



Facultad 1

Título: Propuesta de Arquitectura de la Información del Entorno de escritorio para Nova Escritorio 5.0.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

*Autores: Katileidys Zerquera Canales
Iván Romay Aragón*

*Tutores: Ms.C. Yusleydi Fernández del Monte
Ing. Mairim Delgado Muñiz
Ing. Juan Manuel Fuentes Rodríguez*

*La Habana, Cuba
Junio 2014*

Declaración de autoría

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

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

Katileidys Zerquera Canales

Iván Romay Aragón

Ms.C. Yusleydi Fernández del Monte

Ing. Mairim Delgado Muñiz

Ing. Juan Manuel Fuentes Rodríguez

Nombre y Apellidos: Ms.C. Yusleydi Fernández del Monte

Email: ydelmonte@uci.cu

Especialidad de graduación: Ingeniería en Ciencias Informáticas. Master en Gestión de Proyectos Informáticos.

Categoría docente: Instructora.

Categoría Científica: Investigadora.

Años de experiencia en el tema: 5 años

Años de graduado: 5 años.

Nombre y Apellidos: Ing. Mairim Delgado Muñoz

Email: mdelgado@uci.cu

Especialidad de graduación: Ingeniería en Ciencias Informáticas.

Categoría Científica: Investigadora.

Años de experiencia en el tema: 3 años

Años de graduado: 2 años.

Nombre y Apellidos: Ing. Juan Manuel Fuentes Rodríguez.

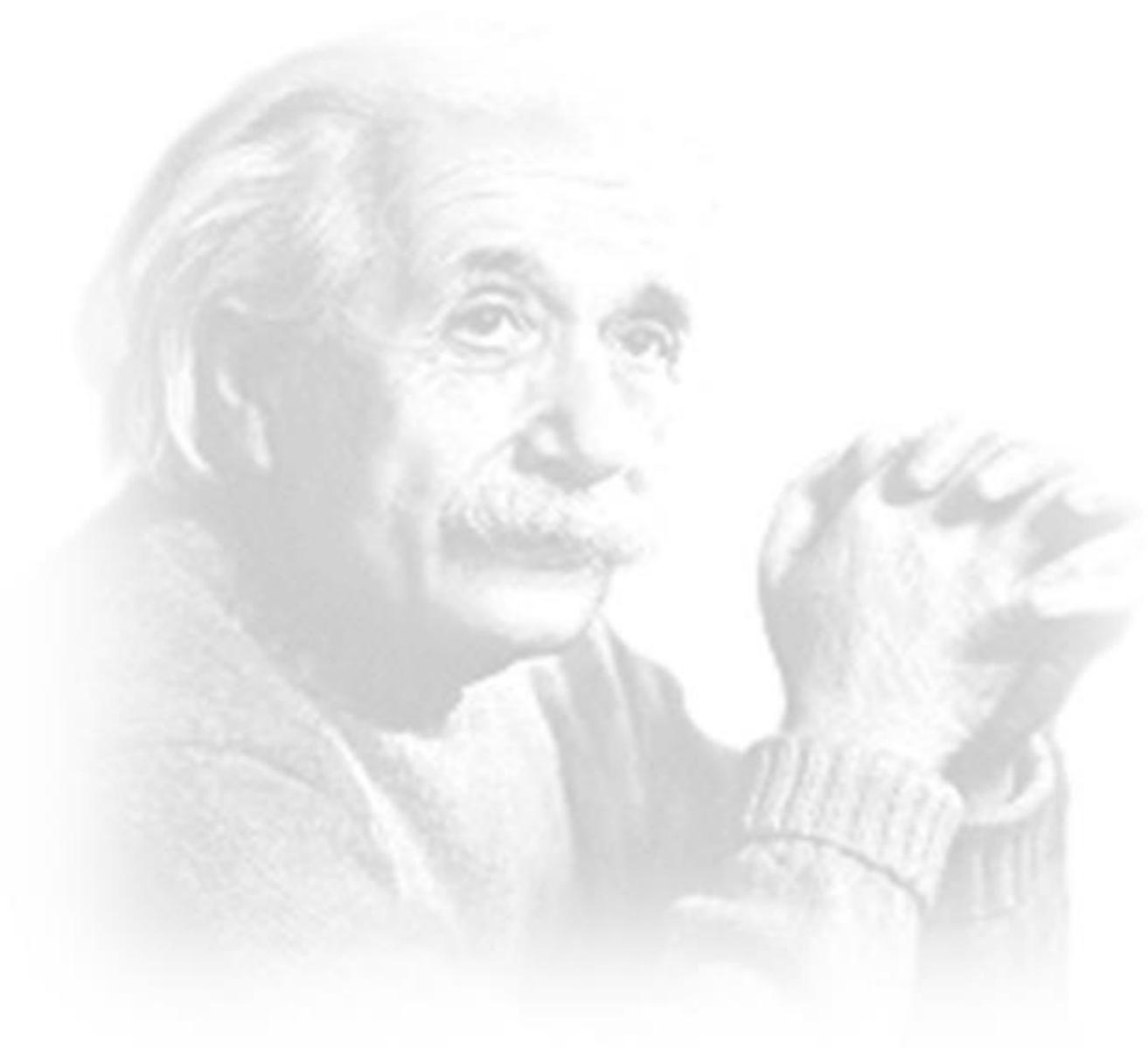
Email: jfuentesr@uci.cu

Especialidad de graduación: Ingeniería en Ciencias Informáticas.

Categoría Científica: Investigador.

Años de experiencia en el tema: 3 años

Años de graduado: 2 años.



*“Si tu intención es describir la verdad, hazlo con sencillez y
la elegancia déjasela al sastre”
Albert Einstein*

Agradecimientos

A mi abuela y mi mamá por regalarme cada día su amor y hacerme sentir la persona más dichosa de este mundo.

A mi abuelo por ser siempre mi ejemplo y guía.

A mi papá por ser especial para mí.

A mis hermanitos por dejarme ser una parte importante de sus vidas.

A mis tíos Ariel, Benitin, Maidelis y Dayana por regalarme su amor incondicional.

A mis primas Yelsy, Yeni y Yani por estar siempre pendientes de mí y apoyarme en los momentos difíciles.

A Dinorah, Yoel, Mario y Sonia por todo el cariño y la alegría que me brindan.

A mi novio por estar siempre a mi lado y apoyarme en todo momento.

A dos personitas muy especiales en mi vida Iván y Wendy.

