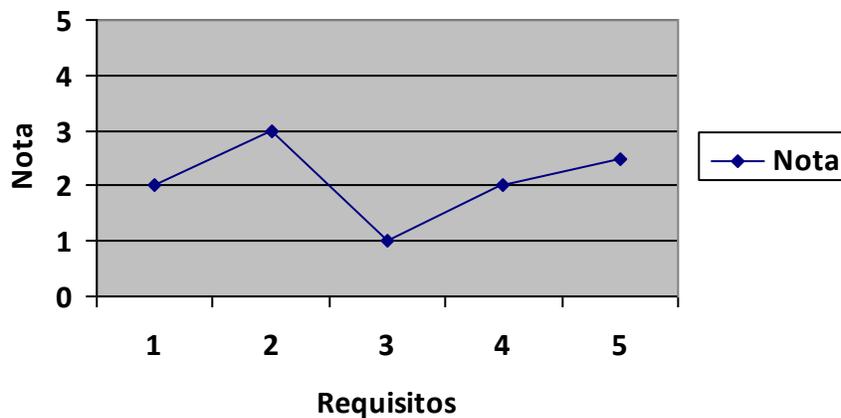


Figura 3.1 Organización antes de proceso de pruebas

Tabla 3.2 Capacitación antes del proceso de pruebas

<b>Capacitación del personal</b>						
	Diseñar las pruebas	Determinar los recursos necesarios	Planificar las pruebas	Capacitar al personal	Realizar todas las iteraciones de pruebas previstas	Analizar resultados de la pruebas

CAPÍTULO 3: VALIDACIÓN DEL PROCESO DE PRUEBA DE LA DISTRIBUCIÓN CUBANA DE SOFTWARE LIBRE GNU/LINUX NOVA.

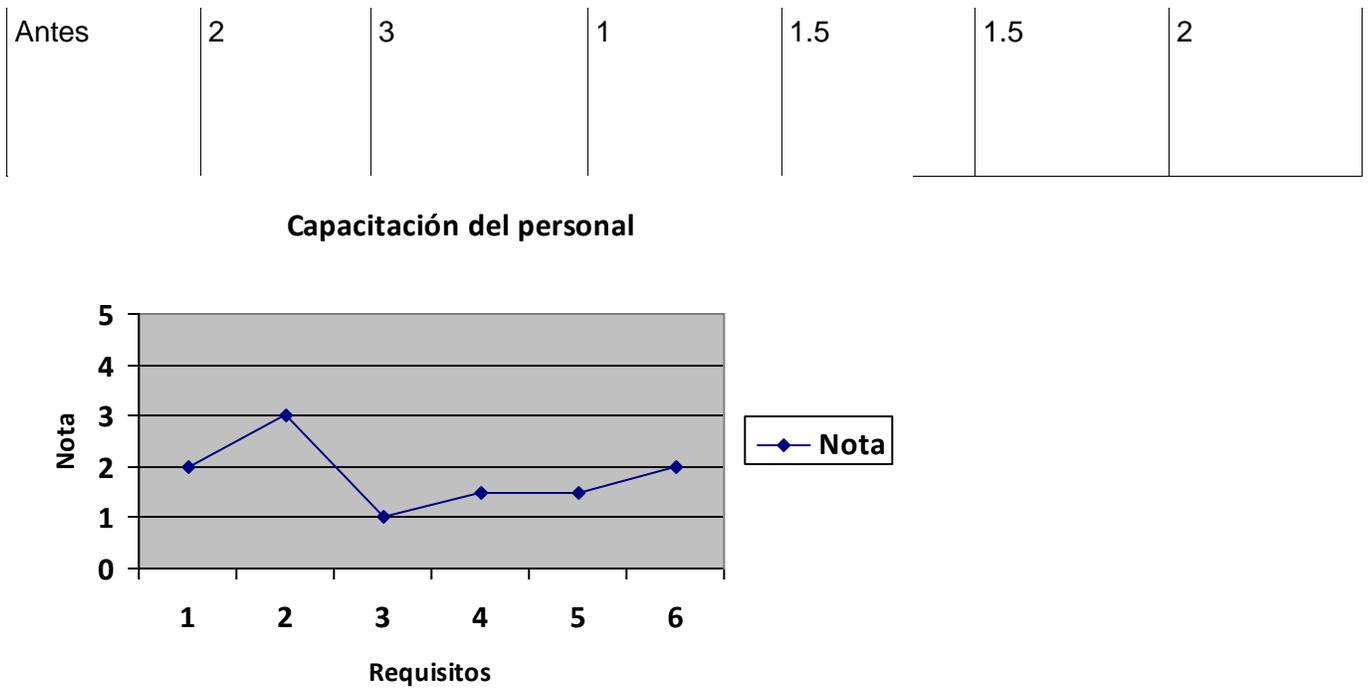


Figura 3.2 Capacitación antes de proceso de pruebas

Tabla 3.3 Calidad del producto antes de proceso de pruebas

<b>Calidad</b>				
	Diseñar las pruebas	Planificar las pruebas	Realizar todas las iteraciones de pruebas previstas	Analizar resultados de la pruebas
Antes	3	2	2	2

