

**Universidad de las Ciencias Informáticas**  
**Facultad 1**

# **ENTORNO DE ESCRITORIO DE NOVA 4.0**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS**

**Autor:** Juan Manuel Fuentes Rodríguez.

**Tutores:** MSc. Allan Pierra Fuentes  
Ing. Yunier Soler Franco.

La Habana  
2012

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 1 de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

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

Juan Manuel Fuentes

Rodríguez

Msc. Allan Pierra Fuentes

Ing. Yunier Soler Franco

\_\_\_\_\_

Firma del autor

\_\_\_\_\_

Firma del tutor

\_\_\_\_\_

Firma del tutor

## **AGRADECIMIENTOS**

A la Revolución por darme la oportunidad de hacerme un profesional.

A mis padres por haber luchado tanto por mí durante 24 años, sus esfuerzos no fueron en vano.

A mi mujer por darme ese bebé precioso, la inspiración necesaria para luchar por el resto de mis días, y por apoyarme a disociarme tanto de la tesis.

A mis familiares por estar siempre a mi lado.

A mis tutores Allan y Yunier por brindar su mayor esfuerzo para la correcta realización de la tesis.

A Daniel por apoyarme y estar siempre presente para aclarar mis dudas.

A mis amigos por estar siempre en el lugar indicado para ayudarme.

A todo el proyecto Nova por darme el honor de ser uno de ustedes.

A mis compañeros de aula.

A todas las personas que un modo u otro han aportado su grano de arena a esta tesis.

A los que se me olvidaron.

**Gracias**  
**Juan Manuel.**

## DEDICATORIA



*A mi bebé.*

## **RESUMEN.**

La migración de las aplicaciones privativas a estándares libres y de código abierto es uno de los pasos más importantes que da el país en aras de conseguir la soñada independencia y soberanía tecnológica. En este trabajo se presenta el Entorno de escritorio de Nova en su versión 4.0, el cual es una propuesta que pretende integrar el estilo visual del entorno de escritorio del sistema operativo Microsoft Windows con las funcionalidades y el alto grado de personalización que tienen los entornos de escritorios modernos de las distintas distribuciones de GNU/Linux. Este pretende dar solución a la necesidad de brindar un entorno de escritorio libre capaz de adaptarse a las experiencias y conocimientos previos de los usuarios que lo usen durante el proceso de migración, aumentando así el nivel de satisfacción y el placer de usar el sistema operativo, proporcionando mejoras en proceso de migración.

# Índice

INTRODUCCIÓN.....	8
FUNDAMENTACIÓN TEÓRICA DE LOS ENTORNOS DE ESCRITORIO.....	13
1.1. METÁFORA DE ESCRITORIO.....	13
1.2. USABILIDAD.....	16
1.2.1. Utilidad.....	17
1.2.2. Accesibilidad.....	17
1.2.4. Arquitectura de información.....	18
1.2.5. Experiencia de usuario.....	18
1.3. ENTORNOS DE ESCRITORIO.....	19
1.3.1. Entorno de escritorio de Microsoft Windows.....	20
1.3.2. KDE.....	21
1.3.3. Gnome.....	22
1.3.4. Gnome 2.30.....	23
1.3.5. Gnome 3.....	24
1.4. SESIONES.....	27
1.5. GESTOR DE VENTANAS.....	28
1.6. JUSTIFICACIÓN DE LA SELECCIÓN DEL ENTORNO DE ESCRITORIO BASE.....	29
1.7. ENTORNO DE DESARROLLO.....	30
1.7.1. Metodología de Desarrollo.....	30
1.7.2. Justificación de la selección de la metodología.....	32
1.7.3. Lenguajes de Programación.....	32
1.7.4. Librerías a utilizar.....	34
CONCLUSIONES PARCIALES.....	36
ANÁLISIS Y DISEÑO DEL ENTORNO DE ESCRITORIO DE NOVA 4.0.....	37
2.1. FASE INICIAL DEL PROCESO DE INGENIERÍA DE SOFTWARE.....	37
2.1.1. Concepción del sistema.....	37
2.1.2. Modelo de Dominio.....	39
2.1.3. Lista de Reserva del Producto (LRP).....	40
2.1.4. Historia de Usuarios y prototipos de interfaces de usuario.....	44
2.1.5. Lista de riegos.....	52
2.1.6. Metáfora del sistema.....	53
2.2. SEGUNDA FASE DEL PROCESO DE INGENIERÍA DE SOFTWARE.....	53
2.2.1. Tareas de Ingeniería.....	54
2.2.2. Diagrama de Componentes.....	56
2.2.3. Plan de entregas.....	56
CONCLUSIONES PARCIALES.....	57
VALIDACIÓN DEL ENTORNO DE ESCRITORIO DE NOVA 4.0.....	58
3.1. CASOS DE PRUEBA DE ACEPTACIÓN.....	58

3.1.1. Casos de Prueba de la Historia de usuario: HU-1.....	58
3.1.2. Casos de Prueba de la Historia de usuario: HU-2.....	60
3.1.3. Casos de Prueba de la Historia de usuario: HU-3.....	61
3.1.4. Casos de Prueba de la Historia de usuario: HU-4.....	61
3.1.5. Casos de Prueba de la Historia de usuario: HU-5.....	62
3.1.6. Casos de Prueba de la Historia de usuario: HU-6.....	63
3.1.7. Casos de Prueba de la Historia de usuario: HU-7.....	63
3.2. RESULTADOS OBTENIDOS.....	64
3.2.1. Acerca de las funcionalidades obtenidas.....	64
CONCLUSIONES PARCIALES.....	65
CONCLUSIONES GENERALES.....	66
RECOMENDACIONES.....	67
REFERENCIAS.....	68
BIBLIOGRAFÍA.....	70

## **INTRODUCCIÓN**

Cuba, con el objetivo de no quedarse atrás en el uso de las Tecnologías de la Información y las Comunicaciones (TIC), ha puesto su empeño en llevar la computación a las distintas esferas de la sociedad. Al ser incapaces de seguir el ritmo de las nuevas tecnologías, por las consecuencias nefastas para la economía de la situación política existente, por los altos precios en el mercado de las licencias y contratos con empresas extranjeras y bajo el riesgo que implican los productos informáticos privativos para la seguridad nacional se hace necesario una solución.

Conscientes de que la solución de este problema conlleva un profundo cambio en la tecnología y forma de pensar se está efectuando un proceso de migración a estándares abiertos con el objetivo de cubrir las carencias y eliminar las dependencias que existen en las empresas cubanas del software privativo. Un punto vital para la realización de este proceso es el lanzamiento al público de la primera versión oficial de Nova, una distribución de GNU/Linux cubana que tiene la misión remplazar a Microsoft Windows como software base en las estaciones de trabajo y de adecuarse a las necesidades del país y los usuarios comunes, un sistema operativo pensado por los cubanos y para los cubanos.

Desde su primer lanzamiento hasta ahora Nova ha traído como entorno de escritorio por defecto a Gnome<sup>1</sup>, un proyecto que desde sus primeras versiones se ha caracterizado por su estabilidad, escogido este por su alto nivel de personalización, lo que le permite al usuario final la adaptación del escritorio y sus aplicaciones a sus necesidades, y por la facilidad de uso de una interfaz intuitiva y amigable que influye en la independencia del usuario final de otras personas, posibilitando además mejoras en el tiempo de aprendizaje.

Para la primera versión del sistema operativo, Nova Baire, se logra un entorno de escritorio haciendo una personalización de Gnome con un diseño centrado en el usuario final. Este entorno tiene una estructura simple con el objetivo de atraer nuevos usuarios y con un escritorio similar al ambiente que se muestra en Microsoft Windows XP para minimizar los tiempos de ajuste en la migración. Pero la realidad dista mucho de lo imaginado. Aunque la distribución está orientada a las estaciones de trabajo se evidencia resistencia al cambio por parte de los usuarios que migran, algo natural en todo proceso de cambio. Está demostrado

---

1 GNU Network Object Model Environment.



que existe una fuerte dependencia del ambiente brindado por Microsoft Windows y sus herramientas y, en general, los usuarios no se sienten identificados con el producto que están usando ya que les cuesta trabajo adaptarse y aprender a trabajar con las herramientas que les brindaba este, todo esto unido al hecho de que existen problemas de incompatibilidad en algunas de las computadoras en las que el sistema operativo se instalaba (1). La motivación, como elemento primordial en el proceso de migración, hace mella en la evolución del aprendizaje.

Para la última versión oficial de la distribución se puede apreciar un entorno de escritorio más maduro, dándole al usuario muchas de las posibilidades que brinda la versión de Gnome con la que es liberada. Todo esto sin apartarse del objetivo principal del sistema operativo, la migración del país, por lo que seguía teniendo gran parecido al entorno de Microsoft Windows sin dejar a un lado la identidad visual del producto y se incluyeron nuevas funcionalidades que dan al usuario mejoras a la hora de la interacción con el entorno. Pero otra vez se evidencia resistencia ante este proceso, ya no solo por parte de los usuarios que migran, sino que también los usuarios ya empapados en el proceso de migración protestan al ver el cambio que da el entorno de escritorio que estaban acostumbrados a utilizar, por lo que nuevamente el proceso de migración se ve frenado por la falta de motivación y compromiso de los usuarios.

A esto se debe sumar que, cumpliendo con su misión en la migración, el entorno brinda una sola sesión por defecto parecida a Microsoft Windows, hecha así con el propósito de minimizar el choque ya que se le brinda un producto con el cuál puede trabajar basándose en sus conocimientos previos, aunque tenga herramientas diferentes. Esto a la vez le pone un límite al usuario ya que no siente la necesidad de aprender y extender sus conocimientos a otras áreas, restringiéndolo a su conocimiento actual. Además la práctica demuestra que existen otros usuarios con conocimientos más amplios sobre los entornos de escritorios de GNU/Linux y que pueden explotar mejor las posibilidades que brindan estos y se decantan por otras distribuciones que, al no estar enfocadas en un público tan específico como Nova, pueden brindar mejores funcionalidades. Esto dificulta el proceso de migración porque un usuario inexperto al ver a otro con un poco más de conocimientos en el tema trabajando en otro ambiente en vez de ayudar al proceso en el que ambos están enfrascados se puede oponer involuntariamente a dicho cambio.

Por otra parte, el proyecto Gnome, no ha hecho cambios en la arquitectura de información del popular entorno de escritorio en casi una década. A costa de la estabilidad se está usando un producto que, si bien es alabado en muchos sentidos, utiliza teorías antiguas en cuanto a la forma en la que muestra la información relevante al usuario, muchas de ellas obsoletas. En un mundo movido por interfaces de usuario, es un error obviar la evolución de los conceptos, metáforas y buenas prácticas para un correcto diseño de GUI<sup>2</sup>.

Considerando la realidad descrita se arriba a la siguiente **situación problemática**: *El entorno de escritorio del sistema operativo Nova no se adapta a los conocimientos y habilidades, de los usuarios que migran a este nuevo escenario tecnológico.*

Lo que trae como consecuencia el **problema científico**: *¿Cómo lograr que el entorno de escritorio del sistema operativo Nova para su versión 4.0, se adapte a los conocimientos y habilidades adquiridas por los usuarios involucrados en el proceso de migración?*

Para dar solución a dicha dificultad se establece que el **objeto de estudio** de esta investigación estará enmarcado en *los entorno de escritorios de GNU/Linux* centrando el **campo de acción** en *el entorno de escritorio de la distribución Nova.*

Por lo que se formula como **objetivo general** de este trabajo: *Desarrollar el entorno de escritorio de la versión 4.0 de Nova.*

Desprendiéndose de este los siguientes **objetivos específicos**:

1. *Analizar las principales características de los entornos de escritorio de las distribuciones de GNU/Linux.*
2. *Modificar el entorno de escritorio libre que mejor cumpla con dichas características, para que sea la base tecnológica sobre la que se construirá la solución propuesta.*
3. *Analizar, diseñar e implementar una variante del entorno de escritorio para reducir la resistencia al cambio de los usuarios que provienen de entornos MS Windows.*
4. *Verificar la solución propuesta mediante la generalización y retroalimentación de las autoridades para la informatización y la migración a SWL en Cuba y la comunidad cubana de SWL.*

---

2 GUI: Graphic User Interfaces (Interfaces gráficas de usuario o GUI por sus siglas en inglés).

En esta investigación se arriba a la siguiente **idea a defender**:

*Con el desarrollo de un entorno de escritorio para Nova 4.0, que haga un tratamiento diferenciado de los usuarios que lo usen basándose en sus conocimientos y experiencias previas, aumentará el nivel de aceptación de estos en el proceso de migración.*

Para cumplir con los objetivos propuestos se proponen las siguientes **Tareas de Investigación**:

1. *Investigación de las principales características y objetivos de la usabilidad, la experiencia de usuario y la metáfora de escritorio.*
2. *Investigación sobre cómo se ponen de manifiesto dichas características en los entornos de escritorio de las distribuciones de GNU/Linux.*
3. *Investigación acerca del funcionamiento de los entornos de escritorios de GNU/Linux.*
4. *Selección del entorno de escritorio que mejor cumpla dichas características.*
5. *Modificación del entorno de escritorio seleccionado para que cumpla los requerimientos visuales de Nova.*

Para dar solución a las tareas propuestas se formulan los siguientes **Métodos de Investigación**:

1. *Métodos empíricos:*
  - a. *Observación: Este método permitirá, mediante la constatación visual, chequear el estado actual de los entornos de escritorio modernos.*
2. *Métodos teóricos:*
  - a. *Histórico-Lógico: Este método permitirá seguir la evolución de los entornos de escritorio desde su surgimiento, lo que servirá para profundizar el conocimiento sobre estos.*
  - b. *Analítico-Sintético: Este método permite la división del problema en varios subproblemas que se analizan por separado para luego comprender el problema completo.*

El presente documento, para dar cumplimiento a los objetivos de este, se encuentra dividido en los siguientes capítulos:

- **Capítulo 1 Fundamentación teórica de los entornos de escritorio:** En este capítulo se estudian los elementos relacionados con los entornos de escritorio. Describe además las buenas prácticas que se deben tener en cuenta para el buen diseño de estos entornos, así como las diferentes

metáforas y paradigmas que existen.

- **Capítulo 2 Análisis y diseño del entorno de escritorio de Nova 4.0:** En este capítulo se realiza el análisis y diseño del entorno de escritorio para Nova 4.0 y el entorno de escritorio para la migración.
- **Capítulo 3 Validación del entorno de escritorio de Nova 4.0:** En este capítulo se exponen el proceso de pruebas para la validación del entorno de escritorio de Nova.

## FUNDAMENTACIÓN TEÓRICA DE LOS ENTORNOS DE ESCRITORIO.

En este capítulo se habla sobre las bases necesarias para el correcto entendimiento de los conceptos que se deben tener en cuenta antes de la construcción del entorno de escritorio para la distribución de GNU/Linux Nova en su versión 4.0. En este se definen los términos sobre el entorno y la metáfora de escritorio y sobre algunos conocimientos básicos a tener en cuenta para su estudio. Adicionalmente se tratan algunos aspectos acerca del ambiente de desarrollo de los procesos utilizados.

### 1.1. Metáfora de Escritorio.

Según la Real Academia de la Lengua Española, una metáfora es la *“aplicación de una palabra o de una expresión a un objeto o a un concepto, al cual no denota literalmente, con el fin de sugerir una comparación (con otro objeto o concepto) y facilitar su comprensión”*. Básicamente es la abstracción, en una palabra o frase, de un objeto de la vida real. En la informática estas ayudan a los usuarios inexpertos a entender el funcionamiento de un concepto o tecnología innovadora permitiéndoles asociar objetos de su vida cotidiana y sus funciones con los nuevos sistemas y sus funcionalidades. Esto permite un desarrollo más eficiente de los nuevos conocimientos y a la vez crean en los usuarios una nueva base de conocimientos que en el futuro podrían ayudar (o dificultar) la entrada de nuevos estándares.