A mis compañeros de aula que tantos momentos buenos y malos compartimos juntos en especial a Yosvany, Hanoi, Yamilka, Dainelys e Isabel.

A los tutores que con tanta dedicación y empeño hicieron posible la realización de mi sueño.

A Haniel por ayudarme con la realización de la tesis y ser un buen amigo en todo momento.

A los profesores que año tras año aportaron su conocimiento, experiencia y amor a mi formación.

Y en especial a la Revolución que me permitió formarme como profesional.

A todos ¡Muchas Gracias!

Katy.

Agradecimientos

A mis padres por estar siempre cuando los necesito, por darme su amor, por confiar en mí y porque gracias a ellos hoy estoy aquí. Los amo.

A mis hermanos que los quiero muchísimo y quiero ser un ejemplo para ellos.

A mis abuelos Berta y Osvaldo por su preocupación y dedicación y por ser tan especiales para mí.

A mi abuelo Iván y a mi abuela Fidélina esté donde esté, porque sé que se encuentran orgullosos de mí.

A mis tíos y primos por darme su apoyo y estar presentes cuando los necesité.

A mi novia Wendy por apoyarme y aguantarme en los momentos más difíciles. Te amo.

A mi amiga y compañera de tesis Katy, por su amistad y por estar presente en estos 5 años maravillosos.

A mis compañeros de aula por tantos momentos que pasamos juntos, en especial a Yamilka, Isabel, Dainelys Yosvany, Hanoi y Gustavo.

A Iosvannis y a Eymeé por su amistad y por ser parte de estos 5 años.

A los tutores y a todas las personas que nos ayudaron a realizar esta tesis, porque sin ellos no hubiera sido posible.

A los profesores que aportaron a mi formación como profesional.

A la Revolución, a Fidel y a la UCI por darme la oportunidad de realizar este sueño.

Y a los que me faltaron... de todo corazón:

¡Muchas Gracias!

Iván.

Este trabajo de diploma está dedicado a:

Katy

A mi Abuela y mi mamá por ser la ternura, la experiencia y la alegría de mi vida.

A mi abuelo y mi papá por ser mi ejemplo a seguir y por confiar en mí.

A mis hermanitos que los adoro.

Iván

A mi mamá por ser mi guía y ejemplo en todo momento, por hacer de mí una persona de bien y por enseñarme a luchar por ser cada día mejor.

A mi papá por estar siempre cuando lo necesito y por apoyarme en todo.

A mis abuelos Berta y Osvaldo por todo el amor que me han dado y por malcriarme tanto.

A mis hermanos que los quiero muchísimo.

La distribución cubana de Software Libre GNU/Linux Nova, es actualmente la alternativa propuesta por la Guía cubana de Migración a Software Libre y Código Abierto, para sustituir en su totalidad los sistemas propietarios utilizados en las empresas cubanas. Desde sus inicios ha mostrado deficiencias y limitaciones en la Arquitectura de la Información de su Entorno de escritorio, que han traído consigo que los usuarios manifiesten inconformidad con los niveles de accesibilidad y organización de la información en el sistema y muestren resistencia al cambio hacia un Entorno de escritorio que no les brinda facilidades de interacción con la interfaz de usuario. Por otra parte la ausencia de elementos visuales y de ayuda necesarios para realizar las tareas, dificulta de alguna manera, la navegación y la usabilidad. Con el propósito de contribuir a la calidad de la distribución el objetivo que se plantea es proponer cambios para la Arquitectura de la Información del Entorno de escritorio de Nova Escritorio 5.0. El análisis crítico de las teorías asociadas a la investigación arrojó como resultado que existe una correspondencia entre las principales potencialidades que tienen los Entornos de escritorio más populares y las deficiencias que tienen las versiones de Nova Escritorio. Para darle una mayor validez a la investigación se utilizan los métodos científicos: Analítico-sintético, Análisis histórico-lógico, Modelado, Entrevista y Encuesta, así como la metodología SXP para guiar los cambios que se desarrollan.

Palabras clave: Arquitectura de la Información, Entorno de escritorio, Nova Escritorio.

Contenido

Introducción	1
Capítulo 1: Fundamentación teórica de la Arquitectura de la Información del Entorno de escritorio de Sistemas Operativos.....	5
Introducción.....	5
1.1 Algunas definiciones de Arquitectura de la Información	5
1.2 Pilares fundamentales de Arquitectura de la Información	5
1.3 Elementos que componen un Entorno de escritorio	6
1.4 Análisis de las distribuciones más populares de GNU/Linux, Windows y Mac.	7
1.5 Evolución de la AI de Nova Escritorio 3.0 a Nova Escritorio 4.0.....	16
1.6 Tecnologías para desarrollar soluciones de la AI.....	19
1.6.1 Metodología de desarrollo	19
1.6.2 Lenguajes de Programación	19
1.6.3 Bibliotecas fundamentales a utilizar	20
1.6.4 Herramientas	21
Conclusiones Parciales	22
Capítulo 2: Propuesta de cambios para la AI del Entorno de escritorio de Nova Escritorio 5.0.	23
Introducción.....	23
2.1 Descripción de la solución propuesta para la AI del Entorno de escritorio de Nova Escritorio 5.0.	23
2.2 Requisitos funcionales.	25
2.3 Historias de usuarios (HU)	25
2.4 Implementación de los requisitos	31
2.4.1 Vistas de implementación	31
2.4.2 Programación por pares	38
Conclusiones Parciales	38
Capítulo 3: Evaluación de los cambios para la AI del Entorno de escritorio de Nova Escritorio 5.0.....	39
Introducción.....	39
3.1 Pruebas unitarias.....	39
3.2 Pruebas de funcionalidad	48
3.3 Pruebas de aceptación.	54
Conclusiones Parciales	55
Conclusiones	56