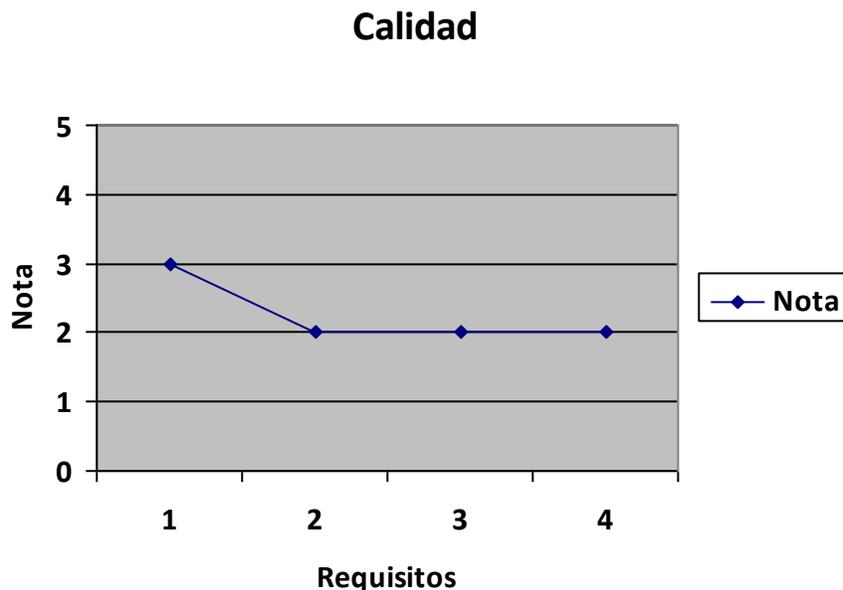


Figura 3.3 Calidad antes de proceso de pruebas

### Después

Nova es un producto con mayor calidad debido a que el proceso de pruebas le ha permitido a los desarrolladores conocer en cada ciclo de prueba cuáles eran las fallas principales de cada versión, además de facilitarles la corrección de las mismas, lo cual ha contribuido a que Nova sea más estable que sus versiones anteriores, elevando su prestigio dentro y fuera de la Universidad de las Ciencias Informáticas (UCI), lo anterior ha consolidado su comunidad de usuarios.

La utilización de la herramienta de pruebas automáticas Phoronix Test Suite es una gran ayuda para los desarrolladores pues les permite comparar el rendimiento de distintas versiones de Nova, otros sistemas operativos y distribuciones GNU/Linux permitiéndoles conocer las mejoras en cuanto a rendimiento que esta versión brinda y aportando datos que sirvan como base para estudiar el comportamiento de futuras versiones.

Después de puesto en práctica el proceso de pruebas para la última versión de Nova la tabla de evaluación de los requisitos quedaría de la siguiente forma:

Tabla 3.4 organización después del proceso pruebas

CAPÍTULO 3: VALIDACIÓN DEL PROCESO DE PRUEBA DE LA DISTRIBUCIÓN CUBANA DE SOFTWARE LIBRE GNU/LINUX NOVA.

Organización					
	Diseñar las pruebas	Determinar los recursos necesarios	Planificar las pruebas	Realizar todas las iteraciones de pruebas previstas	Analizar resultados de la pruebas
Después	4	4	5	5	5

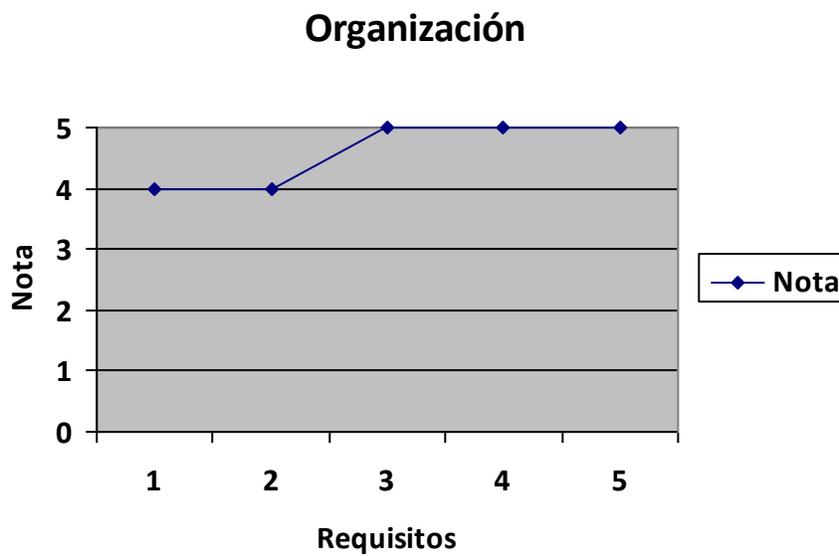


Figura 3.4 Organización después de proceso de pruebas

Tabla 3.5 Capacitación después del proceso de pruebas

Capacitación del personal						
	Diseñar las pruebas	Determinar los recursos necesarios	Planificar las pruebas	Capacitar al personal	Realizar todas las iteraciones de pruebas previstas	Analizar resultados de la pruebas

CAPÍTULO 3: VALIDACIÓN DEL PROCESO DE PRUEBA DE LA DISTRIBUCIÓN CUBANA DE SOFTWARE LIBRE GNU/LINUX NOVA.

Después	4	5	5	4	4	5
---------	---	---	---	---	---	---

**Capacitación del personal**

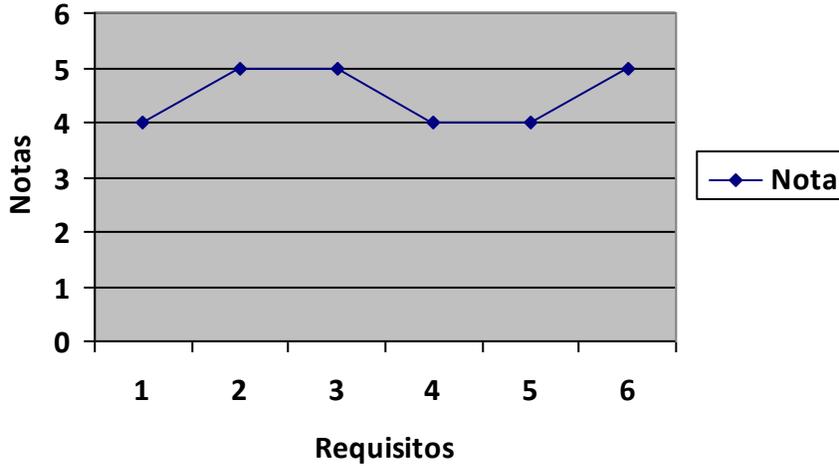


Figura 3.5 Capacitación después de proceso de pruebas

Tabla 3.6 Calidad del producto después de proceso de pruebas

Calidad				
	Diseñar las pruebas	Planificar las pruebas	Realizar todas las iteraciones de pruebas previstas	Analizar resultados de la pruebas
Después	4	5	4	5