La metáfora de escritorio es un conjunto de métodos y conceptos que pretenden acercar el entorno gráfico de los sistemas operativos a la comprensión de los usuarios. Esta tiene su primera aparición en la computadora comercial Xerox PARC en 1981 en la que aparecen elementos muy comunes en los entornos de escritorio actuales (doble clic para abrir los iconos, ventanas que se solapaban, etc.) y a través de los años ha sido popularizada por los fabricantes de sistemas operativos más influyentes del mundo. Esta hace una comparación entre el monitor y el buró de trabajo en el cual el usuario puede tener un conjunto de “objetos” en una disposición conveniente para su trabajo y uso, proponiendo un espacio (o programa) en el sistema operativo en el que el o los usuarios puedan guardar y acceder a las carpetas y documentos de forma rápida, sencilla y natural. Para lograr un perfecto entendimiento por parte de los usuarios y que pudieran estos asociarlo con sitio real en sus vidas cotidianas (metafóricamente), se le llama a este lugar el escritorio y está pensado para ser el punto de partida de toda interacción entre el usuario y el entorno de escritorio, ya que es el lugar donde es accesible toda la información del usuario.

En este también deben estar presentes otros elementos de la vida cotidiana de los usuarios como herramientas de cálculo (calculadora), para el trabajo con los documentos (editores de texto), una papelera donde almacenar los objetos desechables, etc.

A pesar de que su nombre la restringe al escritorio del usuario, la metáfora se expande por otras esferas del entorno gráfico de los sistemas operativos. Un ejemplo de esto son los manejadores de archivos que están implementados de forma tal que se parezcan a los gabinetes de la vida real. Estos muestran un estante general (navegador de archivos) en cual se pueden encontrar carpetas (directorios con iconos que los permite asociarlos con elementos de la vida real) en las que se pueden encontrar documentos (archivos) y otras carpetas. Con los directorios y archivos se pueden realizar las mismas acciones que con sus homónimos en la realidad, pueden ser abiertos, leídos, copiados, movidos de directorio, renombrados y desechados. Otra aplicación de la metáfora es en el conjunto de carpetas y documentos abiertos en el escritorio, mediante la cual se puede ver la información con las que se está trabajando en el momento, brindando la posibilidad de poder cambiar el documento de trabajo por otro y cerrarlos cuando se haya terminado con estos.

Con el desarrollo de nuevas tecnologías, la metáfora de escritorio ha sido ampliada con diversas implementaciones que la han enriquecido. Es posible en el escritorio virtual tener una imagen de fondo que finge como el portarretratos del buró, buzones de correo, herramientas de dibujo, reproductores de sonido y radio, agendas, etc. Otros elementos de los entornos de escritorio como los menús y docks<sup>3</sup> no tienen un significado objetivo en la realidad pero ayudan a mantener un alto nivel de usabilidad que es más importante que seguir al pie de la letra lo que dicta la metáfora.

Aunque en sus inicios fue alabada por las personas, puesto que las introdujo en un nuevo paradigma solo con los conocimientos previos del mundo real, luego de más de 30 años de uso (teniendo en cuenta el profundo cambio tecnológico) es normal que surjan limitaciones e incongruencias en la metáfora de escritorio.

Una limitante es que se usa la pantalla de monitor como única superficie para mostrar y acceder a la información, entrando en conflicto con los procesos que se llevan a cabo en la oficina tradicional, puesto

---

3 Elemento de interfaz gráfica de usuario que permite a los usuarios iniciar, cambiar y monitorizar aplicaciones.

que no es necesario limpiar el escritorio para poder acceder a los documentos que están en un archivador o gabinete. Este problema es corregido en los entornos escritorios modernos en los que se puede acceder al navegador de archivos a través del menú principal.

El hecho que el usuario tenga un único escritorio para trabajar es una restricción que presenta la metáfora, cuando aumenta el número de documentos abiertos se dificulta el trabajo del usuario a la hora de cambiar entre aplicaciones; el usuario de una oficina tradicional puede tener varios burós donde sentarse a trabajar. En las distribuciones de GNU/Linux modernas es posible tener varios escritorios virtuales ayudando a la organización de las aplicaciones abiertas por el usuario. Aunque existe la posibilidad de crear estos espacios de trabajo en los sistemas operativos Windows no se ha hecho muy popular y es de poco uso.

Otra limitación importante a destacar en la metáfora es que la forma de almacenamiento de los entornos basados en la metáfora es un sistema de archivos jerárquico en el cual el usuario, partiendo de una carpeta raíz (root en los sistemas operativos de tipo Unix y Mi PC en Windows), puede crear un número de carpetas y ficheros que permite una navegación simple por el sistema de archivos. Este es un proceso rígido porque no tiene en cuenta la alteración de la importancia de los documentos y el desarrollo del proceso de aprendizaje del usuario con el paso del tiempo.

Algunos especialistas han hecho diseños de algunas aplicaciones que rompen por completo con la dinámica propuesta por la metáfora de escritorio, adentrándose en diversos campos de la psicología para descubrir mejores formas de entendimiento y percepción por parte de los usuarios a los diferentes programas.

Un buen ejemplo es Lifestream, un paradigma que indica que se debe manejar la información como “*un flujo, ordenado en el tiempo, de documentos que funcionan como un diario en la vida electrónica*” (2). Este empieza con los documentos del pasado, moviéndose adelante se encuentran los documentos más recientes y pasando por la fecha actual y siguiendo hacia el futuro se encuentran documentos que serán utilizados (recordatorios, citas, etc.). El diseño se basa en el planteamiento de que la información reciente tiene mayor importancia para el usuario que la antigua. La herramienta permite la creación de un documento vacío que se deposita en el momento actual del Lifestream (al inicio) que no tiene por qué ser

nombrado pues lo importante no es la etiqueta que se usa para denotar el documento sino el tiempo en el que se crea, por lo que el proceso de guardar es transparente al usuario. También permite copiar un documento del Lifestream y lo pone en el presente como el último utilizado; al igual que buscar un archivo, proceso en el que se crea un substream con los archivos encontrados y por último permite hacer un resumen del Lifestream.

## **1.2. Usabilidad.**

Desde que los ordenadores penetraron en los acontecimientos cotidianos, convirtiéndose en parte importante del modo de vida de las personas y un equipo indispensable para ellas, se viene hablando de las interfaces “amigables” refiriéndose a aquellas que cumplían mejor las expectativas de aquellos individuos que las usaban. Luego surge el concepto de usabilidad por la necesidad de definir un área del conocimiento que se encargara de abordar los atributos de calidad de las interfaces. Este término se refiere a la cualidad de un producto de brindar cierta facilidad de uso al usuario. Formalmente la usabilidad es *“el grado en que un producto puede ser usado por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso específico”* (4). La usabilidad es a su vez relativa al uso, a los usuarios y a los objetivos específicos con que fue diseñada. Esto se debe a que todo producto o aplicación se crea con el objetivo de satisfacer a un área específica de la sociedad, por lo que sí es usable para unos pocos y no necesariamente tiene que serlo para el resto. Esto mismo pasa con los objetivos para los que fue hecha cierta aplicación, puede ser usable para ciertas actividades no así para otras.

No debe verse la usabilidad como un mero concepto abstracto ni como una información cualitativa de la calidad de uso de un producto. La idea que se persigue con esta va mucho más lejos. La usabilidad puede ser medida y evaluada como lo que verdaderamente es: una variable de calidad. La norma ISO-9126 postula un conjunto de métricas mediante las cuales puede medirse la usabilidad y la calidad de uso de cierto producto informático. Esta puede ser medida mediante un listado de componentes o variables. Entre estos componentes se encuentran:

- **Facilidad de aprendizaje:** este componente engloba los diferentes atributos a evaluarse cuando los usuarios interactúan por primera vez con el producto y mide cuán fácil es para ellos el uso de este



(o al menos las funciones más básicas) en ese primer encuentro.

- Eficiencia: este sirve como indicador para medir el tiempo que se demora un usuario en realizar las tareas disponibles en el producto una vez que este ya ha aprendido a usarlo.
- Calidad de ser recordado: este componente mide la calidad que tiene el producto de ser usado eficientemente una vez que se ha dejado de interactuar con él por un tiempo.
- Eficacia: este concepto mide la cantidad de errores que puede cometer un usuario en el proceso de utilización del producto, la gravedad de estos y la facilidad que existe de enmendarlos.

Como se puede ver el proceso de medición de la usabilidad no es un flujo unilateral donde lo más importante es el diseño de producto. El usuario es una parte fundamental de este, siendo este el centro de toda medición de calidad que se presente. Es por esto que una de las pruebas más importantes para valorar la usabilidad de un producto es el “test de usuarios”, en cuál se le somete al producto a una prueba frente a sus usuarios reales. Este test brinda una proporción de como el producto es aceptado por este y permite medir los diferentes componentes expuestos anteriormente. Además es muy útil a la hora de medir la calidad de un rediseño con respecto al diseño anterior.

A continuación se muestran otros conceptos que son necesarios para lograr una correcta usabilidad.

### **1.2.1. Utilidad.**

La utilidad es el conjunto de características que tiene un producto que permite satisfacer las necesidades de la persona que lo usa. Por lo tanto es imposible hablar de usabilidad de un producto sin mencionar la utilidad que este conlleva, que es lo que el usuario en realidad busca. Pero que sean conceptos diferentes no implica que estén separados. Según el Informe de APEI sobre la Usabilidad *“la usabilidad representa el grado en que el usuario puede explotar la utilidad”* y *“un producto o aplicación será usable en la medida en que el beneficio que se obtenga de usarlo”*. Al analizar estas citas se comprueba el carácter bidireccional que tiene la relación entre estos conceptos.

### **1.2.2. Accesibilidad.**

La accesibilidad es la característica de un producto de ser usado (accedido) por el mayor número de personas posible. Esta definición tiene en cuenta factores dentro de la audiencia que impiden que la aplicación no pueda ser entendida o usada por una parte de los usuarios. Dentro de estos factores se

encuentran la discapacidad de los usuarios (limitaciones físicas y visuales), cultura, religión, género, ingresos, conocimientos, etc. La accesibilidad es un aspecto de vital importancia para lograr una correcta usabilidad puesto que el producto no llegaría a ser “amigable” para la totalidad de los usuarios. Un producto que no es accesible no es usable y viceversa. Cabe destacar que este concepto no entra en conflicto con la característica relativa de la usabilidad expuesta anteriormente que dice que el producto está hecho para usuarios específicos. La accesibilidad, a pesar del carácter universal que posee, se encarga de que la calidad del producto llegue a cada uno de los usuarios para los que fue previsto, no para el resto de ellos.

#### **1.2.4. Arquitectura de información.**

La arquitectura de información representa la forma en que está expuesta la información en el diseño del producto. Como resultado se define que esta es *“la actividad y resultado de organizar, clasificar, ordenar, estructurar y describir los contenidos , con el fin de que sus usuarios puedan satisfacer sus necesidades informativas con el menor esfuerzo posible ”*(3). De esto se intuye que la arquitectura de información es la base estructural de la aplicación o producto, su esqueleto.

#### **1.2.5. Experiencia de usuario.**

La medición de la usabilidad es un proceso que está expuesto a errores. El usuario es una persona, por lo que sus experiencias están sujetas a muchos factores como las emociones y los sentimientos hacia el producto y su estado de humor a la hora de realizar las tareas, es por esto que no es un proceso exento de dificultades. Las emociones hacia el producto son producidas en el momento de utilización por el usuario. Por su parte el humor es un estado previo a la hora de uso, por lo que juega un papel primordial ya que puede influir decisivamente en las emociones hacia el producto. Al contrario de las emociones y el estado de humor, los sentimientos son evocados en el usuario por un uso previo del producto o de otros similares. Es por esto que surge el término de experiencia de usuario (User eXperience), el cual tiene su origen en el campo del Marketing. *“La Experiencia del Usuario representa un cambio emergente del propio concepto de usabilidad, donde el objetivo no se limita a mejorar el rendimiento del usuario en la interacción –eficacia, eficiencia y facilidad de aprendizaje–, sino que se intenta resolver el problema estratégico de la utilidad del producto y el problema psicológico del placer y diversión de su uso”* (5). La

UX no limita su campo solamente a la interacción usuario-aplicación, sino que expande el concepto de usabilidad para entender el porqué de las emociones del usuario al momento de dicha interacción. Por último destacar que muchos autores sugieren que la experiencia de usuario no solo abarca la respuesta emocional del usuario hacia el producto sino que también sus emociones hacia la empresa o el proveedor de este.

### **1.3. Entornos de escritorio.**

Las computadoras, en sus inicios, brindaban los resultados de sus cálculos a los usuarios mediante el uso de tarjetas perforadas por lo que tenían un grupo reducido de usuarios en centros de investigaciones. Luego se populariza el uso de los monitores para mostrar caracteres de texto, y mediante el uso de teclados se pudo introducir comandos a ejecutar en estas, ampliando así el número de usuarios que las utilizaban. Pero no fue hasta la introducción de las interfaces gráficas de usuario que se extiende el uso de los ordenadores en las esferas comerciales y diarias de la sociedad, la llegada de la metáfora de escritorio fue toda una revolución tecnológica.

En este marco surgen los entornos de escritorio, un conjunto de software especializado en brindar al usuario una interacción cómoda y amigable con el sistema operativo de la computadora. Básicamente un entorno de escritorio es la capa que permite la comunicación entre el usuario y el sistema operativo, siempre teniendo en cuenta la simplicidad y la facilidad.

A pesar de que existe una amplia gama de entornos de escritorio, todos tienen elementos en común. Algunos de estos elementos son (6):

- **Manejador de ficheros:** programa que tiene como función principal poder brindar una navegación por los ficheros de un determinado sistema de archivos. Este debe permitir crear, abrir, copiar, mover y eliminar directorios y archivos y opcionalmente administrar la función de arrastrarlos y soltarlos.
- **Panel:** es un programa que reserva un espacio en la pantalla para mostrar un conjunto de herramientas indispensables para la correcta utilización del escritorio como el menú de aplicaciones, lista de ventanas, accesos directos a programas, área de notificaciones, reloj, administración de red y volumen, configuración del teclado y un conjunto de miniaplicaciones para

mejorar la interacción del usuario con el escritorio (applets).

- **Manejador de ventanas:** es parte de un sistema de ventanas que organiza las ventanas en una pantalla. Es responsable de mover y cambiar el tamaño de las ventanas, y las demás funciones comunes a todas las aplicaciones.
- **Gestor de sesiones:** es el programa que administra las sesiones del entorno de escritorio. Entre sus responsabilidades se encuentran ejecutar los programas iniciales al comienzo de la sesión y brindar las funcionalidades de cerrar sesión, cambiar de usuario, apagar, reiniciar, hibernar y suspender.

A continuación se muestran algunas de las características de algunos de los principales entornos de escritorio.

### **1.3.1. Entorno de escritorio de Microsoft Windows**

Al ser el único entorno de escritorio soportado por Microsoft Windows, es el entorno más usado del mundo. Con un diseño centrado en el usuario, provee gran sencillez, rapidez y elegancia en las operaciones que brinda. En su última versión, Windows 7, se mejoró notablemente el diseño gráfico, puliendo notablemente los errores que cometieron en su predecesor, Windows Vista. Al igual que este, Este trae como gestor de transparencia y efectos visuales a Windows Aero, el cual brinda entre sus funcionalidades el Windows Flip (Alt+Tab), Windows Flip 3D (Super+Tab), Aero Peek (miniaturas en la barra de tareas) y los efectos sobre las ventanas. Algunos elementos importantes de este entorno son:

- **Panel:** tiene un retocado aspecto visual. Este contiene el menú de inicio, el cual provee acceso al usuario a las aplicaciones favoritas e instaladas con un cajón para buscarlos, acceso a las principales bibliotecas del usuario y con accesos directos a las aplicaciones administrativas.
- **Barra de Tareas:** este elemento se encuentra contenido en el panel, luego del menú de inicio. Tiene un mejorado diseño en el que se agrupa las aplicaciones por iconos mostrando una previsualización de cada una de ellas. En esta se muestran también las aplicaciones favoritas dando la posibilidad de mostrar un Jump List (archivos recientes de esa aplicación) y permitiendo anclar (y desanclar) las aplicaciones a la barra de tareas.
- **Área de notificaciones:** está incluido en la parte derecha del panel. Está compuesto por el reloj, el

indicador para la configuración de red y sonido y un icono para mostrar los mensajes importantes, dando la posibilidad de agrupar las notificaciones e indicadores en el menú de este icono ahorrando espacio en el panel. Al final muestra un botón para mostrar el escritorio.