Recomendaciones	57
Bibliografía.....	58

Tablas

Tabla 1. Comparación entre distribuciones.	15
Tabla 2. Evolución de la AI del Entorno de escritorio de Nova Escritorio.	18
Tabla 3. HU_1 agregar los botones maximizar y minimizar en la barra de título.	26
Tabla 4. HU_2 mostrar la sección de todas las aplicaciones en el menú agrupadas por categorías.	26
Tabla 5. HU_3 Modificar el botón Actividades del sistema.	27
Tabla 6. HU_4 adicionar el botón de apagado del sistema al Panel de menú para las computadoras portátiles.	28
Tabla 7. HU_5 soporte para el método de la red cableada.	28
Tabla 8. HU_6 modificar el ícono de Accesibilidad.....	29
Tabla 9. HU_7 adicionar nueva forma para agregar y eliminar las aplicaciones a Favoritos.	30
Tabla 10. HU_8 modificar el Alt+Tab.	30
Tabla 11. Diseño de caso de prueba de Mostrar la sección de todas las aplicaciones en el menú agrupadas por categorías.	49
Tabla 12. Diseño de caso de prueba de Modificar el botón Actividades del sistema.	49
Tabla 13. Adicionar el botón de apagado del sistema al Panel de menú para las computadoras portátiles.	50
Tabla 14. Diseño de caso de prueba de Soporte para el método de la red cableada.	50
Tabla 15. Diseño de caso de prueba de Modificar el ícono Accesibilidad del sistema.....	51
Tabla 16. Diseño de caso de prueba de Agregar los botones minimizar y maximizar.	51
Tabla 17. Diseño de caso de prueba Modificar el Alt+Tab.	53
Tabla 18. Diseño de caso de prueba Adicionar nueva forma para agregar y eliminar las aplicaciones a Favoritos.....	53

Figuras

Figura 1. Tendencias de búsquedas.	8
Figura 2. Entorno de escritorio de Nova Escritorio 5.0.	23
Figura 3. Diagrama de funcionamiento del Entorno de escritorio Gnome.....	24
Figura 4. Diagrama de componentes de soporte para el método de la red cableada.	31
Figura 5. Diagrama de componentes de adicionar el botón de apagado del sistema al Panel de menú para una computadora portátil.	32

Figura 6. Diagrama de componentes de mostrar la sección de todas las aplicaciones en el menú agrupadas por categorías.	33
Figura 7. Diagrama de componentes de modificar el botón Actividades del sistema.....	34
Figura 8. Diagrama de componentes modificar el ícono Accesibilidad del sistema.	35
Figura 9. Diagrama de componentes modificar el Alt+Tab.	36
Figura 10. Diagrama de componentes adicionar nueva forma para agregar y eliminar las aplicaciones a Favoritos.	37
Figura 11. Gráfico de flujo de la funcionalidad getIndicatorIcon.	41
Figura 12. Gráfico de flujo de la funcionalidad syncIcon.....	43
Figura 13. Gráfico de flujo de la funcionalidad onStarPress.	45
Figura 14. Gráfico de flujo de la funcionalidad contCategory.	47
Figura 15. Gráfica representativa de los resultados de las pruebas de funcionalidad.	54
Figura 16. Gráfica representativa de los resultados de las pruebas de aceptación.	55

La revolución de las Tecnologías de la Información ha marcado un momento crucial y decisivo en la sociedad mundial, pues ha penetrado en todas las áreas de la vida humana, no como agente externo, sino también como motor que genera un flujo activo en las interrelaciones sociales.

Las Tecnologías de la Información han abierto un territorio en el cual la mente humana es la fuerza productiva directa de mayor importancia en la actualidad. Lo cual, ha traído consigo la creación de un sin número de productos informáticos con el fin de satisfacer las demandas de los usuarios, que exigen una respuesta cada vez más rápida, directa y relevante a sus necesidades de información.

Con el objetivo de hacer más comprensible dicha información y permitir al usuario encontrar una vía de navegación hacia el conocimiento, se comienzan a realizar los primeros estudios entorno a la organización de la información. En el año 1976 Richard Saul Wurman acuña el término “Arquitectura de Información” para describir lo que percibía entonces como una profesión emergente, la cual se ocuparía de aclarar lo complejo y de organizar “el tsunami de datos que rompe en las playas del mundo civilizado”. (Bustamante, 2003)

La Arquitectura de Información adquiere en este contexto un papel protagónico. Ella, como disciplina, crea las bases y aporta los principios esenciales para el diseño adecuado de productos informáticos, así como para la representación y organización de la información en forma coherente, como única vía posible de materializar un acceso ágil de los usuarios a los contenidos que requieren para su labor en cada momento. Además de lograr una gestión adecuada de la información y el conocimiento.

Uno de estos productos es la distribución cubana de Software Libre GNU/Linux Nova, que se ha convertido en la alternativa propuesta por la Guía cubana de Migración a Software Libre y Código Abierto, para sustituir en su totalidad los sistemas propietarios utilizados en las empresas cubanas. En el año 2009 se da a conocer la primera versión estable de Nova Escritorio (2.0), con motivo de la Feria y Convención Internacional “Informática Habana 2009”, en la que varias instituciones se acercaron para mostrar su interés por el nuevo producto, incrementando así el alcance del mismo (Machín Castillo y Corso Bouza, 2010).

En el año 2011 se lanza la versión 3.0 con el Entorno de escritorio basado en Gnome 2.0. Mediante un estudio que se realiza teniendo en cuenta los elementos que componen un Entorno de escritorio, se detectan un conjunto de deficiencias y limitaciones en la Arquitectura de la Información del mismo, entre las que se destacan el menú de aplicaciones, que le permitía al usuario tener solo 6 aplicaciones favoritas, 3 documentos recientes y el resto de las aplicaciones se mostraban en una ventana aparte. Otras de las deficiencias que se encuentran es que existía poca integración con los elementos del sistema, no se aprovechaba completamente el área del escritorio y solo se podía trabajar en 4 de estas.

Para el año 2013 se lanza la versión 4.0 con un Entorno de escritorio totalmente nuevo, debido a que el anterior ya no recibía soporte. Este elimina gran parte de las deficiencias de su antecesor como es el caso del menú de aplicaciones, que permite tener la cantidad de aplicaciones Favoritas que el usuario desee y ya no es necesario abrir una nueva ventana para ver las demás, se aprovecha al 100% el área del escritorio y se aumenta el número de documentos recientes, así como la cantidad de áreas de trabajo con las que puede interactuar el usuario. También se corrigen algunos elementos de integración como el Chat y la Gestión de tareas en el calendario, pero aún no se han resuelto otros como el Correo electrónico y las actualizaciones del sistema.

Las deficiencias detectadas en Nova 3.0, fueron corregidas en su mayoría con la llegada del Entorno de escritorio basado en Gnome 3.4 en la versión 4.0 del producto. Aun así presenta otros problemas relacionados con la interfaz de usuario que se describen a continuación:

Al instalar el sistema en una computadora portátil no se muestra el botón de apagado en el Panel de menú; no se muestra el nombre completo de las aplicaciones en el menú; no está implementado el método de la red cableada; cuando existen más de 10 tareas programadas en el calendario, la ventana que contiene al menú comienza a agrandar su tamaño hasta que ocupa toda la altura del escritorio y la guía de usuario que aparece al presionar F1 muestra las especificaciones de Ubuntu y no del sistema.