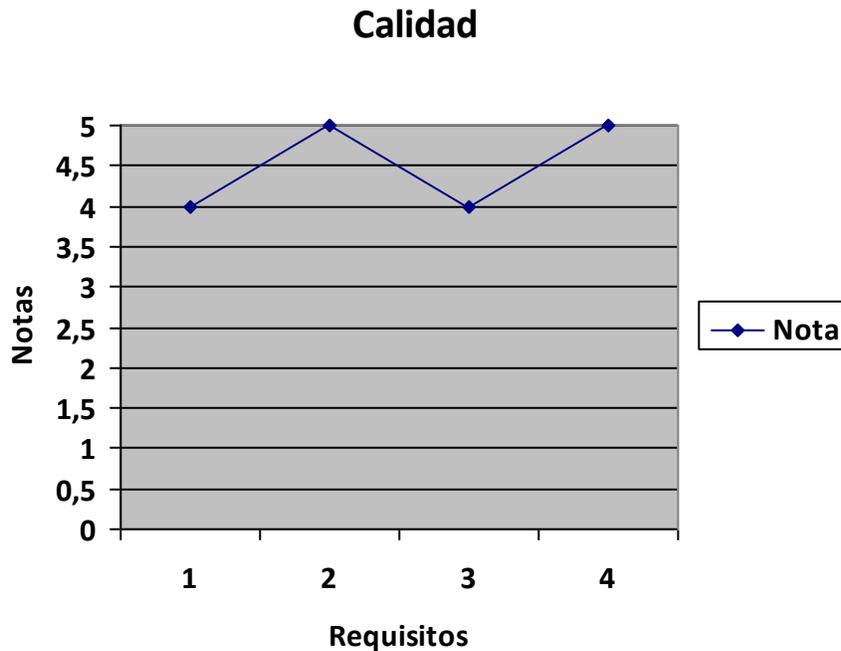


Figura 3.6 Calidad después de proceso de pruebas

### 3.2.2 Método de Recopilación de la Información [21].

La elección del método depende de la estrategia de recopilación de información, la precisión necesaria, el punto de recopilación y la formación del encuestador. Los principales métodos de recopilación de información son:

- **Cuestionarios:** El encuestador plantea preguntas directas a los encuestados, los cuestionarios son formularios que rellenan los encuestados solos. Los cuestionarios pueden entregarse en mano o enviarse por correo y recogerse posteriormente o devolverse en un sobre prefranqueado.
- **Entrevistas:** Una entrevista es una conversación dirigida con un propósito específico que utiliza un formato de preguntas y respuestas. En la entrevista se necesita conocer las opiniones de los entrevistados acerca del estado actual del sistema.
- **Observación:** La observación proporciona información de primera mano en relación con la forma en que se llevan a cabo las actividades. Las preguntas sobre el uso de documentos, la manera en la que se realizan las tareas y si ocurren los pasos específicos como se pre-

establecieron, pueden contestarse rápidamente si se observan las operaciones.

Para la realización de este trabajo se utilizó la entrevista debido a que el cuestionario es un método para realizarse cuando las personas a las que necesitas encuestar se encuentran en ubicaciones dispersas, se está haciendo un estudio preliminar y se desea medir la opinión general antes de que se determine el rumbo que tomara el proyecto de sistemas entre otros. La observación es para procesos más profundos y además se debe tener una visión bastante detallada del proceso de pruebas cuando se esté realizando y la entrevista nos permite conocer todas las opiniones de las personas que realizaron el proceso de pruebas y realizar un profundo análisis general de cuán factible es el mismo.

### **Cinco pasos para preparar una entrevista**

1. Leer los antecedentes.
2. Establecer los objetos de la entrevista.
3. Decidir a quién entrevistar.
4. Preparar al entrevistado.
5. Decidir el tipo de preguntas y la estructura.

### **Tipos de preguntas**

#### **Preguntas abiertas**

Describe las opiniones del entrevistado para responder. La respuesta puede ser de dos palabras o dos párrafos.

#### **Preguntas cerradas**

Las preguntas cerradas de una entrevista limitan las opiniones de los encuestados.

#### **Los Sondeos**

Permiten abordar en las preguntas para conseguir respuestas más detalladas.

Existen además diferentes maneras de organizar la entrevista, combinando el tipo de preguntas como son:

## CAPÍTULO 3: VALIDACIÓN DEL PROCESO DE PRUEBA DE LA DISTRIBUCIÓN CUBANA DE SOFTWARE LIBRE GNU/LINUX NOVA.

---

**Uso de una estructura de pirámide:** El entrevistador empieza con preguntas, a menudo cerradas, muy detalladas. Posteriormente, el entrevistador extiende los temas permitiendo preguntas abiertas y respuesta más generalizadas.

**Uso de una estructura de embudo:** Comienza con preguntas abiertas, y luego limitar las posibles respuestas utilizando preguntas cerradas.

**Uso de una estructura de diamante:** Esta estructura implica empezar de una manera muy específica, después se examinan los aspectos generales y finalmente se termina con una conclusión muy específica.

### ➤ Entrevista

La entrevista se realizará en forma de preguntas especialmente cerradas y estructurada en forma de diamante [anexo IV].

### **Objetivo de la entrevista:**

Conocer el progreso de Nova luego de la realización del proceso de pruebas.

### **Quién Entrevistar:**

Probadores del proyecto Nova.

Luego de la realización de la entrevista al conjunto de probadores del proyecto Nova se llegó a la conclusión de que:

Nova es un producto con mayor calidad debido a que el proceso de pruebas le ha permitido corregir de manera más eficiente y rápida los errores y las no conformidades encontrados, lo cual ha contribuido a que Nova sea más estable que sus versiones anteriores, elevando su prestigio dentro y fuera de la Universidad. El proceso permitió dirigir el personal de calidad hacia todos los frentes sin que esto menguara la calidad de las pruebas, pues se logró un alto grado de especialización en los probadores convirtiéndolos en especialistas de su rama de forma tal, que la calidad y velocidad a la hora de realizar las pruebas se incrementara de un ciclo de pruebas a otro, y de una versión a otra. El cual se evidencia en el gráfico de la Figura 3.7:

CAPÍTULO 3: VALIDACIÓN DEL PROCESO DE PRUEBA DE LA DISTRIBUCIÓN CUBANA DE SOFTWARE LIBRE GNU/LINUX NOVA.