- Gestor de Ventanas: el gestor de ventanas de Windows 7 es Desktop Windows Manager (DWM) el cual permite la realización de las diferentes acciones con las ventanas y que saca provecho de la integración con Aero para lograr la transparencia en la barra de título y bordes de las ventanas y los efectos visuales.
- Navegador de Archivos: el navegador de archivos es el Explorer. Este tiene un diseño centrado en la simplicidad y vistosidad de sus componentes, incorporando informaciones importantes en el mismo icono (como la capacidad de una partición y una previsualización de los elementos contenidos en una carpeta).
- Gadgets: estos son un conjunto de miniaplicaciones disponibles desde la versión Vista, solo que ahora se pueden mover por todo el escritorio, brindándole al escritorio de Windows un mayor número de funcionalidades.

El entorno de Microsoft Windows 7 provee un Panel de Control que permite al usuario realizar una amplia gama de configuraciones. A pesar de esto, el escritorio es casi estático, el panel y el menú de usuario solo acepta unas pocas configuraciones (casi las mismas que en Windows XP), el explorador de archivos tiene una apariencia casi invariable y solo existen la posibilidad de adicionar al panel unas pocas miniaplicaciones.

### **1.3.2. KDE.**

KDE<sup>4</sup> es un entorno de escritorio que basa su ideología en la personalización de sus componentes. La mayor parte de los elementos de KDE son configurables a distintos niveles, desde su gestor de ventanas hasta los controles (botones, menús, etc.) aceptan temas. El proyecto fue iniciado en 1996 y su primera versión expuesta al público dos años después. Este entorno de escritorio tiene los siguientes componentes:

- Menú principal: llamado Kickoff está dividido en cinco pestañas principales. Primero aparece la de

---

4 K Desktop Environment (Entorno de Escritorio K)

Favoritos (brindando acceso a las aplicaciones preferidas por el usuario), luego viene la de Aplicaciones (que da acceso a las aplicaciones instaladas), el panel del Computador (acceso a los archivos y soportes extraíbles), el panel de Usados Recientemente y un panel para las opciones de apagado y cierre de sesión. Además provee una caja de búsqueda para la fácil localización de las aplicaciones.

- Navegador de Archivos: existe una amplia variedad de navegadores de archivos con gran calidad para KDE. Konqueror, uno de los más usados, hace de gestor de archivos, de navegador web y de menú de aplicaciones. Dolphin es otro navegador de archivos que destaca por su sencillez. Este ofrece previsualización para los archivos seleccionados en el panel derecho, así como un gran número de configuraciones para su comportamiento e interfaz.
- Plasma Widget: esta es sin dudas la característica más significativa de KDE. Es una herramienta que une el panel de KDE (Kicker), el escritorio (Kdesktop) y los widgets en un solo componente. Brinda la posibilidad de adicionar una amplia gama de miniaplicaciones (widgets o plasmoides) al panel y al escritorio, destacándose por un logrado diseño y buenos efectos visuales.

Debido a esta última funcionalidad, KDE posee un excelente estilo visual que, junto con la amplia gama de efectos visuales y configuraciones, hacen de este uno de los entornos de escritorios más usados por los usuarios de GNU/Linux.

### **1.3.3. Gnome.**

El proyecto Gnome provee un entorno de escritorio intuitivo y atractivo, poniendo un gran énfasis en la simplicidad, usabilidad y eficiencia. El proyecto fue iniciado por los mexicanos Miguel de Icaza y Federico Mena como una alternativa a KDE, lanzando su primera versión en marzo de 1999. El escritorio Gnome es bastante configurable: se pueden configurar los menús, los iconos, las tipografías, el fondo, el protector de pantalla, el tema, el administrador de ventanas, sonido, la interacción con las ventanas y muchos otros detalles de acuerdo al gusto del usuario. Actualmente se encuentra disponible en 166 idiomas. Después de casi 9 años de desarrollo de las versiones 2.X fue lanzada la versión 3.0 en Abril del 2011, la cual introdujo un nuevo paradigma en el área de los entornos de escritorio.

#### 1.3.4. Gnome 2.30.

Sin dudas uno de los entornos de escritorios más estables y configurables. Gnome 2.30 permite al usuario la adaptación del escritorio a sus necesidades, brindando la posibilidad de cambiar casi todos los elementos que lo componen, ya sea por alguna herramienta o editando manualmente los archivos de configuración. Este está compuesto por los siguientes elementos:

- Panel: el software que se encarga de la gestión de los paneles es gnome-panel. Este tiene un alto nivel de configuración, por lo que a pesar de su aspecto rústico inicial es posible dejarlo a la medida del usuario cambiándole la cantidad de paneles mostrados, el fondo para cada uno de ellos, el tamaño, el ocultamiento, etc. Existen también un sin número de miniaplicaciones para añadir al panel, convirtiéndolo en uno de los componentes más funcionales en el escritorio de Gnome.
- Navegador de archivos: desde la versión 2.0 de Gnome, Nautilus ha sido el navegador de archivos oficial del entorno de escritorio. Este visualiza los archivos locales y remotos (Samba,<sup>5</sup> SSH<sup>6</sup>, FTP<sup>7</sup>, WebDav<sup>8</sup>, etc.) a través de Gnome Virtual File System (GVFS), permitiendo la previsualización por iconos y por contenido. Permite la navegación por pestañas y permite la asociación de emblemas a los ficheros, permitiendo el uso de marcadores para una mejor localización de los archivos más utilizados. Este también es extensible mediante complementos.
- Gestor de Ventanas: el gestor de ventanas por defecto es Metacity, el cual está implementado en GTK+<sup>9</sup> y aunque este está integrado a Gnome puede ser usado en otros entornos de escritorio (al igual que Gnome 2.X puede utilizar otros gestores de ventana). Este gestor acepta temas dándole algunas opciones de configuración al usuario.

Es importante decir de este que su interfaz principal no ha cambiado prácticamente desde la versión 2.0, enfocándose en aumentar la estabilidad y en proporcionar mejoras de usabilidad en sus componentes

---

5 Implementación libre del protocolo de archivos compartidos de Microsoft Windows.

6 Secure Shell (Intérprete de Órdenes Segura).

7 File Transfer Protocol (Protocolo de Transferencia de Archivos)

8 Edición y versionado distribuidos sobre la web.

9 Conjunto de bibliotecas multiplataforma para desarrollar interfaces gráficas de usuario.

principales. A pesar de esto es uno de los entornos de escritorio preferidos por los usuarios en las distribuciones de GNU/Linux. Cabe destacar que este entorno ha sido prácticamente abandonado por Gnome, decidiendo solamente darle soporte, sin hacer ninguna actualización importante, puesto que han puesto casi todo su esfuerzo en el desarrollo de Gnome 3 y Gnome Fallback.

### **1.3.5. Gnome 3.**

En abril del 2011, Gnome lanza la versión 3.0 del popular entorno de escritorio tras cinco años de espera. Esta versión es completamente independiente de sus versiones anteriores, puesto que su desarrollo se hizo implementando desde cero. De los elementos básicos que contienen sus versiones anteriores, Gnome 3 solo mantiene el navegador de archivos, aunque este también sufrió de un largo proceso de rediseño. Sin duda alguna la novedad más significativa de Gnome 3 es Gnome Shell, un reemplazo para el panel de las versiones anteriores.

#### **Gnome Shell.**

Gnome Shell es un programa hecho con el objetivo de reemplazar el panel tradicional del entorno de escritorio Gnome y actúa como gestor de composición de este. Está escrito en los lenguajes de programación C y JavaScript, usando las bibliotecas Clutter para lograr un conjunto de componentes y animaciones de gran calidad visual. Este toma ventaja de la aceleración gráfica por hardware de los ordenadores modernos para brindar una mayor sencillez, facilidad y elegancia a las operaciones de los usuarios, haciendo hincapié en la naturalidad con la que estos pueden llegar a realizarlas y apostando por la limpieza y simplicidad del escritorio.

El panel de Gnome Shell está situado en la parte superior del escritorio y contiene algunos elementos de los paneles tradicionales. Este contiene (de izquierda a derecha) el botón de Actividades que muestra el Overview, un botón que muestra la aplicación que está enfocada en el momento (permite cerrar dicha aplicación y muestra una animación de espera para cualquier aplicación que se ejecute mediante el Shell), un reloj con calendario incluido e integración con el calendario de Evolution, un conjunto de indicadores (teclado, accesibilidad, batería, red, sonido, etc.) y el menú del usuario que permite acceder y modificar los datos de la cuenta, configuración del sistema y brinda las diferentes opciones de apagado y salida de la sesión.



El Overview es el espacio que se le brinda al usuario para poder acceder a las aplicaciones que se están ejecutando y las disponibles, generalizando en un solo componente la función que cumplen la lista de ventanas y el menú principal en otros entornos. En el centro se muestra una vista previa de cada una de las ventanas abiertas por el usuario, permitiendo cerrarlas y cambiarles el tamaño la miniatura mediante la rueda del ratón. En su parte izquierda se encuentra el dash (o dock), una herramienta que lista las aplicaciones favoritas del usuario y las que están abiertas, brindando un menú para cada aplicación que permite abrir una nueva ventana y ancla (o quita) la aplicación al dash, también se permite arrastrar las miniaturas del centro del Overview para añadirlas a los favoritos. A la derecha se encuentra el selector de áreas de trabajo dinámicas al cuál se le pueden arrastrar las miniaturas de la vista previa para cambiar la ventana correspondiente al área de trabajo señalada. En la parte superior tiene un selector que permite cambiar entre las aplicaciones instaladas y la lista de ventanas abiertas. Por último, en el extremo superior derecho se encuentra un cajón para realizar búsquedas en las aplicaciones instaladas, contactos y archivos utilizados recientemente.

El área de notificación fue separada del panel y puesta en la parte inferior de la pantalla y tiene como novedad que ahora es posible acceder a las conversaciones abiertas de la mensajería instantánea desde el mismo lugar en el que son comunicadas, quitando el proceso de abrir otra ventana para poder responder a los mensajes.

Gnome Shell está diseñado para ser una referencia en cuanto a usabilidad: es fácil de usar puesto que con solo presionar una tecla es accesible la lista de ventanas, programas instalados, aplicaciones favoritas y el selector de áreas de trabajo, la eficiencia del usuario al realizar las operaciones de búsqueda de los programas es mucho mayor que utilizando el menú clásico (solo teclear la parte inicial del nombre) y sobre todo muy fácil de recordar. En cuanto a la utilidad, soluciona de forma magistral el problema del entorno de escritorio. Está traducido a más de 60 idiomas, tiene las preferencias de accesibilidad en la parte de los indicadores del panel y con soporte para sistemas de escrituras de derecha a izquierda, resolviendo así el problema de la accesibilidad.

A pesar de los notables éxitos de Gnome Shell es evidente que deben existir dificultades que deben ser limadas sobre la marcha como el poco grado de configuración que tienen el panel y sus componentes,

siendo la única posibilidad la instalación de extensiones que aumenten las funcionalidades brindadas.

### **Extensiones de Gnome Shell**

*“El sistema de extensiones de GNOME Shell está diseñado para brindar alto grado de potencialidad a las partes de la interfaz de GNOME gestionadas por el Shell. Simplemente carga código JavaScript y CSS arbitrarios. Esto brinda a los desarrolladores una manera de hacer y compartir muchos tipos de cambios sin necesidad de parchear el código fuente original y recompilarlo, para luego encontrar una manera de redistribuirlo” (8).*

Las extensiones son la única forma que tiene el usuario de modificar su apariencia de Gnome Shell por defecto. Estas son un código en JavaScript que agregan funcionalidades al Shell y están hechas por la comunidad o por la Fundación Gnome. La estructura básica de una extensión es:

- `extension.js`: fichero con el código fuente de la extensión. Desde la versión 3.2 de Gnome Shell esta debe tener como mínimo las funciones `init(metadata)`, `enable()` y `disable()` que son las funciones que llama el Shell para inicializarlas, habilitarlas y deshabilitarlas eliminando la necesidad de reiniciarlo cuando se cambie el estado de una extensión.
- `stylesheet.css`: fichero que contiene el css necesario para el tema de la extensión.
- `metadata.json`: fichero que guarda datos adicionales de la extensión (`uuid`, nombre, descripción y versión del shell para la cual fue construida la extensión, dirección de los archivos de idioma, página del proyecto, etc.).

Gnome Shell cuenta con la aplicación LookingGlass, el depurador y la herramienta de inspección integrada al sistema, teniendo como objetivo convertirse en el FireBug del Shell (9). Esta está integrada por cuatro paneles principales:

- **Evaluador**: es un el modo interactivo de evaluación de JavaScript. Se puede copiar un código en este lenguaje y este será evaluado.
- **Ventanas**: muestra un resumen de las ventanas abiertas.
- **Errores**: este es el visor básico de errores y logs del shell.
- **Extensiones**: es una lista de todas las extensiones instaladas en el sistema. Permite abrir la carpeta contenedora y la página web de cada extensión.

Además cuenta con una herramienta llamada Selector, que permite seleccionar visualmente cualquier elemento en la escena del shell e inserta el código de este en el panel del Evaluador.

### Arquitectura de Gnome Shell.

Gnome Shell está hecho con un conjunto de componentes que, teniendo como base a OpenGL, hacen que el software pueda explotar las posibilidades de aceleración por hardware que tienen los computadores modernos.

A continuación se muestran todos los elementos que componen a Gnome Shell:

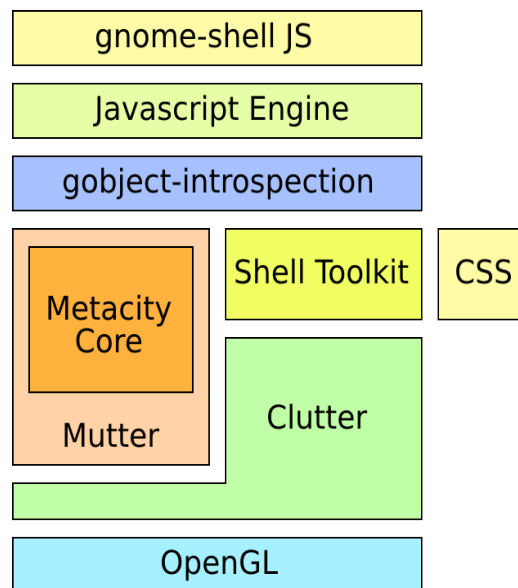


Ilustración 1: Arquitectura de Gnome Shell

## 1.4. Sesiones.

En los entornos de escritorios una sesión es el tiempo que transcurre desde que el usuario se registra en la interfaz de autenticación del entorno hasta el instante en el que sale de este (9). Estas se encargan de levantar los programas básicos para que se pueda ejecutar correctamente el entorno de escritorio y cerrarlos una vez que se termine la sesión. En Gnome el software que se encarga de manejar las sesiones es gnome-session. Este pasa por un conjunto de pasos para lograr la iniciación del entorno de

escritorio (10):

- Inicio: durante esta fase se produce un proceso de configuración interna e inicialización. Es aquí donde se leen y procesan los archivos .desktop que se encuentran en /usr/share/xsessions/
- Inicialización: durante esta fase, el gestor de sesiones inicia los servicios básicos de la sesión. Cualquier servicio que se propone crear el entorno (incluyendo las variables de entorno de ajuste) se debe ejecutar durante esta fase.
- Gestor de Ventanas: en esta fase se iniciarán aplicaciones que provean administración para ventanas y composición (gestores de ventana). A partir de este paso cualquier aplicación que se ejecute se tiene que registrar como cliente en el administrador de sesiones.
- Panel: es en esta etapa que se lanza Gnome Shell (en versiones anteriores se lanzaba el gnome-panel).
- Escritorio: en esta etapa se lanza la aplicación que se encarga de manejar el escritorio (nautilus).
- Aplicaciones: aquí se ejecutan las aplicaciones que quedan sin ejecutarse.
- Ejecución: la sesión es completada y está lista para su uso.
- Consulta para finalizar sesión: es la fase en la que se determina que el cierre de sesión va a llevarse a cabo. Se le informa a todas las aplicaciones registradas el proceso y se pasa a la siguiente fase. Si alguna aplicación no responde no se procede a la salida, y si responde proactivamente entonces se le informa al usuario.
- Fin de la sesión: la sesión entra en esta fase cuando el cierre está a punto de ocurrir e indica a todos los clientes que deben realizar la limpieza final antes de salir.
- Salida: la sesión es terminada y todas las aplicaciones deben terminar con ella.