Los problemas identificados han traído consigo que los usuarios manifiesten inconformidad con los niveles de accesibilidad y organización de la información en el sistema y muestren resistencia al cambio hacia un Entorno de escritorio que no les brinda las facilidades de interacción con la interfaz de usuario. Por otra parte, la ausencia de elementos visuales y de ayuda necesarios para realizar las tareas, dificulta de alguna manera, la navegación y la usabilidad.

Tomando como punto de partida la problemática planteada se identifica como **problema a resolver**: ¿cómo aumentar la calidad de la Arquitectura de la Información del Entorno de escritorio para Nova Escritorio 5.0?

Por tanto, el **objeto de estudio** está centrado en la Arquitectura de Información de los Entornos de escritorio de Sistemas Operativos. El **campo de acción** queda enmarcado en la Arquitectura de Información del Entorno de escritorio de Nova Escritorio 5.0.

Objetivo general: proponer cambios para la Arquitectura de la Información del Entorno de escritorio de Nova Escritorio 5.0 para contribuir a la calidad de la distribución.

Para sustentar el propósito de la investigación se plantea la siguiente **idea a defender**: la propuesta de cambios para la Arquitectura de la Información del Entorno de escritorio de Nova Escritorio 5.0 contribuirá a la calidad de la distribución.

Objetivos específicos:

- Fundamentar teóricamente la Arquitectura de Información de Entornos de escritorio de Sistemas Operativos.

- Identificar cambios para el Entorno de escritorio de Nova Escritorio 5.0.
- Desarrollar cambios para el Entorno de escritorio de Nova Escritorio 5.0.
- Evaluar los cambios para el Entorno de escritorio de Nova Escritorio 5.0.

Tareas específicas de la investigación:

- Estudio del estado del arte sobre las tendencias de la Arquitectura de la Información del Entorno de escritorio de Sistemas Operativos para conocer el estado real de su evolución en la actualidad.
- Definición de los pilares fundamentales e infraestructura tecnológica para la representación de la Arquitectura de Información del Entorno de escritorio de distribuciones de GNU/Linux.
- Análisis de la experiencia de los usuarios finales en la actual Arquitectura de Información del Entorno de escritorio de Nova Escritorio 4.0, para determinar elementos importantes que permitan solucionar el problema identificado.
- Identificación de cambios que faciliten la interacción con los elementos del Entorno de escritorio para la Arquitectura de la Información del Entorno de escritorio de Nova Escritorio 5.0.
- Diseño de los cambios de mayor prioridad para el Entorno de escritorio de Nova Escritorio 5.0 para esbozar los componentes de software que intervienen en los mismos.
- Implementación de los cambios para la Arquitectura de Información del Entorno de escritorio de Nova Escritorio 5.0, para facilitar la interacción con los elementos del Entorno de escritorio.
- Evaluación de los cambios para la Arquitectura de Información del Entorno de escritorio de Nova Escritorio 5.0, para verificar su correcto funcionamiento y la correspondencia con el criterio de los usuarios finales.

Posibles resultados: Arquitectura de la Información del Entorno de escritorio para Nova Escritorio 5.0.

Métodos Teóricos

Análítico-sintético: Se emplea para analizar la información encontrada sobre los rasgos distintivos de la Arquitectura de la Información, específicamente de Entornos de escritorio. Además, permite extraer elementos generales y específicos para establecer una relación entre los diferentes criterios que puedan existir y tributen a la elaboración de una correcta solución para la Arquitectura de Información del Entorno de escritorio de Nova Escritorio 5.0.

Análisis histórico-lógico: Se utiliza para analizar el estado del arte relacionado con la Arquitectura de Información de un Entorno de escritorio, sus principales conceptos, ventajas, desventajas y elementos que definen una buena organización y estructuración de la información de un sistema.

Modelado: Permite modelar el proceso a seguir para realizar los cambios en la Arquitectura de la Información del Entorno de escritorio para Nova Escritorio 5.0, creando una abstracción con el objetivo de explicar la realidad.

Métodos Empíricos

Entrevista: Conversación planificada con especialistas de Arquitectura de Información con el fin de obtener información valiosa para el desarrollo de la investigación. (Ver Anexo 1)

Encuestas: Permiten obtener información respecto al objeto de estudio, a partir de personas con un conocimiento básico en el tema, así como la valoración y evaluación teórica de la solución propuesta teniendo en cuenta la situación antes y después de realizar los cambios. (Ver Anexo 2)

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

Capítulo 1: Fundamentación teórica de la Arquitectura de la Información de Entornos de escritorio de Sistemas Operativos. En este capítulo se exponen los conceptos y fundamentos generales que sirven de soporte teórico a la propuesta de solución que se plantea, así como los principales métodos y herramientas que intervienen en la confección de una Arquitectura de la Información.

Capítulo 2: Propuesta de cambios para la Arquitectura de Información del Entorno de escritorio de Nova Escritorio 5.0. Se presentan los procedimientos a realizar y los componentes a modificar para desarrollar algunos de los cambios propuestos, asegurando su correcta documentación.

Capítulo 3: Evaluación de los cambios desarrollados para la Arquitectura de la Información del Entorno de escritorio de Nova Escritorio 5.0. En este capítulo se evalúan los cambios realizados en la Arquitectura de la Información del Entorno de escritorio para Nova Escritorio 5.0 con el fin de verificar su correspondencia con el criterio de los usuarios finales y el buen funcionamiento de los mismos.

Capítulo 1: Fundamentación teórica de la Arquitectura de la Información del Entorno de escritorio de Sistemas Operativos.

Introducción

En este capítulo se va a efectuar un estudio sobre los principales elementos que se deben tener en cuenta para realizar la Arquitectura de Información (AI) de un Entorno de escritorio. Además se van a caracterizar las herramientas, tecnologías, lenguaje de programación y la metodología de desarrollo de software a utilizar para el desarrollo de los cambios.