---

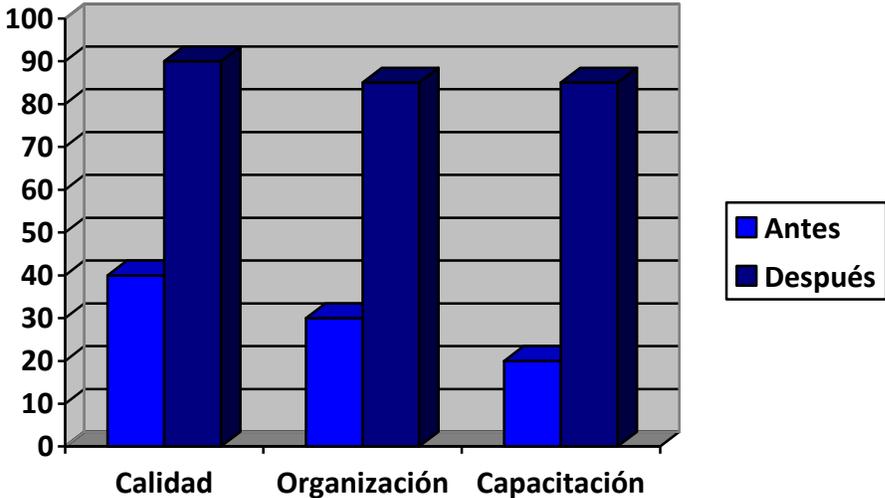


Figura 3.7 Proceso de pruebas

Después de analizar la situación Antes y Después de existir este proceso de pruebas para la distribución de software libre GNU/Linux Nova se evidencia que la calidad, y el nivel de experiencia del personal ha aumentado. El proyecto Nova cuenta con una mayor organización además de una elevada calidad del producto liberado.

### **Conclusiones generales**

En este trabajo de diploma con el estudio de cómo realizar un proceso de pruebas de manera eficaz y teniendo en cuenta los problemas existentes en cuanto a la calidad y el prestigio de Nova se llevaron a cabo las siguientes acciones:

- Se propuso un proceso de pruebas para la distribución cubana de software libre GNU/Linux Nova con el objetivo de aumentar la calidad del producto y organizar y desarrollar las pruebas que se le realizan al software de manera más eficiente.
- Se realizó y validó el proceso de pruebas para Nova teniendo como resultado la eficiencia y calidad al utilizar el mismo.
- Se comprobó que la utilización de un proceso de pruebas bien definido aumenta la calidad y el prestigio de Nova.
- Con la elaboración del proceso de pruebas se obtuvo una mayor organización y planificación a la hora de realizar las pruebas a Nova.

### **Recomendaciones**

Después de concluido el proceso de prueba y conocer las facilidades que brinda a los desarrolladores de los sistemas una adecuada detección de errores, en el momento oportuno, se recomienda:

- Que la(s) persona(s) encargada(s) de desempeñar el rol de probadores dentro del equipo de desarrollo realice todas las pruebas pertinentes al rol que le ocupa.
- Realizar un plan de capacitación para el personal que no esté capacitado a la hora de realizar un proceso de pruebas
- Realizar la traducción de la herramienta Phoronix Test Suite al idioma español para una mejor interacción con la herramienta.

**Referencias bibliográficas**

1. FSF.GNU operating System, 2011. Disponible [<http://www.gnu.org/>]
2. [Www.linux.org/dist](http://www.linux.org/dist)
3. Pablo Pazos Gutiérrez. Ingeniería en sistemas de software, 2010. Disponible: [[ingenieria\\_de\\_Software.blogspot.com](http://ingenieria_de_Software.blogspot.com)]
4. 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
5. Departamento de ingeniería de software.Conferencia-7 pruebas de software, 2010.
6. PER90] Perry, D. y Kaiser, G. "Adequate testing y object-oriented programming". Testing Object-Oriented Software. IEEE Computer Society, 1998. pp 11-17.
7. [KIT96] Kit, E. "Software testing in the real world. Improving the process", ACM.
8. [PRE98] Pressman, R. "Ingeniería del software. Un enfoque práctico". Cuarta Edición. McGraw-Hill/Interamericana de España. 1998.
9. [PRE96] Pressman, R. "Ingeniería del software. Un enfoque práctico". Segunda Edición. McGraw-Hill/Interamericana de España. 1996.
10. [PRE99] Pressman, R. "Ingeniería del software. Un enfoque práctico". Quinta Edición. McGraw-Hill/Interamericana de España. 1999.
11. [PRE00] Pressman, R. "Ingeniería del software. Un enfoque práctico". Sexta Edición. McGraw-Hill/Interamericana de España. 2000.
12. V-Modell XT Portal, Fundamentals of the V-Modell, Version 1.2.1, <http://www.v-modell-xt.de>. Visited: 2008.
13. Boehm, B. W., "A spiral model of software development and enhancement, "Computer , vol.21, no.5, pp.61-72, May 1988
14. Mónica Ma Albo Castro. Propuesta de proceso de pruebas generales para la distribución GNU/Linux Nova. Universidad de las Ciencias Informáticas, 2010. 17.
15. Towner, Mitch. Testing. Wikipedia Enero 2010. [Consultado 29 Marzo 2010]. Disponible

- .[<<https://wiki.ubuntu.com/Testing>>]
16. Trac.Dogtail, 2006.Disponible [<https://fedorahosted.org/dogtail/>]
  17. Michael Larabel.Phoronix Test Suite,2008 disponible [<http://www.phoronix-test-suite.com/>]
  18. Jean-Baptiste Lallement.Mago Una iniciativa de Pruebas de Escritorio ,2008 disponible[<http://mago.ubuntu.com>]
  19. Dayana, Yanisleydis. Automatización de pruebas de carga y estrés del Sistema de Información de Perforación de Pozos Petroleros (SIPP). Universidad de las Ciencias Informáticas. Habana, 2010,84.
  20. Yusleydy. Capacitación orientada a la calidad del software, sistema de gestión de conocimientos para pruebas de software. Universidad de la Ciencias Informáticas. Habana, 2008. 120.
  21. Rafael. Recopilación de información: métodos interactivos, 2010. Disponible [<http://www.monografias.com/trabajos61/recopilacion-informacion-metodo-interactivo>]