## 1.5. Gestor de Ventanas.

Un gestor de ventanas es un paquete de software que permite el seguimiento y control de diferentes actividades separándolas físicamente en diferentes partes de la pantalla (ventanas) (11). Es el encargado de realizar la decoración de la ventana y de permitirle al usuario la posibilidad de moverlas, redimensionarlas y cerrarlas. Algunos administradores de ventanas en los entornos de escritorio de

distribuciones GNU/Linux son:

- Kwin: es el administrador de ventanas que por defecto trae KDE. Acepta composición y permite la configuración de su estilo visual, funcionalidades y efectos.
- Metacity: es el administrador de ventanas que trajo el entorno de escritorio Gnome por defecto en sus versiones 2.X. Permite la instalación de temas para su apariencia visual y tiene integración con Compiz y Beryl, aceptando composición en el proceso.
- Mutter: es el administrador de ventanas que trae Gnome 3.

El desarrollo de Mutter empezó a partir del código de Metacity. Utiliza las librerías Clutter, las cuales están basadas en OpenGL<sup>10</sup>, por lo que usándolas Mutter hace posible la aceleración de hardware para un administrador de ventanas. El problema con Mutter es que es vital para el correcto funcionamiento de Gnome Shell, por lo que tiene que ser el administrador de ventanas para Gnome 3 obligatoriamente.

## **1.6. Justificación de la selección del entorno de escritorio base.**

En este estudio se abordaron solamente dos de los entornos de escritorio de GNU/Linux, Gnome (en sus distintas versiones) y KDE. A pesar de la existencia de otros entornos estos son los que mayor cantidad de usuarios acaparan en el mundo por los altos niveles de configuración que brindan, así como la gran cantidad de aplicaciones que tienen para incorporarles funcionalidades y riqueza visual. Para cumplir los objetivos de este trabajo cualquiera de estos dos puede servir de base tecnológica, gracias a la flexibilidad de su arquitectura. Teniendo esto en cuenta, el entorno que servirá como base tecnológica para la realización de este trabajo es Gnome porque, a pesar de que KDE es una buena opción para el desarrollo, resulta imposible olvidar el objetivo principal del sistema operativo, la migración, por lo que se tiene que tener en cuenta que existen usuarios que ya están enmarcados en dicho proceso y que poseen cierto grado de experiencia en el uso de las aplicaciones de Gnome. Un cambio demasiado radical en la interfaz gráfica del sistema operativo dificultaría más la marcha hacia las alternativas libres.

En cuanto a la versión que se debe utilizar la 3.4 porque ya no existe ningún proyecto de Gnome que se enfoque en el desarrollo de las versiones anteriores, por lo que en poco tiempo estas versiones estarán completamente obsoletas teniendo en cuenta el ritmo en el que se actualizan las tecnologías en

---

<sup>10</sup> Open Graphics Library (Librería gráfica de código abierto).

GNU/Linux. Además Gnome 3 fue aclamado por la comunidad de Nova como el entorno de escritorio para la próxima versión del sistema operativo, complaciendo así a una parte del auditorio en el proceso de migración.

## **1.7. Entorno de Desarrollo.**

### **1.7.1. Metodología de Desarrollo.**

Las metodologías de desarrollo se pueden definir como guías para la construcción de un producto. Tienen como objetivo aumentar la calidad del producto y definen un conjunto de etapas por las cuales debe transitar el proceso de desarrollo. Estas definen el qué, cómo y cuándo deben desarrollarse cada una de las actividades. Dada la amplia variedad de las características de los proyectos informáticos es imposible que exista una metodología universal, por lo que están divididas en dos grupos principales: las tradicionales o pesadas y las ágiles.

Entre las metodologías de desarrollo se encuentran:

#### **RUP**

RUP o Proceso Unificado de Rational es una de las metodologías de desarrollo de software más usadas. Se encuentra en el grupo de las tradicionales y es definida como una metodología robusta con gran uso en grandes proyectos de desarrollo.

RUP se caracteriza por:

- Dirigido por casos de uso: *“Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema”* (12).
- Centrado en la arquitectura: *“La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo”* (12).
- Iterativo e incremental: *“El trabajo se divide en partes pequeñas o mini proyectos. Permitiendo que*

*el equilibrio entre Casos de Uso y Arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos flujos de trabajos fundamentales) del cual se obtiene un incremento que produce crecimiento en el producto” (13).*

El desarrollo del producto, según RUP está dividido en cuatro fases:

- Inicio: se define el alcance y los requerimientos del sistema.
- Elaboración: Se define la arquitectura del sistema.
- Construcción: Se construye y prueba el sistema.
- Transición: Se realiza el despliegue y el soporte.

#### **XP.**

XP o Programación Extrema es una metodología ágil que tiene las relaciones interpersonales del equipo como clave para lograr el éxito en el desarrollo de software. Está basada en una relación continua entre el grupo de desarrollo y el cliente, haciendo de este un participante más en el personal que desarrolla el producto. Es usada ampliamente en proyectos con cortos períodos de entrega, con requisitos cambiantes e imprecisos y donde existe un alto riesgo técnico (14).

XP define una serie de Historias de usuario, las cuales son la técnica utilizada para especificar los requisitos del software. Son tarjetas en las cuales el cliente describe las características que el sistema debe cumplir (requisitos funcionales o no funcionales). Estas deben ser lo suficiente breves y sencillas para que el programador las pueda implementar en unas pocas semanas. A efectos de planificación, las historias pueden ser de una a tres semanas de tiempo de programación (para no superar el tamaño de una iteración). Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

#### **SCRUM.**

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Es especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos.

El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración (15).

### **SXP.**

*“SXP es un híbrido cubano de metodologías ágiles que tiene como base las metodologías SCRUM y XP que permiten actualizar los procesos de desarrollo de software para el mejoramiento de su producción.*

*[...]*

*Esta metodología ayuda a fortalecer el trabajo en equipo, enfocados en una misma dirección, permitiendo además seguir de forma clara el avance de las tareas a realizar, a partir de la inserción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la producción, aumentando el nivel de interés del equipo” (16).*

SXP combina la rapidez con que se desarrolla software en la metodología XP y crea un marco para la Gestión de Proyectos gracias a la utilización de SCRUM. Para esto define cuatro fases principales: Planificación – Definición, Desarrollo, Entrega y Mantenimiento.

#### **1.7.2. Justificación de la selección de la metodología**

Para el desarrollo del siguiente trabajo se selecciona la metodología SXP puesto que se adapta completamente a las exigencias de este.

- Número reducido de trabajadores para el desarrollo del trabajo.
- Requisitos cambiantes.
- Alto riesgo técnico.
- Entrega rápida de resultados sin dejar a un lado la calidad.
- Alta flexibilidad.
- Marco para la Gestión de Proyecto.

#### **1.7.3. Lenguajes de Programación**

A continuación se realiza un análisis de los diferentes Lenguajes de Programación a utilizar en el desarrollo del Entorno de Escritorio de Nova 4.0:



### **JavaScript.**

JavaScript es un lenguaje de programación interpretado, basado en el estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. JavaScript fue desarrollado originalmente por Brendan Eich de Netscape con el nombre de Mocha, el cuál fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript. El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó soporte para la tecnología Java en su navegador web Netscape Navigator en la versión 2.003 en diciembre de 1995.

JavaScript soporta gran parte de la sintaxis de programación estructurada a partir de C (por ejemplo, las declaraciones if, los bucles while, las declaraciones switch, etc). Al igual que C, JavaScript hace una distinción entre expresiones y declaraciones. Una diferencia sintáctica de C es la inserción automática de punto y coma (17).

Al igual que la mayoría de los lenguajes de scripts los tipos de datos se asocian a los valores y no a las variables, haciendo posible que en su ámbito una variable pueda tener diferentes tipos de datos. Otra similitud es que en JavaScript no existe diferencia entre las funciones y los métodos. JavaScript es completamente orientado a objetos, ya que hasta los tipos de datos primitivos (números, cadenas, etc.) son objetos. JavaScript utiliza prototipos en lugar de clases para simular la herencia, por lo que es posible tener muchas subclases con dichos prototipos.

A pesar de su amplio uso en las aplicaciones o páginas web, JavaScript es usado ampliamente en otros softwares que no están vinculados la web directamente. Por ejemplo es usado para la construcción de las interfaces gráficas de Mozilla (Firefox y Thunderbird), así como en Google Chrome y en Safari 5 para la construcción de sus extensiones, entre muchos otros.

### **ANSI C.**

ANSI C, o simplemente C, es un lenguaje de programación creado en 1972 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B, a su vez basado en BCPL. Al igual que B, es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix.

C es un lenguaje de programación de nivel medio ya que combina los elementos del lenguaje de alto nivel

con la funcionalidad del ensamblador, pero es muy versátil y eficiente y revolucionó las técnicas y estilo de programación. Su característica principal es ser portable, es decir, es posible adaptar los programas escritos para un tipo de computadora en otra. Otra de sus características principales es el ser estructurado, es decir, el programa se divide en módulos (funciones) independientes entre sí (18).

Sin embargo, la popularidad, eficacia y potencia de C se ha producido porque este lenguaje no está prácticamente asociado a ningún sistema operativo, ni a ninguna máquina en especial. Esta es la razón fundamental por la que C es conocido como el lenguaje de programación de sistemas por excelencia. Es un lenguaje de programación ordenado, potente y muy fácil de adaptar a diferentes arquitecturas.

#### **1.7.4. Librerías a utilizar**

##### **Gobject Introspection**

Las principales librerías de las distribuciones de GNU/Linux están escritas en el lenguaje de programación C, puesto que es uno de los lenguajes ideales para el desarrollo de bibliotecas gráficas, de multimedia y para el trabajo con sistemas de bajo nivel, no así para sistemas complejos ya que se hace difícil y propenso a errores en el desarrollo. Esto no sucede en otros lenguajes como Python, Java, Vala, JavaScript, etc. donde los sistemas de recolección de basura facilitan la lógica de la aplicación, por lo que tiene sentido que las aplicaciones estén construidas en varios lenguajes de programación. Para que esto fuera posible se hacían bindings de las bibliotecas para varios lenguajes y de esta forma poder ser utilizadas desde varios lenguajes de programación. Este proceso era engorroso y con varias lagunas ya que siempre existía algo que no se podía convertir de lenguaje (a través del binding no se podía conocer qué tipo de datos tenían los elementos que estaban dentro de un GList por ejemplo), además con el alto número de bibliotecas usadas en el mundo de GNU/Linux es un trabajo agotador y extenso hacerle un binding a cada biblioteca a utilizar.

Para esto surge la librería Gobject Introspection, la cual está en desarrollo y fue lanzada estable por primera vez junto con GTK 3. Esta sirve de puente para comunicar a los distintos lenguajes de programación usados en GNU/Linux, por lo que mediante esta es posible que una biblioteca hecha en C pueda ser utilizada sin mayor esfuerzo por otros lenguajes y por otras aplicaciones.

Gobject Introspection pone los metadatos de la implementación dentro de la misma librería Gobject,

usando anotaciones dentro de los comentarios que son interpretados luego. GObject Introspection también crea una base de datos llamada GObject Introspection Repository (gir) en la que se guardan un fichero con formato XML y con extensión gir por cada biblioteca que puede ser usada a través de GObject Introspection con la información sobre las funciones, procedimientos y variables que se encuentran en dicha librería. Al ser compiladas estas librerías un escáner revisa el fuente en busca de las anotaciones en los comentarios y crea un el fichero gir que es compilado y situado en el repositorio (/usr/share/gir-1.0), donde podrá ser usado por otras librerías (19).

Para que GObject Introspection funcione correctamente en un lenguaje determinado, deben existir los bindings para GObject Introspection para ese lenguaje. Esto evidencia una clara ventaja sobre el antiguo sistema de bindings pues ahora estos solo se hacen para algunos lenguajes mientras que antes se tenían que hacer para un gran conjunto de librerías. Actualmente GObject Introspection tiene bindings para varios lenguajes de programación como Vala, Python, Perl, Javascript, Java, Qt, Ruby, Lua y muchos otros. Además es usado en muchas aplicaciones muy usadas en el entorno de escritorio Gnome como Folks, Gnome Shell y telepathy-glib.

### **Clutter**

Clutter es una biblioteca gráfica basada en GObject que permite la creación de aplicaciones con interfaces gráficas de usuario de manera rápida y con una gran riqueza visual. Esta biblioteca trabaja mediante la manipulación de una escena gráfica de superficies 2D (llamadas actores) dentro de un espacio de tres dimensiones (20).

En esta librería la clase base de todos los tipos de superficies es ClutterActor y con estos se pueden realizar un gran número de acciones como posicionar, escalar y girar en el espacio 3D. Además tiene un conjunto de propiedades que facilitan el trabajo con este como el recorte en 2D, los componentes hijos y la opacidad con la que se muestra, siempre teniendo en cuenta que cualquier cambio realizado por el padre se le aplican a todos los hijos. Entre las subclases más importantes de ClutterActor se encuentra ClutterStage, el cual es una ventana de nivel superior en la que los actores hijos se colocan y manipulan y es creada automáticamente cuando se inicia Clutter.

Clutter también proporciona una subclase de ClutterActor para proporcionar los servicios básicos de

animación llamada ClutterTimeLines. Los actores también pueden ser animados explícitamente mediante las diferentes implementaciones existentes de ClutterBehaviour o implícitamente a través de la función clutter\_actor\_animate ().

### **Conclusiones parciales.**

El análisis de las metáforas, conceptos y estado del arte en cuanto a los entornos de escritorio permitió una visión más amplia de las funcionalidades que se desean para el entorno de escritorio de Nova 4.0. En la selección del entorno de escritorio se obtuvo que Gnome 3 es la mejor base tecnológica para el desarrollo de dicho entorno de escritorio. Esta selección tiene la ventaja de que no es necesario la implementación desde cero y que a la vez se puedan implementar nuevas funcionalidades. Dada la selección de la base tecnológica se requiere que el desarrollo se realice en los lenguajes de programación C y JavaScript, utilizando CSS como lenguaje para proporcionar los estilos. Para guiar el proceso de ingeniería se seleccionó la metodología SXP, mostrándose a continuación los principales artefactos recogidos en las iteraciones de la metodología.

## ANÁLISIS Y DISEÑO DEL ENTORNO DE ESCRITORIO DE NOVA 4.0

En este capítulo se inicia el desarrollo de la propuesta de solución haciendo uso de la metodología seleccionada, para ello se hace hincapié en sus primeras fases, recogiendo los principales artefactos generados en estas.