1.1 Algunas definiciones de Arquitectura de la Información

Richard Saul Wurman es conocido como el padre de la AI debido a los aportes que realizó a esta disciplina. En 1996, en su libro *"Information Architects"* acuña el término "Arquitectura de la Información" como una combinación de la organización de la información del contenido del sitio en categorías y la creación de una interfaz para sostener esas categorías. (Ronda León, 2005)

Como precursores de esta terminología se encuentran Louis Rosenfeld y Peter Morville, quienes la definen en su libro *"Information Architecture for the World Wide Web"* en 1998 como el proceso que: "... Clarifica la misión y visión del sitio, equilibrando las necesidades del patrocinador y las necesidades de la audiencia...", "...determina el contenido y funcionalidad que el sitio va a tener...", "... especifica cómo los usuarios van a encontrar la información al definir su organización, navegación, etiquetado y sistemas de búsqueda...", "...representa cómo el sitio se va a acomodar al cambio y crecimiento en el tiempo..." (Rosenfeld y Morville, 2002)

Otro de los precursores es Steve Toub de *Argus Associates*, el cual la define en su artículo *"AP Practical Guide To Assessing Web Site Organization"* como: "El arte y ciencia de estructurar y organizar el entorno informativo, para ayudar a los usuarios eficientemente a satisfacer sus necesidades informativas." (Toub, 2000)

Después de revisar la bibliografía y teniendo en cuenta las definiciones de los distintos autores que se citan anteriormente, los realizadores de la presente investigación definen la AI como la disciplina que se encarga de estructurar y organizar la información que va a poseer un sistema informático, con el objetivo de lograr la calidad deseada del producto final y con ello la plena satisfacción de los usuarios al facilitar la realización de sus tareas.

1.2 Pilares fundamentales de Arquitectura de la Información

Organización: ¿Cómo puede ordenarse la información?

Usualmente la computadora guarda en forma automática nuestros documentos en un área predeterminada, que es "Descargas". Cada documento, sea un texto, planilla o una imagen se guarda como un archivo. Cada archivo tiene un nombre y formato de acuerdo a las características de su contenido. Por lo que un recurso fundamental del sistema operativo es mostrar las carpetas con los

directorios de los archivos que contiene, y a su vez, si contiene otras carpetas, mostrarlas también con sus correspondientes directorios para darle mayor información al usuario. (Tenzer, 2008)

Navegación: Es la interactividad que posee un Sistema Operativo tal que permita a sus usuarios moverse por su estructura e identificar las diferentes secciones y contenidos de una forma sencilla, efectiva y sin perderse en él. (Romero, 2013)

Usabilidad: Es la capacidad con la que un Sistema Operativo puede ser usado por diferentes usuarios para conseguir sus objetivos en un determinado contexto de uso y se determina por los atributos:

Efectividad: estudia si los usuarios son capaces de cumplir las tareas de forma completa y precisa.

Eficiencia: analiza la cantidad de recursos usados para completar tareas eficientemente.

Satisfacción: estudia la actitud del usuario frente al producto.

Hay cuatro factores que nos permiten evaluar la usabilidad de un producto o sistema: la utilidad, la facilidad de aprendizaje, el rendimiento y la satisfacción de sus usuarios. (Marcos Mora y Rovira Fontanals, 2005)

Ayuda: Está diseñada para hacer búsqueda directa. Se requiere de una palabra clave para que el menú o el manual de ayuda nos puedan ser de utilidad, necesitamos conocer el nombre del comando para poder hacer referencia a él o buscarlo dentro del menú mencionado.

1.3 Elementos que componen un Entorno de escritorio

Un Entorno de escritorio es un conjunto de software especializado en brindar al usuario una interacción cómoda y amigable con el sistema operativo de la computadora. Básicamente es la capa que permite la comunicación entre el usuario y el Sistema Operativo, siempre teniendo en cuenta la simplicidad y la facilidad (Fuentes Rodríguez, 2012). Estos se componen por un conjunto de elementos que se mencionan a continuación:

Paneles

Los paneles son áreas del entorno del escritorio en las cuales se puede acceder a todas las aplicaciones del sistema y los menús. Los paneles ofrecen muchas posibilidades de configuración. Un panel particularmente importante es el Panel de menú que ocupa el ancho del borde superior del escritorio. El Panel de menú tiene dos menús especiales:

- Menú de aplicaciones: contiene todas las aplicaciones y las herramientas de preferencias. Este menú también incluye el gestor de archivos y el navegador de ayuda.
- Menú Acciones: contiene varios comandos que permiten ejecutar funciones del Entorno del escritorio, como Bloquear la pantalla y Terminar sesión.

Menú

Se puede acceder a todas las funciones del Entorno de escritorio mediante los menús. El Panel de menú contiene menús, de modo que se puede utilizar tanto los menús como los paneles para realizar cualquier

tarea. Los menús de aplicaciones y acciones se utilizan para acceder a la mayoría de aplicaciones estándar, comandos y opciones de configuración.

Ventanas

Se pueden mostrar varias ventanas al mismo tiempo y en cada una de ellas se pueden ejecutar diferentes aplicaciones. El gestor de ventanas proporciona marcos y botones para cada ventana y permite que se ejecuten acciones estándar, como mover, cerrar y redimensionar ventanas.

Áreas de trabajo

El Entorno de escritorio se puede subdividir en diferentes áreas de trabajo. Un área de trabajo consiste en un espacio delimitado del entorno del escritorio en el que se puede trabajar y se puede especificar la cantidad que deben aparecer. Se puede cambiar a un área de trabajo diferente, pero sólo puede mostrarse una cada vez por pantalla.

Gestor de archivos

El administrador de archivos proporciona un punto de acceso integrado a los archivos y aplicaciones. Se puede visualizar el contenido de los archivos mediante una ventana de Nautilus o abrir archivos con la aplicación correcta desde el propio Nautilus. El gestor de archivos permite gestionar los archivos y carpetas.

Preferencias del Entorno de escritorio

Las herramientas de preferencias del escritorio permiten configurar la mayoría de las características del Entorno del escritorio. Cada herramienta controla un aspecto particular del comportamiento del escritorio. Por ejemplo, se puede utilizar una de estas herramientas para seleccionar un tema del Entorno del escritorio. Un tema es un conjunto de valores coordinados que configuran el aspecto visual de la interfaz.

Área de notificaciones

El área de notificación es una característica del Entorno de escritorio que añade un pequeño ícono al panel mientras el programa se está ejecutando, esto permite controlar la aplicación desde la bandeja del sistema y recibir información aun cuando la interfaz no sea visible. (Oracle, 2010)

1.4 Análisis de las distribuciones más populares de GNU/Linux, Windows y Mac.

Debido a la cantidad desconcertante de distribuciones de Linux que existen en la actualidad se realiza un análisis teniendo en cuenta cuales son consideradas como las más usadas por los usuarios de Linux alrededor del mundo (Bodnar, 2011). En la Figura 1 se muestra una gráfica que representa las tendencias de búsqueda en diciembre del 2013 de las distribuciones más populares de los Sistemas Operativos.