**Bibliografía consultada**

- Boehm, B. W., "A spiral model of software development and enhancement, "Computer , vol.21, no.5, pp.61-72, May 1988
- Dayana, Yanisleydis. Automatización de pruebas de carga y estrés del Sistema de Información de Perforación de Pozos Petroleros (SIPP). Universidad de las Ciencias Informáticas. Habana, 2010,84.
- Departamento de ingeniería de software.Conferencia-7 pruebas de software, 2010.
- FSF.GNU operating System, 2011. Disponible[<http://www.gnu.org/>]
- Jean-Baptiste Lallement.Mago Una iniciativa de Pruebas de Escritorio ,2008 disponible[<http://mago.ubuntu.com/>]
- Kit, E. "Software testing in the real world. Improving the process", ACM.
- Michael Larabel. Phoronix Test Suite,2008 disponible[<http://www.phoronix-test-suite.com/>]
- Mónica Ma Albo Castro. Propuesta de proceso de pruebas generales para la distribución GNU/Linux Nova. Universidad de las Ciencias Informáticas, 2010. 17.
- Pablo Pazos Gutiérrez. Ingeniería en sistemas de software, 2010. Disponible: [[ingenieria\\_de\\_Software.blogspot.com](http://ingenieria_de_Software.blogspot.com)]
- Perry, D. y Kaiser, G. "Adequate testing y object-oriented programming". Testing Object-Oriented Software. IEEE Computer Society, 1998. pp 11-17.
- Pressman, R. "Ingeniería del software. Un enfoque práctico". Cuarta Edición. McGraw-Hill/Interamericana de España. 1998.
- Pressman, R. "Ingeniería del software. Un enfoque práctico". Quinta Edición. McGraw-Hill/Interamericana de España. 1999.
- Pressman, R. "Ingeniería del software. Un enfoque práctico". Segunda Edición. McGraw-Hill/Interamericana de España. 1996.
- Pressman, R. "Ingeniería del software. Un enfoque práctico". Sexta Edición. McGraw-Hill/Interamericana de España. 2000.
- Rafael. Recopilación de información: métodos interactivos, 2010. Disponible

[\[http://www.monografias.com/trabajos61/recopilacion-informacion-metodo-interactivo\]](http://www.monografias.com/trabajos61/recopilacion-informacion-metodo-interactivo)

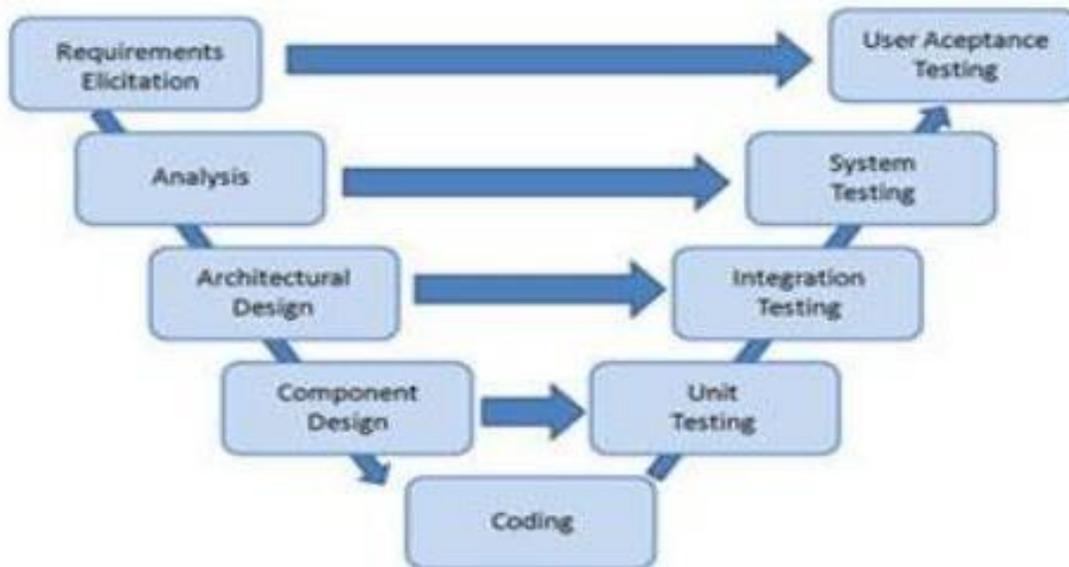
- Towner, Mitch. Testing. Wikipedia Enero 2010. [Consultado 29 Marzo 2010]. Disponible [.<https://wiki.ubuntu.com/Testing>](https://wiki.ubuntu.com/Testing)
- Trac.Dogtail, 2006.Disponible en[\[https://fedorahosted.org/dogtail/\]](https://fedorahosted.org/dogtail/)
- V-Modell XT Portal, Fundamentals of the V-Modell, Version 1.2.1, <http://www.v-modell-xt.de>. Visited: 2008.
- 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
- Yusleydy. Capacitación orientada a la calidad del software, sistema de gestión de conocimientos para pruebas de software. Universidad de la Ciencias Informáticas. Habana, 2008. 120.

---

## Anexos

### Anexo I

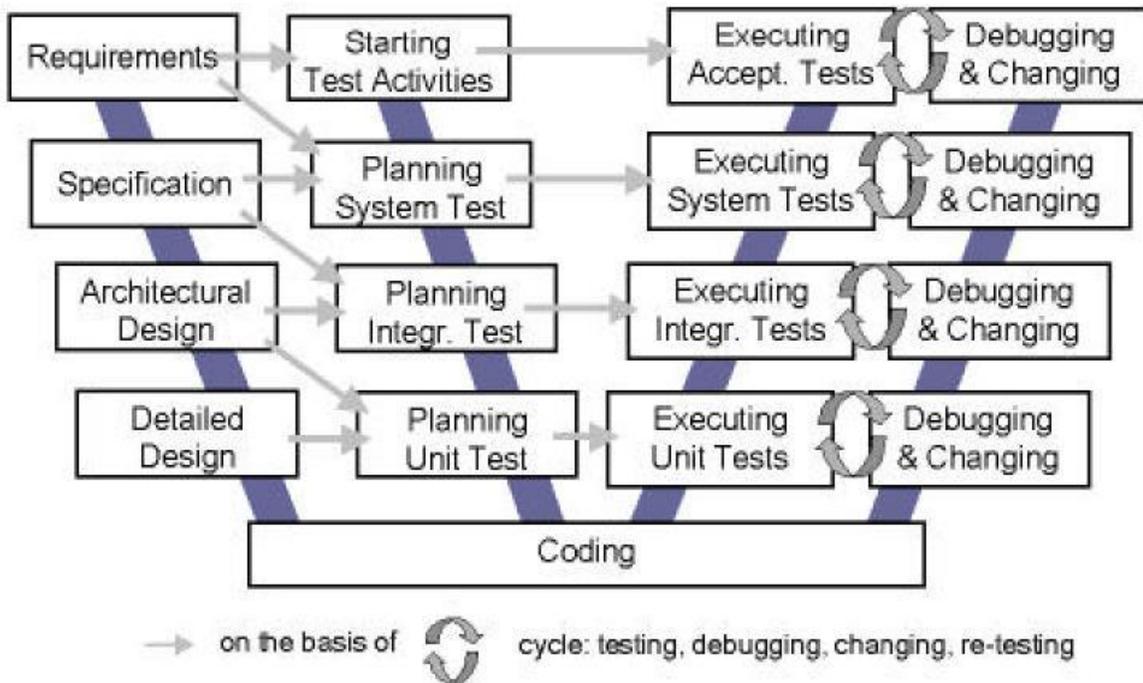
#### Modelos de desarrollo de software V