### 2.1. Fase Inicial del Proceso de Ingeniería de Software

Las metodologías de software, como se explicada en el capítulo anterior, definen un conjunto de procedimientos que se deben llevar a cabo para el correcto desarrollo de software. SXP, metodología acordada para el desarrollo del entorno de escritorio de Nova 4.0, en su primera fase (Planificación – Definición) define la realización de un conjunto de artefactos relacionados con la concepción inicial del sistema a desarrollar. A continuación se recogen los principales artefactos que se concibieron durante la primera etapa de la metodología.

#### 2.1.1. Concepción del sistema

Concluida la entrevista con el cliente se genera la plantilla concepción del sistema, en la cual se recogen los datos fundamentales que dan inicio al desarrollo de la propuesta de solución, incluyendo la descripción de esta.

#### Descripción de la solución propuesta: Nova Escritorio 4.0.

Para darle solución a la situación problemática planteada se propone desarrollar un entorno de escritorio orientado al usuario final, el cual contribuirá a llevar la migración a Estándares Abiertos en Cuba con grandes novedades en la experiencia de usuario y con un diseño innovador y moderno.

El desarrollo de Nova Escritorio se centrará esencialmente en la modificación del entorno de escritorio seleccionado: Gnome en su versión 3.4 para que cumpla con los requerimientos visuales de la distribución de GNU/Linux Nova. Para esto el entorno tendrá tres sesiones que servirán para darles un tratamiento diferenciado a sus usuarios:

- Sesión Oficial de Nova: esta es la sesión principal del entorno de escritorio. Esta sesión tiene como novedad el uso de Gnome Shell como software que cumpla los requerimientos del panel y tiene como gestor de ventanas a Mutter.

- Sesión para la Migración: esta es la sesión dedicada a suavizar el cambio de los usuarios que migran al sistema operativo Nova desde Microsoft Windows, por lo que su diseño se parece mucho al de este último. Esta tiene como software para simular el panel a Nova Shell, el cual es una rama a desarrollar a partir de Gnome Shell y que se debe parecer en algunas características al panel de Windows 7.
- Sesión de poco consumo de recursos y a prueba de fallos: esta sesión está dedicada a las computadoras que por alguna razón no puedan explotar las facilidades y efectos que se brindan en las sesiones anteriores. Esta sesión tiene como panel el software Gnome Panel en su versión 3.4 y como gestor de ventanas la versión 2.34 de Metacity.

En todas las sesiones se utiliza Nautilus como software para el manejador de archivos y para simular el escritorio.

Para la unión de cada uno de los componentes, se hace necesario un software que maneje las sesiones e integre el funcionamiento de cada uno de los componentes del entorno de escritorio.

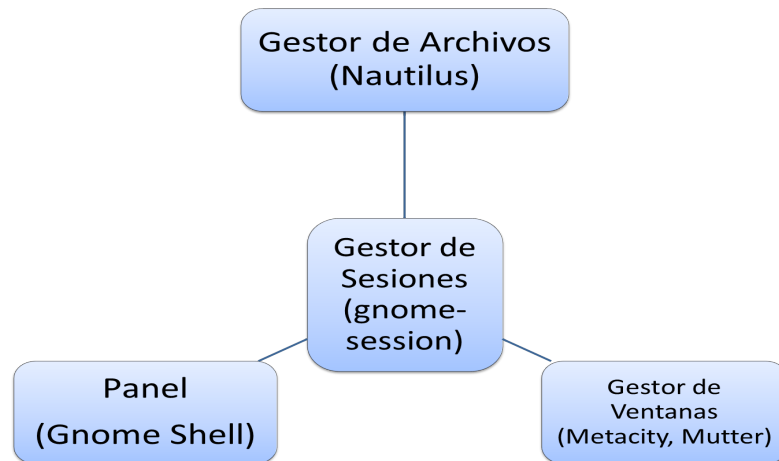
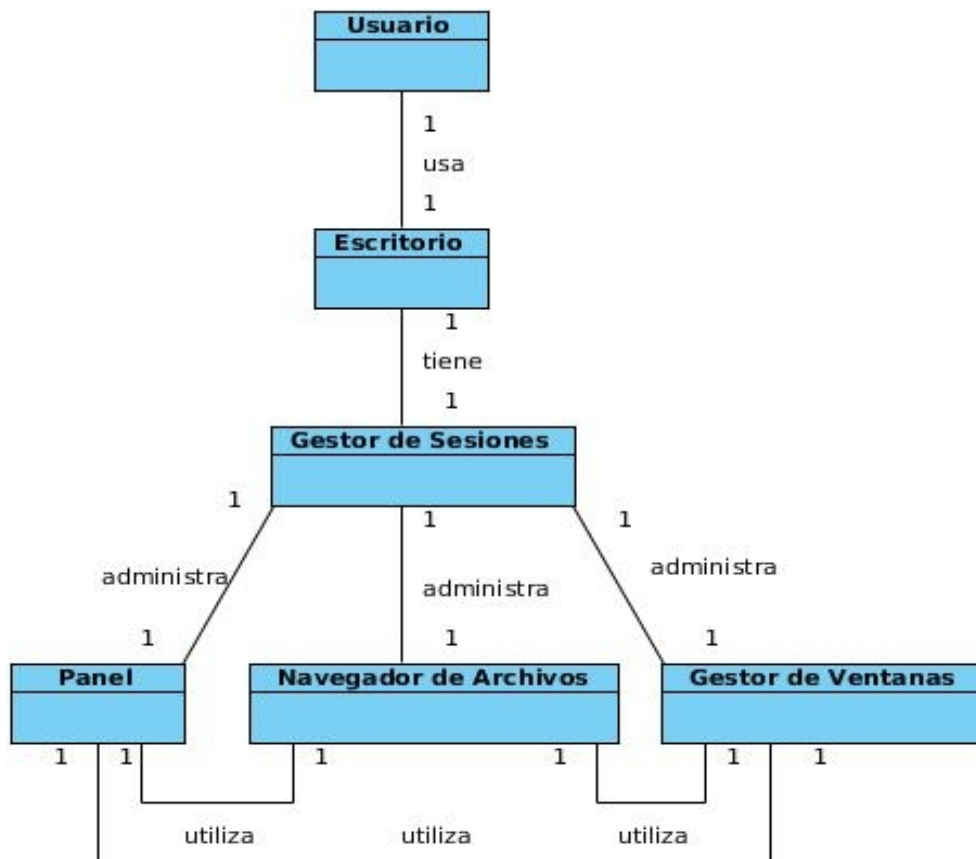


Ilustración 2: Estructura del entorno de escritorio de Nova 4.0.

### 2.1.2. Modelo de Dominio.

El modelo de dominio es una representación gráfica de objetos reales relacionados con el problema y las relaciones entre ellos. Este consiste en un conjunto de diagramas de clases sin la definición de sus operaciones. El modelo de dominio ayuda a comprender los conceptos claves y el vocabulario que será utilizado por el equipo de desarrollo al referirse al sistema, además puede servir para que los clientes tengan una mayor comprensión del problema.

El modelo de dominio del presente trabajo se muestra a continuación.



*Ilustración 3: Modelo de dominio del entorno de escritorio de Nova 4.0.*

### 2.1.3. Lista de Reserva del Producto (LRP)

La Lista de Reserva de Producto o LRP es un artefacto generado en captura de requisitos y constituye una lista priorizada con los elementos que se van a desarrollar en el proyecto. A continuación se muestra la Lista de Reserva del Producto del entorno de escritorio a desarrollar.

Asignado a	Item	Descripción	Estimación	Estimado por
Prioridad Muy Alta				
Juan Manuel Fuentes Rodríguez	1	Permitirle al usuario autenticarse para entrar a la sesión.	1	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	2	Permitirle al usuario escoger la sesión por la cual desea entrar.	1/2	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	3	Permitir integrar los componentes del Entorno de Escritorio.	2	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	4	Permitirle al usuario interactuar con las opciones de apagado y cierre de sesión.	2	Juan Manuel Fuentes Rodríguez
Prioridad Alta				
Juan Manuel Fuentes Rodríguez	5	Permitirle al usuario ejecutar y visualizar las aplicaciones instaladas.	1	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	6	Permitirle al usuario navegar por el sistema de archivos.	1	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	7	Permitirle al usuario navegar entre las ventanas abiertas.	1	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	8	Permitirle al usuario realizar las diferentes acciones sobre las aplicaciones abiertas (maximizar, minimizar, cerrar).	1	Juan Manuel Fuentes Rodríguez



*ANÁLISIS Y DISEÑO DEL ENTORNO DE ESCRITORIO DE NOVA 4.0*

Juan Manuel Fuentes Rodríguez	9	Permitirle al usuario ver e interactuar con las notificaciones del entorno de escritorio y el sistema operativo.	1	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	10	Permitirle al usuario ver e interactuar con los indicadores del Entorno de Escritorio.	1	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	11	Permitirle al usuario ejecutar un comando por medio de una interfaz gráfica.	1	Juan Manuel Fuentes Rodríguez
Prioridad Media				
Juan Manuel Fuentes Rodríguez	12	Permitirle al usuario acceder y modificar las configuraciones del Entorno de Escritorio.	1/2	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	13	Permitirle al usuario visualizar, ejecutar y eliminar las aplicaciones favoritas.	1/2	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	14	Permitirle al usuario ver y modificar el reloj y el calendario.	1/2	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	15	Permitirle al usuario montar particiones extraíbles (memorias, particiones de disco)	1/2	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	16	Permitirle al usuario montar directorios utilizando los diferentes protocolos de red (SSH, SMB, FTP).	1/2	Juan Manuel Fuentes Rodríguez
Juan Manuel	17	Permitirle al usuario compartir una	1/2	Juan Manuel Fuentes

ANÁLISIS Y DISEÑO DEL ENTORNO DE ESCRITORIO DE NOVA 4.0

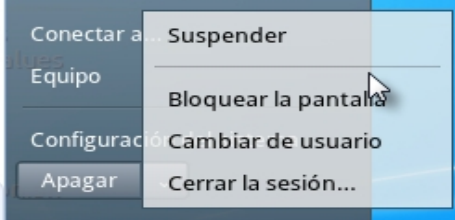
Fuentes Rodríguez		carpeta.		Rodríguez
Juan Manuel Fuentes Rodríguez	18	Permitirle al usuario acceder a la carpeta personal y a los marcadores en el sistema de archivos definidos.	1/2	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	19	Permitirle al usuario realizar las diferentes opciones con los archivos y directorios (copiar, mover, pegar, eliminar, comprimir, descomprimir).	1/2	Juan Manuel Fuentes Rodríguez
Prioridad baja				
Juan Manuel Fuentes Rodríguez	20	Permitirle al usuario buscar las aplicaciones instaladas mediante un cajón de búsqueda.	1/2	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	21	Permitirle al usuario acceder al calendario del sistema.	1/2	Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	22	Permitirle al usuario ver las aplicaciones más utilizadas.	1/2	Juan Manuel Fuentes Rodríguez
Requisitos No Funcionales				
Juan Manuel Fuentes Rodríguez	1	El sistema debe tener al menos tres sesiones: <ul style="list-style-type: none"> <li>• Nova Oficial.</li> <li>• Nova Migración.</li> <li>• A Prueba de Fallos.</li> </ul>		Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	2	El sistema debe ser: <ul style="list-style-type: none"> <li>• Configurable.</li> <li>• Sencillo y simple.</li> </ul>		Juan Manuel Fuentes Rodríguez

		<ul style="list-style-type: none"> <li>• Intuitivo.</li> </ul>		
Juan Manuel Fuentes Rodríguez	3	El sistema debe funcionar en computadoras con buenas prestaciones de hardware (como mínimo 1 Gb de Ram y 3 GHZ de velocidad de procesamiento)		Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	4	El sistema debe ser extensible por lo que, con el tiempo se le puedan adicionar nuevas funcionalidades.		Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	5	La sesión Nova Migración debe tener un gran parecido al entorno de escritorio de Microsoft Windows 7.		Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	6	La sesión a prueba de fallos no puede tener efectos para que pueda ser accedida por usuarios con computadoras que no puedan explotar las facilidades de las otras sesiones. Esta debe funcionar bajo las siguientes condiciones de hardware: <ul style="list-style-type: none"> <li>➤ Ram: mayor que 512 Mb.</li> <li>➤ Velocidad de procesamiento: 2.5 GHZ.</li> </ul>		Juan Manuel Fuentes Rodríguez
Juan Manuel Fuentes Rodríguez	7	La sesión de Nova Oficial debe cumplir con los requerimientos visuales del sistema operativo		Juan Manuel Fuentes Rodríguez

		Nova.		
Juan Manuel Fuentes Rodríguez	8	<p>El sistema debe brindar un conjunto de herramientas que faciliten el trabajo con el entorno de escritorio:</p> <ul style="list-style-type: none"> <li>• Suite Ofimática.</li> <li>• Navegador Web.</li> <li>• Herramienta de Mensajería Instantánea.</li> <li>• Reproductor de audio.</li> <li>• Reproductor de Video.</li> <li>• Herramienta de quemado de discos.</li> <li>• Accesorios (Calculadora, Terminal, Editor de Textos, etc.).</li> <li>• Juegos Sencillos.</li> <li>• Visor de Imágenes.</li> <li>• Cliente de correo y calendario.</li> </ul>		Juan Manuel Fuentes Rodríguez

#### 2.1.4. Historia de Usuarios y prototipos de interfaces de usuario.

La Historia de Usuario es la forma que tiene la metodología SXP para especificar los requisitos. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema tiene que tener. Las historias deben ser lo suficientemente comprensibles y delimitadas como para que el programador las implemente en unas pocas semanas.

Historia de usuario.	
Número: HU-1	Nombre Historia de Usuario: Gestionar sesión.
Modificación de Historia de Usuario Número:	
Usuario: Juan Manuel Fuentes Rodríguez	Iteración Asignada: 1.
Prioridad en Negocio: Alta.	Puntos Estimados:
Riesgo en Desarrollo: Bajo.	Puntos Reales:
<p><b>Descripción:</b> Es el desarrollo de Gnome Session. Este software es el encargado de arrancar los programas iniciales una vez que el usuario se autentique correctamente. También es el encargado de la salida de la sesión (ya sea porque se apague, reinicie, suspenda o hiberne la computadora, o porque se cierre la sesión) y de cerrar los programas iniciales y los que están en ejecución de forma correcta para evitar la pérdida de datos innecesaria.</p> <p><b>Observación:</b> Gnome Session es un programa que se ejecuta como demonio, por lo que va a estar ejecutándose por detrás de todas las aplicaciones desde que se inicia la sesión hasta que se termine esta. Durante el proceso de Cambio de Usuario este proceso se mantiene ejecutándose para poder guardar los datos de las aplicaciones abiertas en la sesión anterior.</p>	
<p><b>Prototipo de Interfaz de usuario.</b></p> 	

Historia de usuario.	
Número: HU-2	Nombre Historia de Usuario: Mostrar la lista de Indicadores.
Modificación de Historia de Usuario Número:	
Usuario: Juan Manuel Fuentes Rodríguez	Iteración Asignada: 1.

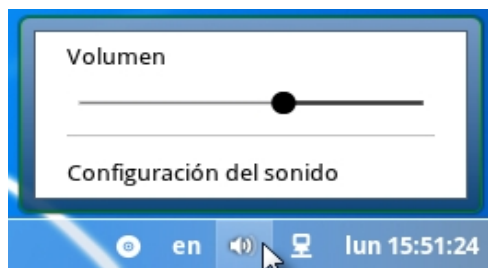
Rodríguez	
<b>Prioridad en Negocio:</b> Alta.	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Bajo.	<b>Puntos Reales:</b>

**Descripción:** Los indicadores son un conjunto de miniaplicaciones que muestran el estado de algún proceso en específico, permitiendo editar algunas opciones para la configuración de estos procesos. Estos están colocados en un extremo del panel. Algunos indicadores a utilizar son:

- Accesibilidad: para controlar las opciones de accesibilidad para aquellas personas con discapacidades.
- Bluetooth: para mostrar y configurar el estado de las conexiones por bluetooth de la computadora.
- Teclado: para mostrar y cambiar la distribución de teclado.
- Red: para mostrar y configurar las conexiones de red (inalámbricas y cableadas).
- Batería: para mostrar el porcentaje de utilización de la batería (solo para computadoras con una batería incrustada).
- Volumen: para mostrar y configurar el volumen y los dispositivos de audio.