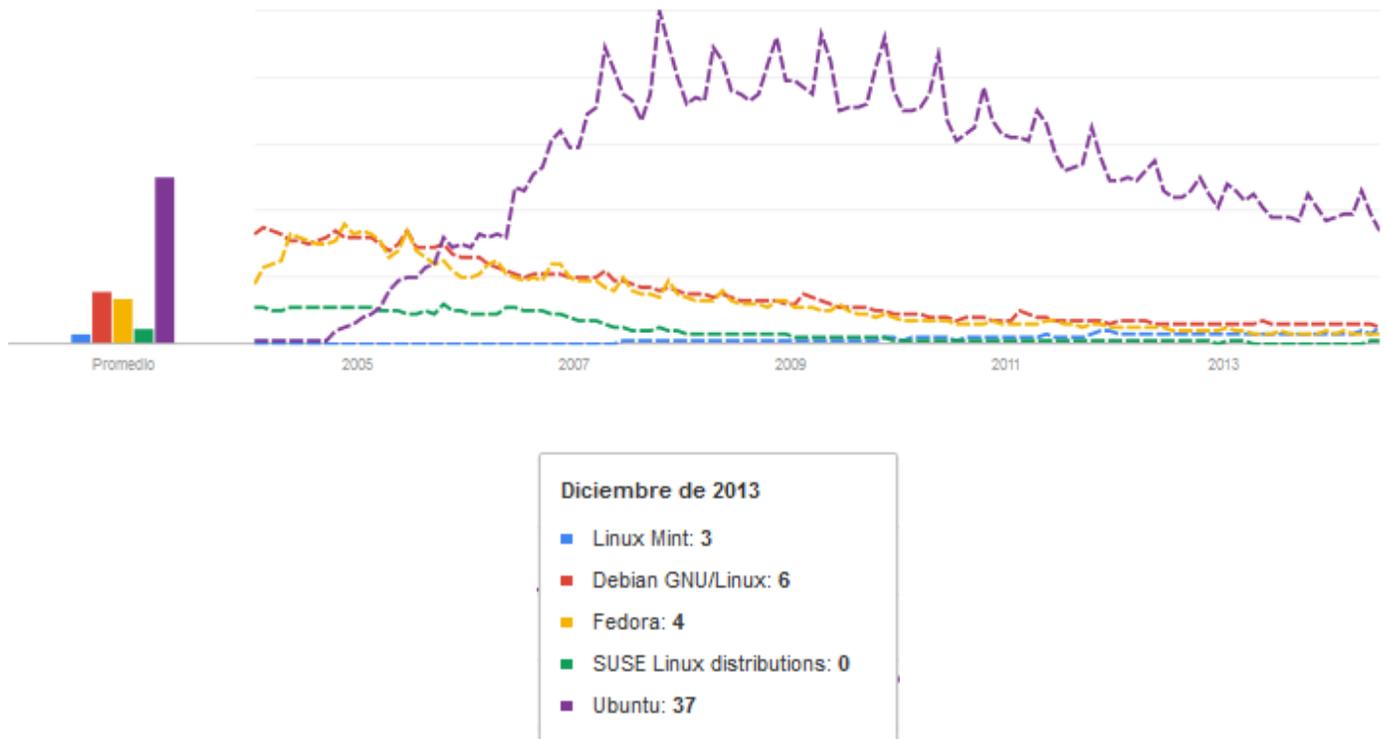


Figura 1. Tendencias de búsquedas.

Números incluidos en el gráfico

Los números del gráfico reflejan la cantidad de búsquedas que se han realizado de un término específico en relación con el número de búsquedas totales realizadas en Google a lo largo del tiempo. No representan cifras de volumen de búsquedas absolutas, porque los datos están normalizados y se presentan en una escala de 0 a 100. Cada punto del gráfico se divide entre el valor del punto más alto o 100. Si no disponemos de datos suficientes, se mostrará el valor 0.

Barras situadas junto al gráfico

Al comparar varios elementos, aparecen barras junto al gráfico. La altura de la barra representa la media de todos los puntos del gráfico de ese término de búsqueda.

Líneas descendentes

Una tendencia descendente significa que ha disminuido su popularidad, no que haya descendido el número de búsquedas absoluto, o total, de ese término. (Google, 2014)

A continuación se brindan un conjunto de datos de interés de las distribuciones más populares de los Sistemas Operativos que son objeto de análisis en la investigación.

Linux Mint

Es una distribución basada en Ubuntu. Fue liberada por primera vez en el 2006 por Clement Lefebvre, un especialista en Tecnología de Información. Desde sus inicios los desarrolladores han ido agregando una variedad de herramientas gráficas "nuevas" para un uso mejorado. Esto incluye a *mintDesktop*, una utilidad para configurar el ambiente del escritorio; *mintMenu*, un menú de estructura nueva y elegante

para una navegación más sencilla; *mintInstall*, un instalador de programas fácil de usar y a *mintUpdate*, un actualizador de software, por mencionar sólo unos pocos de los más prominentes entre otras herramientas y cientos de mejoras adicionales. El proyecto también diseña su propio arte a la par de que su reputación por facilidad de uso ha sido aumentada aún más por la inclusión de controladores multimedia propietarios y puestos detrás de patentes haciendo que frecuentemente estén ausentes en distribuciones más grandes debido a potenciales amenazas legales. No obstante, una de las mejores características de Linux Mint es el hecho de que los desarrolladores escuchan a los usuarios y siempre son rápidos implementando buenas sugerencias. (Bodnar, 2011)

Debian

Fundada por Ian Murdock en agosto de 1993, Debian es una organización formada exclusivamente por voluntarios que se dedican al desarrollo de software libre, es por tanto una distribución no comercial. De hecho, es la distribución no comercial más importante de GNU/Linux, superando en importancia incluso a la mayoría de las comerciales. Esta distribución ha tenido tal repercusión que ha sido la inspiración de más de un centenar de nuevas distribuciones y *liveCD*, entre ellas Ubuntu que también se analiza en esta investigación.

Ha sido innovadora en muchos aspectos, por ejemplo fue la primera en incluir un sistema de gestión de paquetes, conocido como APT, el cual permitía instalar o desinstalar el software con mayor facilidad. También fue la primera distribución que permitía actualizarse sin necesidad de reinstalar el sistema.