### Anexo II

#### Modelo de desarrollo de software W

## W - Model



### Anexo III

Entrevista a los desarrolladores del proyecto Nova

	A	B	C	D	E	F	G	H	I
1		Servidores	<u>Machin</u>	<u>Daniel</u>	clientes ligeros	base	promedio	Máximo 0,5	
2	Procesador	1	1	1	1	1	1	1	1
3	<u>Ram</u>	1	1	1	1	1	1	1	1
4	rendimiento sistema	1	1	1	1	1	1	1	1
5	Disco duro	0	1	1	1	1	0,8	1	1
6	interfaz GTK	1	1	1	1	1	1	1	1
7	Gráficos Mesa	1	1	1	0	1	0,8	1	1
8	sistema en portátiles	1	1	1	0	1	0,8	1	1
9	servidor X	1	1	1	0	1	0,8	1	1
10	codificación audio y <u>video</u>	0	1	0	0	1	0,4	0	1
11	compresión de archivos	1	1	0	0	1	0,6	1	1
12	capacidad criptográfica	0	1	0	0	1	0,4	0	1
13	juegos libres	0	1	0	0	0	0,2	0	0
14	Múltiples procesadores	0	1	1	1	1	0,8	1	1
15	juegos privativos	0	1	0	0	0	0,2	0	0
16									
17									

	A	B	C	D
1	Aspectos Encuesta	Suites <u>Phoronix</u>		
2	1 Procesador	1,2,5,18,23,24		
3	2 <u>Ram</u>	1,2,4,21,		
4	3 Sistema	1,2,4,6,7,9,12,13,14,16,17,19,20,26,27,28,29,30,33,34,35,36,38		
5	4 Disco Duro	8,11,31		
6	5 Interfaz GTK		15	
7	6 Gráficos Mesa		22	
8	7 Sistema portátiles		25	
9	8 Servidor X		34,39	
10	9 Codificación de audio y <u>video</u>		1	
11	10 Compresión de archivos		2	
12	11 Criptografía		5	
13	12 Juegos Software libre		13	
14	13 Multiprocesador		24	
15	14 juegos de código cerrado		12	
16				
17				
18				

### **Anexo IV**

#### **Entrevista a los probadores del proyecto Nova**

➤ ¿Los casos de pruebas diseñados para el proceso realizado se corresponden con los requisitos definidos para el producto? De no ser positiva su respuesta indique el porqué.

\_\_\_ Si \_\_\_ No \_\_\_ No se

Porque \_\_\_\_\_  
\_\_\_\_\_

➤ ¿Las indicaciones dadas en cada caso de prueba sobre cómo llevarlo a cabo se entiende sin

necesidad de consultar dudas a quien las diseñó?

Si       No

➤ ¿Cree usted que el flujo de actividades del proceso realizado es correcto?

Si       No

3.1 De no ser positiva su respuesta indique si eliminaría alguno de los propuestos o que agregaría

Actividades propuestas (señale para eliminar)

Agregar

Pruebas al LiveCD

Prueba de Instalación

Pruebas del Gestor de paquetes

Pruebas de actualización

Pruebas de uso diario

Pruebas de rendimiento

➤ ¿Luego de realizar cada ciclo de pruebas el equipo de desarrollo corrige los errores y no conformidades en el tiempo requerido? De no ser positiva su respuesta indique el porqué.

Si       No       No se

Porque \_\_\_\_\_  
\_\_\_\_\_

➤ ¿Considera usted que existe buena comunicación entre el equipo de pruebas y los desarrolladores? De no ser positiva su respuesta indique el porqué.

Si       No       No se

Porque \_\_\_\_\_  
\_\_\_\_\_

➤ ¿Observó mejoras en cada una de las versiones de la distribución Nova en cada ciclo de pruebas? De no ser positiva su respuesta indique el porqué.

Si       No       No se

---

Porque

---

## **Anexo V**

### **Manual de usuario de la herramienta de pruebas automáticas Phoronix Test Suite**

#### **Instalación del Phoronix Test Suite**

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

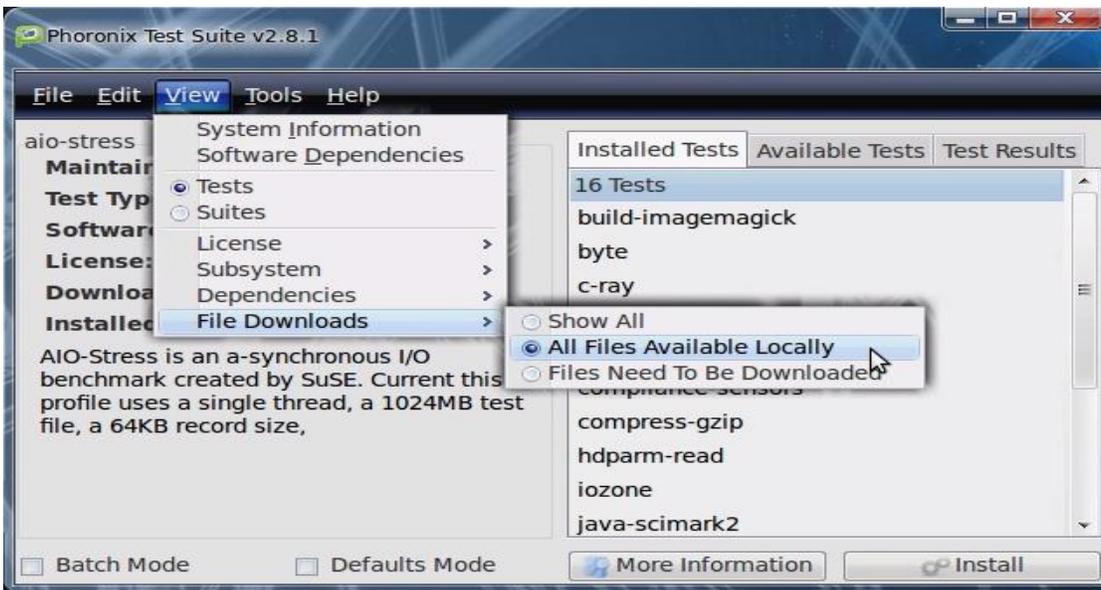
1. Abrir el menú de nova e ir a Añadir/Quitar software
2. Buscar Phoronix Test-suites mediante la pestaña Buscar
3. Instalar los paquetes `phoronix-test-suite_2.8.1_all`, `php-gtk_2.0.0-1nova2_i386` y `pecl-cairo_0.2.0-beta-1_i386`.