**Observación:**

**Prototipo de Interfaz de usuario.**



### Historia de usuario.

<b>Número:</b> HU-3	<b>Nombre Historia de Usuario:</b> Mostrar la lista de notificaciones.
<b>Modificación de Historia de Usuario Número:</b>	
<b>Usuario:</b> Juan Manuel Fuentes	<b>Iteración Asignada:</b> 1.

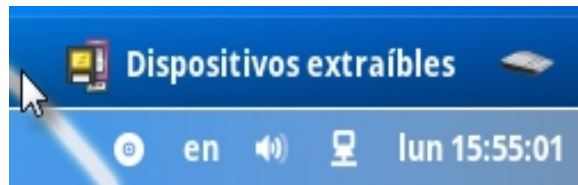
Rodríguez	
<b>Prioridad en Negocio:</b> Media.	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Bajo.	<b>Puntos Reales:</b>

**Descripción:** Las notificaciones son acciones realizadas por el entorno de escritorio hechas para informarle al usuario un suceso de interés para él, por ejemplo:

- Notificaciones de mensajería instantánea.
- Notificaciones de correo.
- Notificaciones de descarga.
- Notificaciones de actualizaciones.

**Observación:** El sistema de notificaciones es un programa que se ejecuta como demonio, por lo que va a estar ejecutándose por detrás de todas las aplicaciones desde que se inicia la sesión hasta que se termine esta, esperando que algunas de estas aplicaciones le indique que tiene que lanzar alguna notificación.

**Prototipo de Interfaz de usuario.**



Historia de usuario.	
<b>Número:</b> HU-4	<b>Nombre Historia de Usuario:</b> Gestionar hora, fecha y calendario.
<b>Modificación de Historia de Usuario Número:</b>	
<b>Usuario:</b> Juan Manuel Fuentes Rodríguez	<b>Iteración Asignada:</b> 1.
<b>Prioridad en Negocio:</b> Media.	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Bajo.	<b>Puntos Reales:</b>
<b>Descripción:</b> El entorno de escritorio actualiza la hora y la fecha mediante el uso de la librería Glib a	

través de Gobject-Introspection. El calendario se visualiza a través de la conexión con algún software para la gestión de citas y recordatorios (por ejemplo el Calendario de Evolution).

**Observaciones.**

**Prototipo de Interfaz de usuario.**



**Historia de usuario.**

**Número:** HU-5 **Nombre Historia de Usuario:** Gestionar navegación en los archivos.

**Modificación de Historia de Usuario Número:**

**Usuario:** Juan Manuel Fuentes **Iteración Asignada:** 1.  
Rodríguez

**Prioridad en Negocio:** Alta. **Puntos Estimados:**

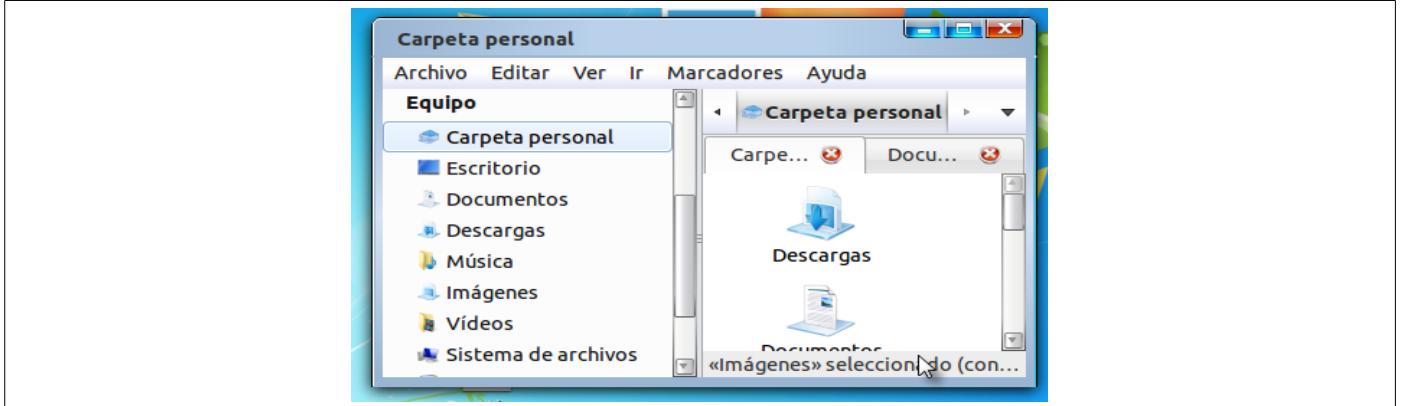
**Riesgo en Desarrollo:** Bajo. **Puntos Reales:**

**Descripción:** El navegador de archivos Nautilus es el encargado de gestionar la navegación en los archivos, así como del control de las acciones que se realizan con estos (copiar, mover, eliminar, crear). También es el encargado de la gestión del escritorio.

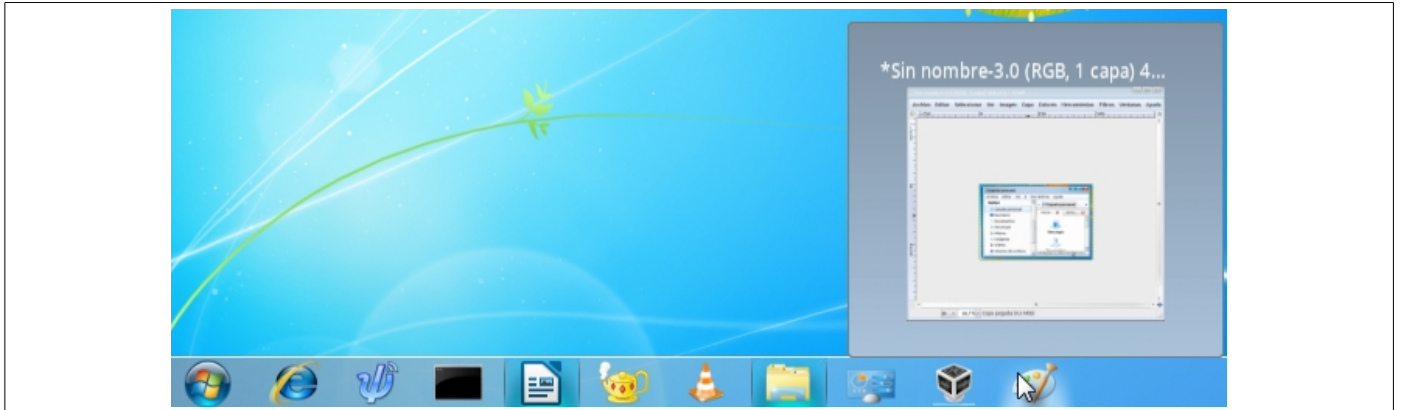
**Observación:** Nautilus es el encargado de la gestión de los dispositivos extraíbles, directorios montados por la red y de los marcadores.

**Prototipo de Interfaz de Usuario.**

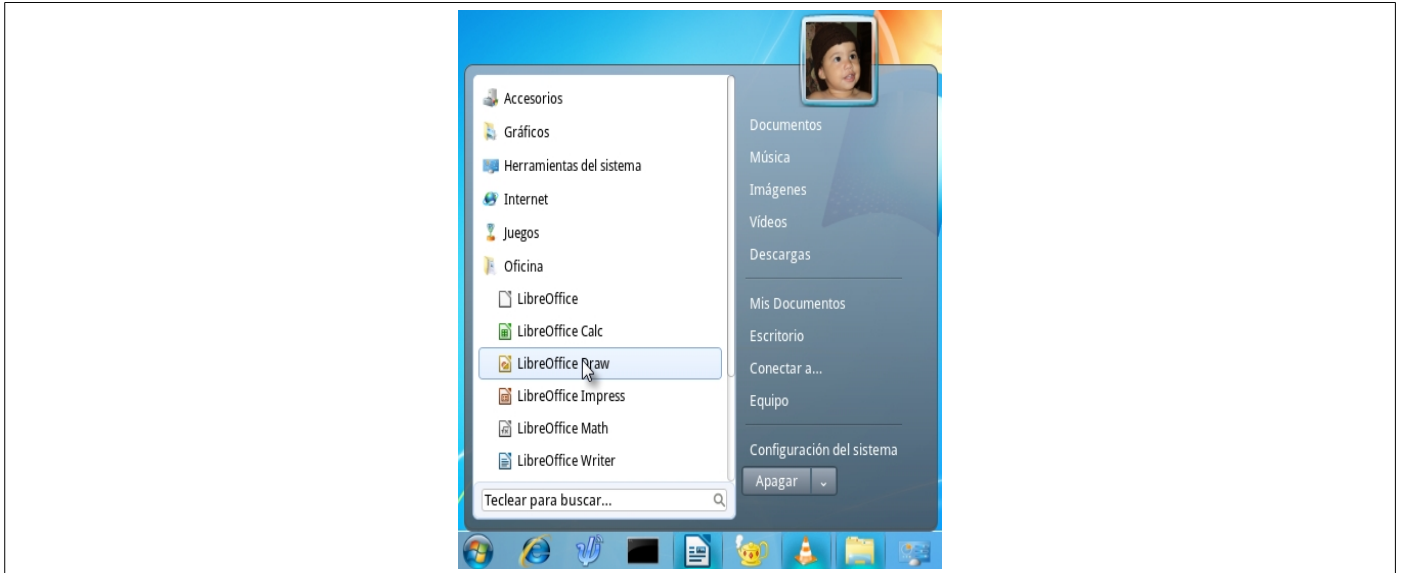




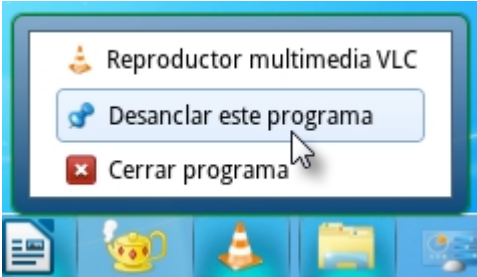
Historia de usuario.	
<b>Número:</b> HU-6	<b>Nombre Historia de Usuario:</b> Gestionar la lista de ventanas abiertas.
<b>Modificación de Historia de Usuario Número:</b>	
<b>Usuario:</b> Juan Manuel Fuentes Rodríguez	<b>Iteración Asignada:</b> 1.
<b>Prioridad en Negocio:</b> Alta.	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Bajo.	<b>Puntos Reales:</b>
<b>Descripción:</b> Desarrollo de una aplicación, que utilizando las funciones y operaciones dadas por el gestor de ventanas de la sesión, administre las acciones que se puedan realizar sobre las ventanas abiertas.	
<b>Observaciones.</b>	
<b>Prototipo de Interfaz de Usuario.</b>	



Historia de usuario.	
<b>Número:</b> HU-7	<b>Nombre Historia de Usuario:</b> Gestionar la lista de aplicaciones instaladas.
<b>Modificación de Historia de Usuario Número:</b>	
<b>Usuario:</b> Juan Manuel Fuentes Rodríguez	<b>Iteración Asignada:</b> 1.
<b>Prioridad en Negocio:</b> Media.	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Bajo.	<b>Puntos Reales:</b>
<b>Descripción:</b> Desarrollo de un menú que facilite la interacción del usuario con la lista de programas instalados, permitiendo la ejecución de cada uno de estas.	
<b>Observación:</b> Para la edición del menú se hace uso de la herramienta Alacarte.	
<b>Prototipo de Interfaz de Usuario.</b>	



Historia de usuario.	
<b>Número:</b> HU-8	<b>Nombre Historia de Usuario:</b> Gestionar la configuración de los elementos del entorno de escritorio.
<b>Modificación de Historia de Usuario Número:</b>	
<b>Usuario:</b> Juan Manuel Fuentes Rodríguez	<b>Iteración Asignada:</b> 1.
<b>Prioridad en Negocio:</b> Baja.	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Bajo.	<b>Puntos Reales:</b>
<b>Descripción:</b> Los componentes del entorno de escritorio tienen un programa que es el encargado de guardar la configuración llamado Gsettings. La utilización de este programa va a posibilitar una forma de almacenamiento de las preferencias.	
<b>Observación:</b> Gsettings es un demonio por lo que está corriendo en todo el ciclo de vida de la sesión por detrás de las aplicaciones.	
<b>Prototipo de Interfaz de Usuario.</b>	

Historia de usuario.	
Número: HU-9	Nombre Historia de Usuario: Gestionar las aplicaciones favoritas.
Modificación de Historia de Usuario Número:	
Usuario: Juan Manuel Fuentes Rodríguez	Iteración Asignada: 1.
Prioridad en Negocio: Media.	Puntos Estimados:
Riesgo en Desarrollo: Bajo.	Puntos Reales:
Descripción: Desarrollar un programa que permita la gestión de las aplicaciones favoritas.	
Observación:	
Prototipo de Interfaz de Usuario.	
	

### 2.1.5. Lista de riesgos

Todo desarrollo de software está expuesto a distintos riesgos, por lo que es necesario evitarlos realizando varias estrategias de mitigación y planes de contingencia en caso de que se materialice el riesgo. Los riesgos en el desarrollo del entorno de escritorio de Nova 4.0 se muestran a continuación.

Riesgo	Tipo de Riesgo	Impacto	Descripción	Probabilidad	Efectos	Mitigación del riesgo
	Tecnológico	Retraso en el trabajo	Rotura en la computadora en la que se trabaja.	Medio	Altos	Revisión diaria de la computadora para ver si existe algún defecto

	Tecnológico	Retraso en el trabajo.	Falta de fluido eléctrico en el laboratorio de producción	Alto	Altos	técnico Aprovechar los horarios laborables para que no ocurra ningún retraso.
	Personal	Retraso en las actividades que se desarrollan.	Ausencia de un integrante del proyecto por enfermedad.	Bajo	Medio	Aprovechar los horarios laborables para que no ocurra ningún retraso.
	Personal	Retraso en el trabajo.	Ausencia de algún miembro del proyecto.	Bajo	Bajo	Aprovechar los horarios laborables para que no ocurra ningún retraso.

### 2.1.6. Metáfora del sistema

Según la metodología SXP, metodología escogida para el desarrollo, la evolución del producto debe estar guiada por una o varias metáforas que dictan como trabaja o debería trabajar el sistema.

La metáfora de la solución propuesta es:

*“Antes de autenticarse en el sistema el usuario debe escoger una sesión con la que interactuar y basado en su elección se debe mostrar un entorno de escritorio con una configuración específica”.*

## 2.2. Segunda Fase del Proceso de Ingeniería de Software

En esta segunda fase se describen un conjunto de actividades que se deben de llevar a cabo en el desarrollo para la correcta implementación de las historias de usuario descritas en la primera fase del proceso de ingeniería. Además se muestra el diagrama de componentes, vital para el correcto entendimiento del desarrollo.

### 2.2.1. Tareas de Ingeniería

A continuación se relacionan algunas de las tareas de ingenierías correspondientes a las historias de usuario, basándose en la prioridad que tienen. Debido a que el proceso es cambiante estas deben ir adecuándose a las nuevas funcionalidades propuestas, por lo que esta es solo una planificación inicial.

Tarea de Ingeniería	
Número de la tarea: 1	Nombre de la historia de usuario: HU-1
Nombre Tarea: Programar la sesión de Nova Migración.	
Tipo de Tarea: Desarrollo.	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Programador Responsable: Juan Manuel Fuentes Rodríguez	
Descripción: Sesión que mostrará el entorno de escritorio para la migración.	