OpenSUSE

Tiene sus orígenes en 1992 cuando cuatro usuarios alemanes de Linux lanzaron el proyecto *SUSE*. Cuatro años más tarde, en mayo del 1996 lanzaron la versión 4.2, su primera versión independiente de *Linux-SUSE*, incorporando una nueva distribución al mercado. A finales del 2003 fue comprada por Novell que introdujo cambios importantes, haciendo que el desarrollo de la distribución fuese abierto para todos los usuarios y dando lugar al nacimiento de *OpenSUSE*, cuya versión 10.0 lanzada en octubre del 2005 fue la primera totalmente libre.

Históricamente esta distribución ha destacado por sus excelentes y amigables Entornos de escritorio, su magnífico sistema de administración de aplicaciones conocido como *YAST* o por la frecuencia de sus lanzamientos, apenas pasan unos meses entre nuevas versiones y cualquier error que aparezca en el sistema es corregido rápidamente.

Fedora

A pesar de que Fedora fue formalmente develada sólo en septiembre del 2004, sus orígenes datan desde 1995 cuando fue lanzada por dos visionarios de Linux Bob Young y Marc Ewing bajo el nombre de *Red Hat Linux*. El primer producto de la empresa, *Red Hat Linux 1.0*, fue liberado en el mismo año y rápidamente fue seguido por varias actualizaciones para corregir errores. (Bodnar, 2011)

Se puede decir por tanto que Fedora es la evolución de la antigua *Red Hat Linux*. Su frecuencia de actualizaciones es semestral, cuenta con versión *liveCD* y con versión para descarga de imágenes en tamaño NET, CD o DVD. Tiene soporte para i386 y x86_64, da soporte a los escritorios GNOME, KDE, LXDE y XFCE.

Ubuntu

Creada en 2004, es la más reciente de todas las distribuciones de la comparativa. Fundada por Mark Shuttleworth, multimillonario y antiguo trabajador de Debian estaba claro que su distribución tenía que estar basada en Debian, en concreto en Sid, una versión no estable de Debian. A pesar de su juventud es en la actualidad la distribución más popular de todas, algunas estadísticas sugieren que Ubuntu aglutina el 50% del mercado en lo que a distribuciones se refiere. Uno de los motivos de su éxito, es que al igual que Mandriva se enfoca en la facilidad de uso y en mejorar la experiencia del usuario.

Como curiosidades se destacan que ofrece asistencia a los usuarios para migrar sus equipos desde Windows o que envían los CD gratuitamente a aquellos usuarios que lo soliciten. Por otro lado, dispone de *liveCD* estable, tiene una frecuencia de nuevos lanzamientos semestral, oficialmente solo soporta arquitecturas x86, tanto en 32 como 64 bits y utiliza de manera predeterminada un escritorio llamado *Unity* basado en GNOME.

Windows XP

Fue el gran cambio que tuvo *Microsoft* con *Windows*, siendo esta una de sus versiones más populares a lo largo de toda su historia, con increíbles mejoras gráficas, un gran incremento de velocidad y agilidad. También incorpora su versión de 64 bits, y logra hacer una explosión en el mercado de sistemas operativos. Incluso hoy en día muchas empresas utilizan *Windows XP*, habiendo nuevas versiones como *Windows 7* disponibles en el mercado.

Se incorporaron funciones como la encriptación de archivos del sistema, el soporte para redes *Wi-Fi*, y la asistencia remota. También existieron dos versiones de *Windows XP*, entre las cuales estaban el *Windows XP Home* y *Windows XP Professional*. (Aguilar, 2011)

Apple Mac OS X

Es el Sistema Operativo creado por *Apple Inc.* que viene instalado en todos los computadores *Mac*. Todas las versiones recientes son conocidas como *Mac OS X* y los nombres específicos de cada una de ellas son *Mountain Lion* (2012), *Lion* (2011) y *Snow Leopard* (2009). *Apple* también ofrece una versión llamada *Mac OS X Server* que está diseñado para ejecutarse en los servidores.

Los usuarios de *Mac OS X* representan el 7,5% del mercado de Sistemas Operativos a enero de 2013 (mucho menor que el porcentaje de usuarios de *Windows*). Una de las razones es que los ordenadores *Apple* tienden a ser más caros. Sin embargo, muchas personas los están empezando a preferir por su confiabilidad, diseño y desempeño. (Community Foundation, 2013)

A continuación se realiza una comparación en cuanto a los elementos fundamentales de un Entorno de escritorio de las distribuciones más populares de los Sistemas Operativos.

	LinuxMint 16	Debian 7.3.0	OpenSuse 13.1	Fedora 19	Ubuntu 13.10	Windows XP	Mac OS
Paneles	Cuenta con un Panel de menú en la parte inferior izquierda del escritorio que contiene un menú de aplicaciones y un menú de acciones.	Panel superior o de menú donde muestra las principales aplicaciones y el diálogo de cierre de sesión. Panel inferior muestra accesos directos a home, emulador de terminal y escritorio.	Panel de menú en la parte superior del escritorio.	Panel de menú en la parte superior del escritorio.	Panel de menú en la parte superior del escritorio.	Cuenta con una barra de tareas en la parte inferior del escritorio que contiene un menú de aplicaciones, un panel de control y un menú de acciones.	Panel superior que cuenta con el menú de acciones, archivo, edición y ventana.
Menú	Menú Inicio incluye el menú de acciones, favoritas y de aplicaciones. Menú de ventana (clic derecho en la barra de título de la ventana). Al dar clic derecho en el escritorio muestra un menú con opciones como carpeta nueva,	Menú de aplicaciones. Menú de ventana (clic derecho en la barra de título de la ventana). Menú/home/Debian donde se muestran las carpetas del sistema como descargas, documentos, música.	Menú Actividades. Submenú Mostrar aplicaciones. Menú de ventana (clic derecho en la barra de título de la ventana). Menú de acción se encuentra en la parte derecha del panel de menú.	Menú Actividades. Submenú Mostrar aplicaciones. Menú de ventana (clic derecho en la barra de título de la ventana). Menú de acción se encuentra en la parte derecha del panel de menú.	Menú de aplicaciones Menú de ventana (clic derecho en la barra de título de la ventana). Menú de acción se encuentra en la parte derecha del panel superior. Al dar clic derecho en el escritorio muestra un menú	Menú Inicio incluye el menú de acciones y de aplicaciones (se puede personalizar). Submenú Todo los programas. Menú de ventana (clic derecho en cada una de las barras de la ventana).	Menú de aplicaciones en la parte inferior del escritorio, lo oculta automáticamente. Menú de acciones en la parte izquierda del panel superior.