#### **Instalación de las pruebas de Phoronix**

Esta aplicación nos brinda un conjunto de 132 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.

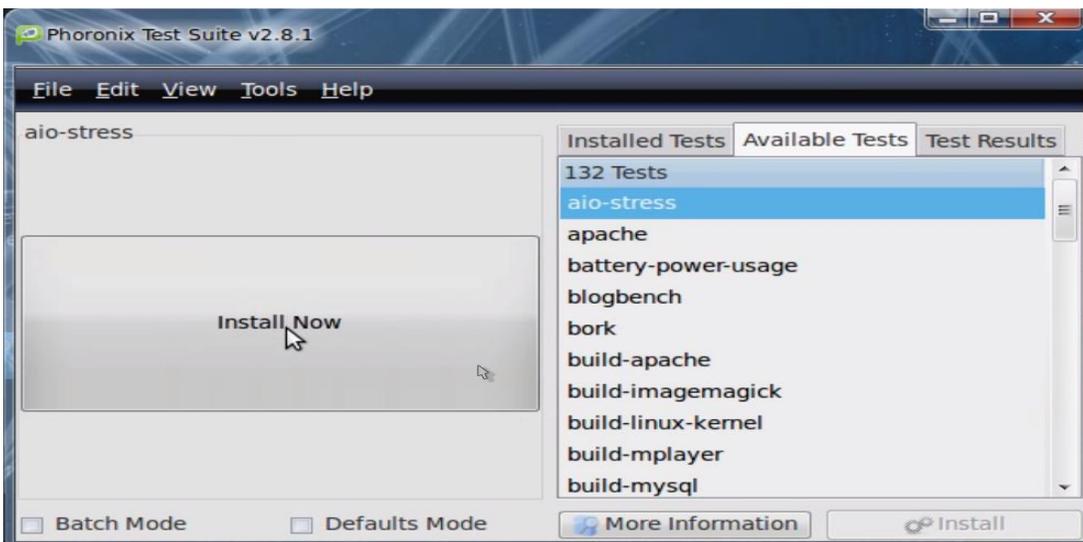
Para instalar estas pruebas una vez que han sido descargadas, se sigue un procedimiento más o menos estándar:

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".



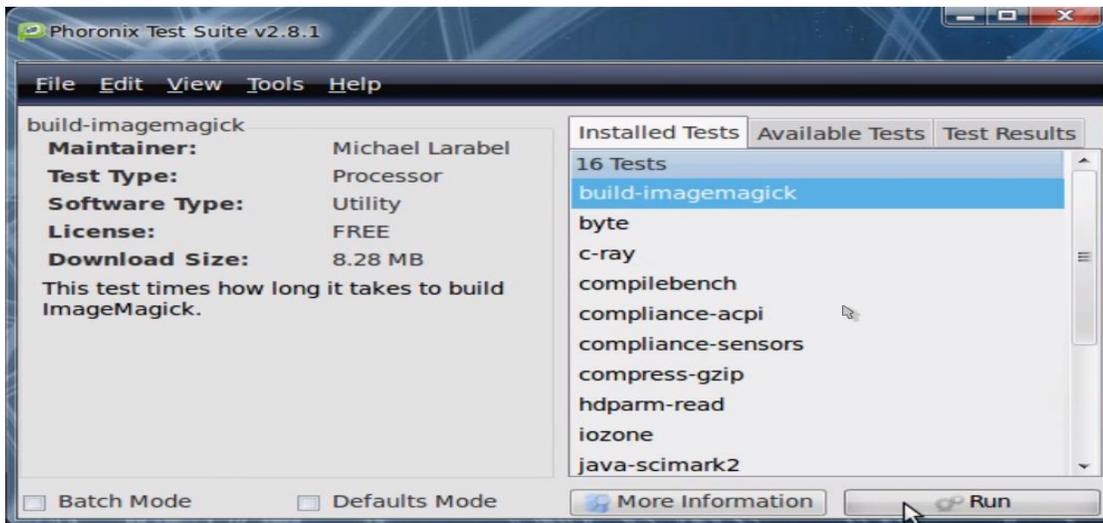
Se selecciona la pestaña "available tests" en la interfaz principal.

Se escoge la prueba a instalar y se presiona el botón "Install" y luego el boton "Install Now".



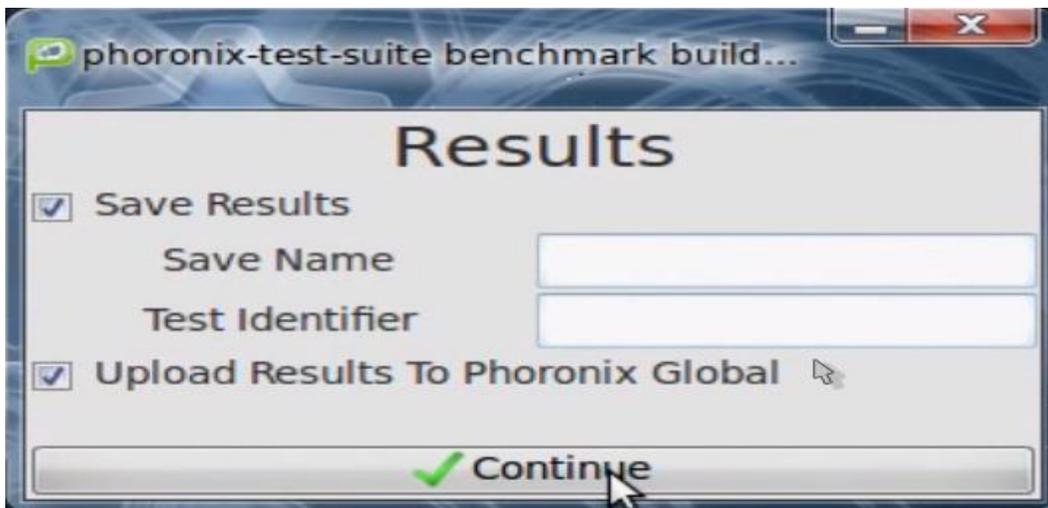
Las pruebas pueden ejecutarse desde la ventana principal de la aplicación

Se selecciona la pestaña "installed tests".