Tarea de Ingeniería	
Número de la tarea: 2	Nombre de la historia de usuario: HU-1
Nombre Tarea: Programar el cierre de sesión de Nova Migración.	
Tipo de Tarea: Desarrollo.	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Programador Responsable: Juan Manuel Fuentes Rodríguez	
Descripción: Funcionalidad que permite cerrar la sesión de Nova Migración.	

Tarea de Ingeniería	
Número de la tarea: 3	Nombre de la historia de usuario: HU-2
Nombre Tarea: Estudiar el uso de los indicadores.	
Tipo de Tarea: Investigación.	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Programador Responsable: Juan Manuel Fuentes Rodríguez	
Descripción: Investigar cómo son manejados los indicadores en Gnome Shell.	

Tarea de Ingeniería	
Número de la tarea: 4	Nombre de la historia de usuario: HU-2
Nombre Tarea: Desarrollar el menú de los indicadores en la sesión de Nova Migración.	

ANÁLISIS Y DISEÑO DEL ENTORNO DE ESCRITORIO DE NOVA 4.0

Tipo de Tarea: Desarrollo.	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Programador Responsable: Juan Manuel Fuentes Rodríguez	
Descripción: Funcionalidad que muestra los indicadores al estilo de Windows 7.	

Tarea de Ingeniería	
Número de la tarea: 5	Nombre de la historia de usuario: HU-3
Nombre Tarea: Programar los notificaciones en la sesión de Nova Migración.	
Tipo de Tarea: Desarrollo.	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Programador Responsable: Juan Manuel Fuentes Rodríguez	
Descripción: Funcionalidad que permite mostrar las notificaciones al estilo de Windows 7.	

Tarea de Ingeniería	
Número de la tarea: 6	Nombre de la historia de usuario: HU-4
Nombre Tarea: Programar el botón de la hora y el calendario para la sesión de Nova Escritorio.	
Tipo de Tarea: Desarrollo.	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Programador Responsable: Juan Manuel Fuentes Rodríguez	
Descripción: Funcionalidad que muestra el la hora y el calendario al estilo de Windows 7.	

Tarea de Ingeniería	
Número de la tarea: 7	Nombre de la historia de usuario: HU-6
Nombre Tarea: Programar la lista de ventanas de la sesión de Nova Migración.	
Tipo de Tarea: Desarrollo.	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Programador Responsable: Juan Manuel Fuentes Rodríguez	
Descripción: Funcionalidad que muestra la lista de ventanas abiertas y los favoritos al estilo de Windows 7.	

### 2.2.2. Diagrama de Componentes.

El diagrama de componentes muestra las organizaciones de las dependencias lógicas entre los diferentes componentes de software del producto, ya sea binarios o código fuente o ejecutables.

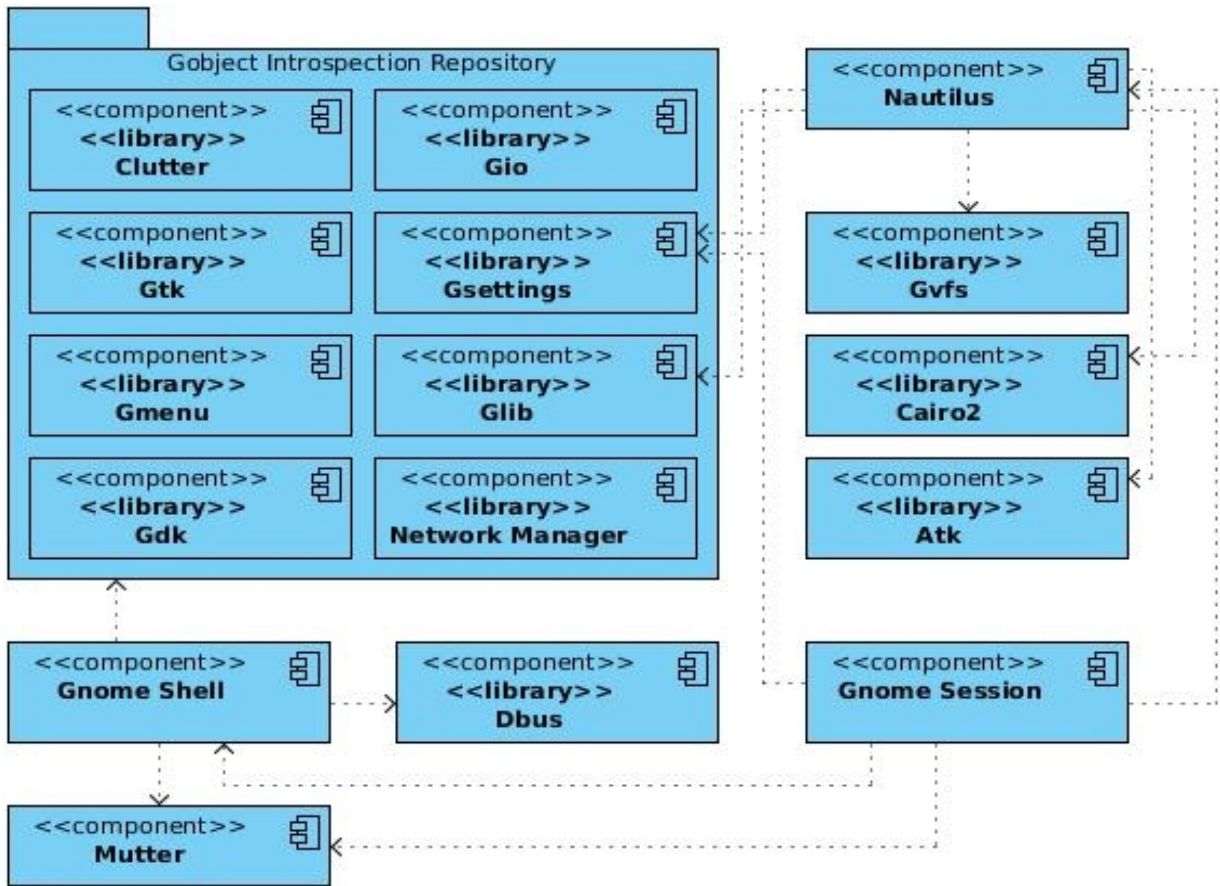


Ilustración 4: Diagrama de Componentes del entorno de escritorio de Nova 4.0.

### 2.2.3. Plan de entregas

Release	Descripción de la iteración	Orden de la Historia de Usuario a Implementar	Duración Total
1	Para este lanzamiento el usuario contará	HU-1	15/10/2011-1/2/2012



	con un entorno en el cual puede acceder mediante la vista de acceso al sistema. Este gestionará la navegación por la estructura de archivos y mostrará una lista de ventanas.	HU-5 HU-6	
2	Para esta iteración el panel contará con un Menú de Inicio, la lista de ventanas tendrá gestión de favoritos y se mostrarán las listas de notificaciones e indicadores.	HU-7 HU-9 HU-2 HU-3	2/22/012-1/5/2012
3	Para el lanzamiento final el panel tendrá un lugar desde donde se gestione la hora y la fecha, además el entorno en general será configurable.	HU-4 HU-8	2/5/2012-1/7/2012

### Conclusiones parciales

Una vez descrita la solución propuesta, se puede concluir que esta consiste en un paquete de componentes que suministra los elementos básicos para el trabajo con el entorno de escritorio. Entre ellos se encuentran las tres sesiones combinando así las potencias que brinda el software libre con un diseño centrado en migrar los usuarios inexpertos.

En este capítulo se definieron además los requisitos funcionales y no funcionales, el modelo de dominio y se redactaron las historias de usuario, como resultado de la fase inicial de la metodología SXP. Para la segunda fase se realizó la planificación del proyecto, se establecieron para cada historia de usuario las principales tareas asociadas para la primera planificación y se priorizaron. La planificación de iteraciones se hizo teniendo en cuenta la prioridad para el negocio de cada historia de usuario. Finalmente se realizaron los diagramas de componentes el cual permitió dar una mejor visión del proyecto.

## VALIDACIÓN DEL ENTORNO DE ESCRITORIO DE NOVA 4.0

Una etapa de suma importancia en el desarrollo de software es el periodo de las pruebas. En la metodología usada en esta investigación es necesario una etapa de pruebas en cada una de las iteraciones en el desarrollo del software, con el objetivo de comprobar que el sistema cumple correctamente con cada una de las funcionalidades previstas, permitiendo que el usuario del producto determine su grado de aceptación con el desarrollo.

En el presente capítulo se exponen algunos casos de pruebas a los que fue sometido el entorno de escritorio de Nova para su versión 4.0 en cada una de las iteraciones para comprobar el funcionamiento de cada una de las historias de usuario propuestas.

### 3.1. Casos de Prueba de Aceptación

El objetivo de los casos de prueba de aceptación en la metodología SXP es validar que el sistema cumple con el funcionamiento esperado. Estos son comprobados por el usuario de forma tal que sea este el que determine el grado de aceptación que tiene con respecto a las funcionalidades y el rendimiento del producto. Para la validación del entorno de escritorio de la distribución de GNU/Linux Nova en la versión 4.0 se definieron los casos de prueba para las diferentes historias de usuario.

#### 3.1.1. Casos de Prueba de la Historia de usuario: HU-1

Caso de Prueba Aceptación	
<b>Código Caso de Prueba:</b> 1	<b>Nombre de la Historia de usuario:</b> Gestionar sesión.
<b>Nombre de la persona que realiza la prueba:</b> Juan Manuel Fuentes Rodríguez	
<b>Descripción de la prueba:</b> Se iniciará la sesión de Nova Oficial. Una vez iniciado se comprueba que se hayan ejecutado correctamente todos los componentes del entorno de escritorio (Nautilus, Gnome Shell y Mutter) y que se cargaron las configuraciones específicas para esta sesión.	
<b>Condiciones de ejecución:</b> No debe haber errores durante el levantamiento de la sesión y se deben cargar las configuraciones específicas establecidas por el usuario (o de ser la primera vez que se ejecuta el sistema la configuración por defecto).	

<b>Entrada / Pasos de ejecución:</b> Ejecutar cualquier gestor de acceso (GDM <sup>11</sup> , XDM <sup>12</sup> , KDM <sup>13</sup> , etc.) y seleccionar la sesión de Nova Oficial mediante este y autenticar un usuario.
<b>Resultado Esperado:</b> Se inicia la sesión con todas las aplicaciones y componentes en correcto funcionamiento.
<b>Evaluación de la prueba:</b> Satisfactoria.

Caso de Prueba Aceptación	
<b>Código Caso de Prueba:</b> 2	<b>Nombre de la Historia de usuario:</b> Gestionar sesión.
<b>Nombre de la persona que realiza la prueba:</b> Juan Manuel Fuentes Rodríguez	
<b>Descripción de la prueba:</b> Se iniciará la sesión de Nova Migración. Una vez iniciado se comprueba que se hayan ejecutado correctamente todos los componentes del entorno de escritorio (Nautilus, Nova Shell y Mutter) y que se cargaron las configuraciones específicas para esta sesión.	
<b>Condiciones de ejecución:</b> No debe haber errores durante el levantamiento de la sesión y se deben cargar las configuraciones por defecto de la sesión.	
<b>Entrada / Pasos de ejecución:</b> Ejecutar cualquier gestor de acceso (GDM, XDM, KDM, etc.) y seleccionar la sesión de Nova Migración mediante este y autenticar un usuario.	
<b>Resultado Esperado:</b> Se inicia la sesión con todas las aplicaciones y componentes en correcto funcionamiento.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Caso de Prueba Aceptación	
<b>Código Caso de Prueba:</b> 3	<b>Nombre de la Historia de usuario:</b> Gestionar sesión.
<b>Nombre de la persona que realiza la prueba:</b> Juan Manuel Fuentes Rodríguez	

11 Gnome Display Manager (Gestor de pantalla de Gnome).

12 X Display Manager (Gestor de pantalla X).

13 KDE Display Manager (Gestor de pantalla de KDE).

<b>Descripción de la prueba:</b> Se iniciará la sesión de Fallback. Una vez iniciado se comprueba que se hayan ejecutado correctamente todos los componentes del entorno de escritorio (Nautilus, Gnome-Panel y Metacity) y que se cargaron las configuraciones específicas para esta sesión.
<b>Condiciones de ejecución:</b> No debe haber errores durante el levantamiento de la sesión y se deben cargar las configuraciones específicas establecidas por el usuario (o de ser la primera vez que se ejecuta el sistema la configuración por defecto).
<b>Entrada / Pasos de ejecución:</b> Ejecutar cualquier gestor de acceso (GDM, XDM, KDM, etc.) y seleccionar la sesión de Fallback mediante este y autenticar un usuario.
<b>Resultado Esperado:</b> Se inicia la sesión con todas las aplicaciones y componentes en correcto funcionamiento.
<b>Evaluación de la prueba:</b> Satisfactoria.

### 3.1.2. Casos de Prueba de la Historia de usuario: HU-2

Caso de Prueba Aceptación	
<b>Código Caso de Prueba:</b> 3	<b>Nombre de la Historia de usuario:</b> Mostrar lista de indicadores.
<b>Nombre de la persona que realiza la prueba:</b> Juan Manuel Fuentes Rodríguez	
<b>Descripción de la prueba:</b> Al iniciarse y durante la sesión de Nova Migración se deben mostrar y actualizar un conjunto de indicadores que describen el estado de un proceso en específico. El entorno de escritorio por defecto debe ser capaz de mostrar algunos indicadores como: distribuciones del teclado, acceso universal, configuración del sonido, configuración del network-manager <sup>14</sup> , estado del bluetooth y la batería.	
<b>Condiciones de ejecución:</b> Debe estar previamente autenticado en el entorno de escritorio mediante la sesión de Nova Migración.	
<b>Entrada / Pasos de ejecución:</b> Ejecutar cualquier gestor de acceso (GDM, XDM, KDM, etc.) y seleccionar la sesión de Nova Migración mediante este y autenticar un usuario. Luego interactuar con los indicadores mostrados en la parte derecha del panel.	

<sup>14</sup> Software encargado de la gestión de la red en las distribuciones de GNU/Linux.

**Resultado Esperado:** Cada uno de los indicadores funciona correctamente.

**Evaluación de la prueba:** Satisfactoria.

### 3.1.3. Casos de Prueba de la Historia de usuario: HU-3

#### Caso de Prueba Aceptación

<b>Código Caso de Prueba:</b> 5	<b>Nombre de la Historia de usuario:</b> Mostrar la lista de notificaciones.
<b>Nombre de la persona que realiza la prueba:</b> Juan Manuel Fuentes Rodríguez	
<b>Descripción de la prueba:</b> Al iniciarse y durante la sesión de Nova Migración se deben mostrar y actualizar un conjunto de notificaciones que describen un suceso de algún proceso en segundo plano y que tienen importancia para el usuario o necesitan intervención de este. El entorno de escritorio debe ser capaz de notificar algunos eventos tales como: arribo de mensajes (mensajería instantánea o correo), fin de las descargas mediante el Firefox, arribo de actualizaciones a los repositorios.	
<b>Condiciones de ejecución:</b> Para que sean visibles las notificaciones se debe estar autenticado en el entorno de escritorio. La mensajería solo es posible si los clientes de mensajería instantánea o de correo están en ejecución. Si se desea que la mensajería instantánea esté integrada al sistema se deben usar clientes específicos como el Empathy, el Pidgin y el Gajim (estos dos últimos requieren una extensión del shell para ser usados)	
<b>Entrada / Pasos de ejecución:</b> Entrar a la sesión y probar cada uno de los programas que producen notificaciones para ver su comportamiento.	
<b>Resultado Esperado:</b> El entorno de escritorio debería notificar correctamente ante los eventos expuestos anteriormente.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

### 3.1.4. Casos de Prueba de la Historia de usuario: HU-4

#### Caso de Prueba Aceptación

<b>Código Caso de Prueba:</b> 6	<b>Nombre de la Historia de usuario:</b> Gestionar hora, fecha y calendario.
---------------------------------	--

<b>Nombre de la persona que realiza la prueba:</b> Juan Manuel Fuentes Rodríguez
<b>Descripción de la prueba:</b> Una vez iniciada la sesión de Nova migración, en la parte derecha del panel se debe mostrar un elemento que contenga la hora actual del sistema operativo y el día de la semana. Cuando se activa este elemento se debe mostrar un menú emergente que muestre los siguientes elementos: una etiqueta con la fecha completa, un almanaque, una etiqueta que permita acceder a las configuraciones de la hora y la fecha.
<b>Condiciones de ejecución:</b> Para hacer uso de este elemento se debe estar autenticado en el sistema.
<b>Entrada / Pasos de ejecución:</b> Autenticarse en el entorno de escritorio.
<b>Resultado Esperado:</b> El sistema debe mostrar una etiqueta en la esquina derecha del panel que muestre la hora y que levante un menú emergente que muestre la fecha, el calendario y de acceso a las configuraciones de la hora.
<b>Evaluación de la prueba:</b> Satisfactoria.