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

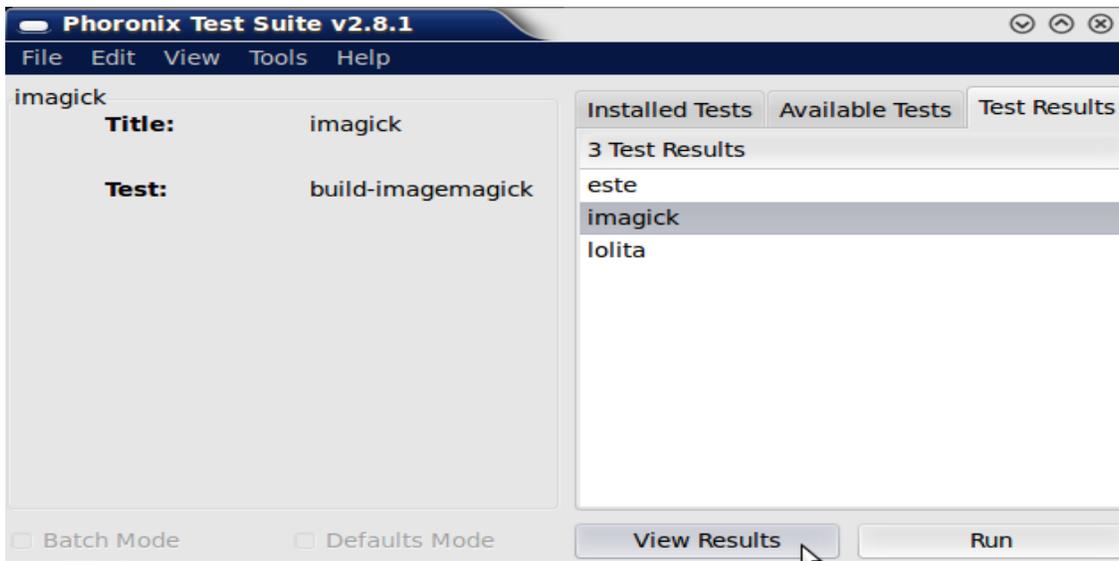
En el cuadro de diálogo que se muestra a continuación se coloca el nombre con el cual se guardarán los resultados de la prueba.



### Visualización de los resultados de las pruebas

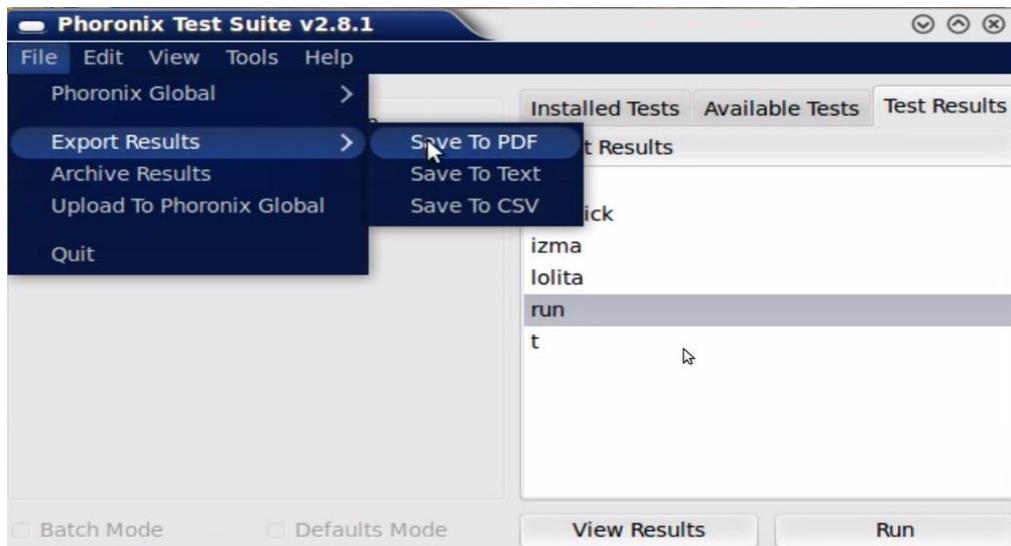
Seleccionar la pestaña "Test Results".

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.



### Glosario de términos

**Activos:** Es el conjunto de bienes tangibles o intangibles que posee una empresa.

**Benchmarking:** El benchmarking es un proceso sistemático y continuo para evaluar los productos, servicios y procesos de trabajo de las organizaciones que son reconocidas como representantes de las mejores prácticas, con el propósito de realizar mejoras organizacionales.

**Clientes (término computacional):** En informática, es un equipo o proceso que accede a recursos y servicios brindados por otro llamado servidor, generalmente de forma remota.

**Costes o costo:** Montante económico que representa la fabricación de cualquier componente o producto, o la prestación de cualquier servicio.

**Error conocido:** es el problema al cual se le han determinado las causas.

**HTTP:** (Hypertext Transfer Protocol, en español protocolo de transferencia de hipertexto).

**Infraestructura:** Conjunto de elementos o servicios que se consideran necesarios para el funcionamiento de una organización o para el desarrollo de una actividad.

**Plataforma:** La plataforma define un estándar alrededor el cual un sistema puede ser desarrollado. En informática una vez que la plataforma ha sido definida, se produce el software y el hardware apropiado para su uso. El término a menudo es usado como un sinónimo de sistema operativo.

**Proveedor:** es la empresa u organismo que proporciona los servicios solicitados por el cliente.

**Servidores:** En informática, un servidor es una computadora que, formando parte de una red, provee servicios a otras computadoras denominadas clientes.

**Solución:** es la propuesta de la solución definitiva al problema, es analizada con respecto a su: posible impacto, viabilidad y conveniencia.

**Releases:** Nueva versión de una aplicación informática.

**Usuarios:** las personas que utilizan el servicio.