### 3.1.5. Casos de Prueba de la Historia de usuario: HU-5

Caso de Prueba Aceptación	
<b>Código Caso de Prueba:</b> 7	<b>Nombre de la Historia de usuario:</b> Gestionar navegación en los archivos.
<b>Nombre de la persona que realiza la prueba:</b> Juan Manuel Fuentes Rodríguez	
<b>Descripción de la prueba:</b> La sesión de Nova Migración debe tener en todo momento en ejecución al Nautilus, el programa que se encarga de la gestión de archivos y que brinda el escritorio.	
<b>Condiciones de ejecución:</b> Para hacer uso de este programa se debe estar autenticado en el sistema.	
<b>Entrada / Pasos de ejecución:</b> Autenticarse en el entorno de escritorio y hacer uso del escritorio y los archivos haciendo sobre estos las distintas operaciones (eliminar, renombrar, copiar, mover, abrir, etc.).	
<b>Resultado Esperado:</b> En el entorno de escritorio, el Nautilus se ejecuta correctamente brindándole al usuario un escritorio y permitiéndole realizar las diferentes operaciones con los archivos.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**3.1.6. Casos de Prueba de la Historia de usuario: HU-6**

<b>Caso de Prueba Aceptación</b>	
<b>Código Caso de Prueba:</b> 8	<b>Nombre de la Historia de usuario:</b> Gestionar la lista de ventanas abiertas.
<b>Nombre de la persona que realiza la prueba:</b> Juan Manuel Fuentes Rodríguez	
<b>Descripción de la prueba:</b> En la sesión de Nova Migración, en la parte izquierda del panel (después del menú principal de usuarios), debe aparecer la lista de aplicaciones abiertas y favoritas. Este debe mostrar el icono de estas aplicaciones. Al poner el puntero por encima a un icono este mostrar una previsualización de cada una de las ventanas abiertas de estas aplicaciones, posibilitando traer al frente una ventana mediante la selección de su previsualización y cerrarla mediante el icono de cerrar que se encuentra en la parte superior derecha. Al presionar el icono se debe mostrar un menú emergente que brinde acceso a las diferentes opciones con dicha aplicación (cerrar todas sus ventanas, anclarla o desanclarla y abrir una nueva instancia).	
<b>Condiciones de ejecución:</b> Para hacer uso de este programa se debe estar autenticado en el sistema.	
<b>Entrada / Pasos de ejecución:</b> Autenticarse en el entorno de escritorio y hacer uso de la lista de aplicaciones del panel, probando cada una de sus funcionalidades.	
<b>Resultado Esperado:</b> La lista de aplicaciones muestra todas las aplicaciones abiertas y las favoritas, actualizándose cuando se cierra o se abre una. Provee correctamente la previsualización de cada una de las ventanas abiertas de una aplicación al poner encima del icono el puntero y permite cerrar una ventana mediante el icono de cerrar o traerla al frente cuando se selecciona su previsualización. También debe proveer del menú auxiliar al seleccionar una aplicación con el botón derecho del mouse.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**3.1.7. Casos de Prueba de la Historia de usuario: HU-7**

<b>Caso de Prueba Aceptación</b>	
<b>Código Caso de Prueba:</b> 9	<b>Nombre de la Historia de usuario:</b> Gestionar la lista de aplicaciones instaladas.
<b>Nombre de la persona que realiza la prueba:</b> Juan Manuel Fuentes Rodríguez	

<p><b>Descripción de la prueba:</b> La sesión de Nova Migración debe tener un menú principal de usuario que permita acceder a las aplicaciones instaladas y que sea accesible desde el extremo izquierdo del panel. También es encargado de brindar acceso a los marcadores del sistema de archivos, a los principales directorios, a la configuración del sistema y a la configuración de la cuenta de usuario en uso, así como de brindar las opciones de apagado.</p>
<p><b>Condiciones de ejecución:</b> Para hacer uso de este programa se debe estar autenticado en el sistema.</p>
<p><b>Entrada / Pasos de ejecución:</b> Autenticarse en el entorno de escritorio y hacer uso de los elementos del menú principal de usuario: aplicaciones instaladas, accesos a los marcadores y principales directorios, opciones de apagado y configuración.</p>
<p><b>Resultado Esperado:</b> En el entorno de escritorio, el menú principal de usuario debe brindar acceso a todas las aplicaciones instaladas mediante el árbol de aplicaciones de Gnome, además brindar acceso a los marcadores, principales directorios y opciones de apagado y configuración.</p>
<p><b>Evaluación de la prueba:</b> Satisfactoria.</p>

## 3.2. Resultados Obtenidos.

Como resultado del desarrollo se obtuvo el entorno de escritorio del sistema operativo Nova en su versión 4.0, el cual brinda la posibilidad a los usuarios de usar diferentes sesiones, teniendo en cuenta sus conocimientos y experiencias previos. Adicionalmente se obtuvo la versión 0.1 del software Nova Shell.

### 3.2.1. Acerca de las funcionalidades obtenidas.

Entre las principales funcionalidades brindadas en el entorno de escritorio de Nova 4.0 se encuentran:

- Brindar varias sesiones para hacer un tratamiento diferenciado a los usuarios del sistema operativo.
- Una sesión hecha con el objetivo de brindar nuevas experiencias de usuario.
- Una sesión destinada para el proceso de migración que provee un sistema con grandes similitudes al entorno de escritorio de Microsoft Windows 7.
- Una sesión sin efectos gráficos y a prueba de errores para que sea ejecutada cuando por alguna razón no puedan ejecutarse correctamente ninguna de las sesiones anteriores.



## **Conclusiones parciales.**

Durante este capítulo se realizaron un conjunto de pruebas de aceptación a las historias de usuario con el objetivo de obtener un conjunto de funcionalidades validadas esenciales para la entrega de un producto con calidad.

## **CONCLUSIONES GENERALES.**

Mediante este trabajo se hizo una investigación de los principales elementos a tener en cuenta para el entorno de escritorio de Nova, realizando un estudio del estado del arte de las principales características que tienen los entornos de escritorio modernos, así como de un conjunto de conceptos que rigen la evolución de los entornos de escritorio.

Además se cumplieron los objetivos de la investigación al obtener un entorno de escritorio validado capaz de tratar diferenciadamente a los usuarios de la migración, dando como resultado dos sesiones principales, una para brindar nuevas experiencias de usuario y otra, desarrollada basándose en esta última, que brinda una experiencia de usuario con una gran similitud a la que se brinda en Windows 7, combinando así, el estilo visual de los entornos de escritorio de Microsoft con las posibilidades que brindan los entornos de escritorios para distribuciones de GNU/Linux y proponiendo una manera innovadora para disminuir la resistencia al cambio en el proceso de migración. Adicionalmente se brinda una sesión extra para que esta iniciada cuando no se puedan ejecutar ninguna de las dos anteriores.

El desarrollo de este trabajo abre una nueva etapa en el sistema operativo Nova, puesto que con la asimilación de los conocimientos sobre las tecnologías utilizadas se hace posible la realización de entornos de escritorio a la medida, sin estar atados a las tecnologías utilizadas anteriormente.

Junto con el desarrollo, la retroalimentación con la comunidad del software libre y las autoridades para la migración a estándares abiertos en Cuba, sirvió para constatar la opinión de los usuarios y especialistas de dicho proceso con respecto a la investigación realizada.

El lanzamiento de este producto mejorará la situación existente en el proceso de migración a estándares abiertos que se realiza en el país, ya que permitirá el uso del sistema operativo Nova por la mayoría de los usuarios sin importar sus experiencias y conocimientos acerca de los entornos de escritorio del software libre.

## **RECOMENDACIONES.**

El autor de este trabajo recomienda:

- Profundizar en la investigación de las limitaciones de la metáfora de escritorio con el objetivo de desarrollar nuevas funcionalidades que limen dichas incongruencias.
- Profundizar en la investigación del entorno de Gnome Shell buscando otros aportes que son necesarios y requeridos por la comunidad internacional.
- Profundizar en la investigación del entorno de Gnome Shell para el desarrollo de extensiones que brinden mayor adaptabilidad e interoperabilidad con los servicios consumidos por los usuarios de los OACEs de Cuba.
- Valorar la posibilidad de implementar un gestor de configuraciones propio para la sesión de la migración.
- Mejorar la integración a través de Gobject Introspection con Zeitgeist para poder implementar funcionalidades eficientes con los archivos recientes.

## REFERENCIAS.

- [1]. **Pierra, A.** *Nova, distribución cubana de GNU/Linux. Reestructuración estratégica de su proceso de desarrollo.* Universidad de las Ciencias Informáticas, La Habana, 2011.
- [2]. **KAPTELININ, V. y CZERWINSKI, M.** *Beyond the Desktop Metaphor.* Massachusetts : MIT Press, 2007. ISBN-13: 978-0-262-11304-5.
- [3]. ISO 9241-11:1998: Ergonomic requirements for office work with visual display terminals (VDT) - Part 11 Guidance on usability, 1998.
- [4]. **MONTERO, Y. y ORTEGA SANTAMARÍA, S.** *Informe APEI sobre usabilidad.* Guijón: APEI, 2009. ISBN: 978-84-692-3782-3.
- [5]. **D'HERTEFELT, S.** Emerging and future usability challenges: designing user experiences and user communities. [En línea]. 2000. [Consultado el: 15 de diciembre del 2011]. Disponible en <http://www.interactionarchitect.com/future/vision20000202shd.htm>.
- [6]. **RODRÍGUEZ, A. y FÍRVIDA, A.** *Guano, entorno de escritorio cubano, libre y de código abierto.* Universidad de las Ciencias Informáticas, La Habana. 2009.
- [7]. **GnomeShell/Extensions - Gnome Live!**. [En línea]. 30 de marzo del 2012. [Consultado el: 10 de abril del 2012]. Disponible en <https://live.gnome.org/GnomeShell/Extensions>.
- [8]. **GnomeShell/LookingGlass - Gnome Live!**. [En línea] 15 de marzo del 2011. [Consultado el: 30 de noviembre del 2011]. Disponible en <https://live.gnome.org/GnomeShell/LookingGlass>.
- [9]. **SessionManagement/GnomeSession - Gnome Live!**. [En línea]. 4 de marzo del 2009. [Consultado el: 1 de diciembre del 2011]. Disponible en <http://live.gnome.org/SessionManagement/GnomeSession>.
- [10]. **Introducción a las Sesiones - Gnome Library.** [En línea] Marzo del 2004. [Consultado el: 1 de diciembre del 2011]. Disponible en <http://library.gnome.org/admin/system-admin-guide/stable/sessions-1.html.es>.
- [11]. **MYERS, B.** *Window Interfaces: Taxonomy of Window Manager User Interfaces.* 1988.
- [12]. **KRUCHTEN, P.** *The Rational Unified Process: An Introduction.* Addison Wesley. 2000.
- [13]. **JACOBSON, BOOCH, G., RUMBAUGH, J.** *El Proceso Unificado de Desarrollo.* Traducido por: PERSON EDUCACIÓN, S.A. Traducido de: THE UNITED SOFTWARE DEVELOPMENT PROCESS.

Addision Wesley. 2000. ISBN: 0-201-57169-2.

[14]. **BECK, K.** *Extreme Programming Explained. Embrace Change.* 1999.

[15]. **CANÓS, J., LETELIER, P. y PENADÉS, M.** *Metodologías Ágiles en el Desarrollo de Software .*

[16]. **PEÑALVER ROMERO, G.** *Metodología ágil para proyectos de software libre.* Universidad de las Ciencias Informáticas. La Habana. 2008.

[17]. About JavaScript. Mozilla Developer Network. [En línea]. 6 de Octubre del 2011. [Consultado el: 10 de diciembre del 2011]. Disponible en [https://developer.mozilla.org/en/About\\_JavaScript](https://developer.mozilla.org/en/About_JavaScript).

[18]. **SANTOS ESPINO, J.** *Introducción al lenguaje C.* [Documento] .

[19]. GobjectIntrospection - Gnome Live!. [En línea]. 31 de enero del 2012. [Consultado el: 22 de febrero del 2012]. Disponible en <https://live.gnome.org/GobjectIntrospection>.

[20]. Clutter - Gnome Live! [En línea]. 19 de mayo del 2011. [Consultado el : 15 de enero del 2012]. Disponible en <https://live.gnome.org/Clutter>.

## BIBLIOGRAFÍA.

1. ISO 9241-171:2008.Ergonomics of human-system interaction -- Part 171: Guidance on software accessibility. 2008.
2. Desktop Methaphor. [En línea]. 1998. [Consultado el: 25 de enero del 2012]. Disponible en <http://www.csd.tamu.edu/~l0f0954/academic/cpsc610/hw2-3.htm>.
3. Desktop Environment. [En línea]. 2012. [Consultado el: 23 de marzo del 2012]. Disponible en [https://wiki.archlinux.org/index.php/Desktop\\_Environment](https://wiki.archlinux.org/index.php/Desktop_Environment).
4. Desktops. [En línea]. 2012. [Consultado el: 25 de febrero del 2012]. Disponible en <http://www.freedesktop.org/wiki/Desktops>.
5. ¿Qué es Aero de Windows? [En línea]. [Consultado el 12 de octubre del 2011]. Disponible en: <http://windows.microsoft.com/es-xl/windows-vista/What-is-Windows-Aero>
6. KDE – About KDE. [En línea]. [Consultado el: 15 de diciembre del 2011]. Disponible en <http://www.kde.org/community/whatiskde/>.
7. Escritorio de GNOME 2.0 para el sistema operativo Solaris: Guía del usuario. [En línea]. [Consultado el: 13 de noviembre del 2012]. Disponible en <http://docs.oracle.com/cd/E19754-01/817-4012/6mjlr9p8/index.html>.
8. **ALBALAT, M. y RAMÍREZ, B.** *Solución para hacer de Guano un entorno de escritorio usable*. Universidad de las Ciencias Informáticas. La Habana. 2009.
9. DistroWatch Page Hit Ranking. [En línea]. [ Consultado el: 12 de febrero del 2012]. Disponible en <http://distrowatch.com/dwres.php?resource=popularity>.
10. **Gómez, Y.** Por dónde va. Trabajadores. [En línea]. 15 de agosto del 2011. [Consultado el: 15 de enero del 2012]. Disponible en: <http://www.trabajadores.cu/news/2011/08/15/por-donde-va>.
11. **Suzarte, S.** La emigración al software libre es un asunto político. Trabajadores. [En línea]. 23 de febrero del 2012. [Consultado el: 1 de marzo del 2012]. Disponible en: <http://www.trabajadores.cu/news/20120223/257886-la-emigracion-al-software-libre-es-un-asunto-polit>.
12. **Suzarte, S.** Frente al bloqueo de Windows, NOVA sí va. Cubadebate. [En línea]. 13 de enero del 2010. [Consultado el: 15 de noviembre del 2011]. Disponible en:

<http://www.cubadebate.cu/especiales/2010/01/13/cuba-frente-bloqueo-windows-nova/>.