	<p>cambiar fondo de escritorio, entre otras.</p> <p>Al dar clic derecho en las aplicaciones del menú de Aplicaciones muestra un submenú con opciones como adicionar al escritorio, a favoritas y al panel.</p> <p>Al dar clic derecho en las carpetas del escritorio muestra un menú con opciones como abrir, propiedades, entre otras.</p> <p>Al dar clic derecho en el panel se accede a un menú para configurarlo.</p>	<p>Al dar clic derecho en las carpetas del escritorio muestra un menú con opciones como abrir, propiedades, así como un submenú aplicaciones.</p> <p>Al dar clic derecho en el escritorio muestra un menú con opciones como abrir en ventana nueva, configuración del escritorio, así como un submenú aplicaciones.</p>			<p>con opciones como carpeta nueva, cambiar fondo de escritorio, entre otras.</p> <p>Al dar clic derecho en las aplicaciones del lanzador muestra un menú con opciones como abrir una ventana nueva, no mantener en el lanzador y otras.</p>		
Ventanas	Posee la opción de minimizar, maximizar y cerrar en la parte	Posee la opción de minimizar, maximizar y cerrar en la parte derecha	No posee la opción de minimizar y maximizar visibles	No posee la opción de minimizar y maximizar visibles	Posee la opción de minimizar, maximizar y cerrar en la parte derecha	Posee la opción de minimizar, maximizar y cerrar en la parte	Posee la opción de minimizar, maximizar y cerrar en la parte

	<p>derecha de la ventana. Cuenta con la barra de título y al dar clic derecho encima de ella muestra un menú que contiene las opciones de minimizar y maximizar. Cuenta con la barra de herramientas.</p>	<p>de la ventana. Cuenta con la barra de título y al dar clic derecho encima de ella muestra un menú que contiene las opciones de minimizar y maximizar. Cuenta con la barra de herramientas.</p>	<p>en la ventana. Cuenta con la barra de título y al dar clic derecho encima de ella muestra un menú que contiene las opciones de minimizar y maximizar. Muestra el botón de cerrar en la parte derecha de la barra de título. Permite personalizar la barra de herramientas de las ventanas al dar clic derecho encima de ella.</p>	<p>en la ventana. Cuenta con la barra de título y al dar clic derecho encima de ella muestra un menú que contiene las opciones de minimizar y maximizar. Muestra el botón de cerrar en la parte derecha de la barra de título. Permite personalizar la barra de herramientas de las ventanas al dar clic derecho encima de ella.</p>	<p>de la ventana, omite la barra de título por lo que las opciones de minimizar, maximizar y cerrar se encuentran en la barra de menú. Permite personalizar la barra de herramientas de las ventanas al dar clic derecho encima de ella.</p>	<p>derecha de la ventana. Permite personalizar la barra de herramientas de las ventanas. Se accede a las opciones de minimizar, maximizar y cerrar al dar clic derecho encima de la barra de título.</p>	<p>izquierda de la ventana. Contiene la barra de título, la barra de herramientas se incluye el panel superior para cada ventana.</p>
Área de trabajo	<p>Cuenta con una sola área de trabajo.</p>	<p>El sistema brinda la facilidad de trabajar con cuatro áreas de trabajo por defecto a las que se acceden por la derecha del panel superior. Pero</p>	<p>Contiene el aplique conmutador de áreas de trabajo en la parte izquierda del escritorio. (No</p>	<p>Contiene el aplique conmutador de áreas de trabajo en la parte izquierda del escritorio. (No</p>	<p>Una vez que se activan las áreas de trabajo en (Apariencia/Comportamiento) el sistema brinda la facilidad de trabajar con cuatro</p>	<p>Cuenta con una sola área de trabajo.</p>	<p>Cuenta con tres áreas de trabajo por defecto.</p>

		permite trabajar hasta con 100 áreas más.	están restringidas las áreas de trabajo con las que se desea interactuar)	están restringidas las áreas de trabajo con las que se desea interactuar)	áreas de escritorio diferentes a las que se acceden por el menú de aplicaciones.		
Gestor de archivos	Al Sistema de archivos se accede por un ícono en el escritorio, por el Panel de menú y por el menú Favoritas.	Al Sistema de archivos se accede por un ícono en el escritorio.	El gestor de archivo se encuentra en el menú actividades/archivo.	El gestor de archivo se encuentra en el menú actividades/archivo.	El gestor de archivo se encuentra en el menú actividades/archivo.	El gestor de archivo se encuentra en el menú Inicio/Mi PC.	El gestor de archivos (Documentos) se encuentra a la derecha del menú de aplicaciones en la parte inferior del escritorio.
Preferencias del entorno	Se accede a la configuración del sistema por el menú de aplicaciones/configuración.	Se accede a la configuración del sistema por el menú de aplicaciones/configuración.	Se accede a la configuración del sistema por el menú de acciones.	Se accede a la configuración del sistema por el menú de acciones.	Se accede a la configuración del sistema por el menú de acciones y por el menú aplicaciones.	Al dar clic derecho/propiedad es encima de cualquier componente del escritorio este se puede configurar. Se accede a todas las configuraciones del entorno por el menú Inicio/Panel de control.	Se accede a las preferencias del sistema por el menú de acciones.
Área de	Se encuentra en la	Se encuentra en la	Se encuentra en la	Se encuentra en la	Se encuentra en la	Se encuentra en la	Se encuentra en

notificaciones	derecha del Panel de menú.	parte superior derecha del escritorio por defecto, pero puede personalizarse colocándose en cualquiera de las 4 esquinas del escritorio, las notificaciones aparecen en el área de trabajo en forma de diálogo.	parte superior derecha del escritorio.	parte superior derecha del escritorio.	parte superior derecha del escritorio.	parte inferior derecha del escritorio a continuación de la barra de tareas y acepta ser personalizada.	la parte superior derecha del escritorio.
-----------------------	----------------------------	---	--	--	--	--	---

Tabla 1. Comparación entre distribuciones.

Luego de realizar el análisis de la AI de Entornos de escritorio de las distribuciones más populares de los Sistemas Operativos se considera, que para contribuir a la calidad deseada de Nova Escritorio 5.0, se debe mejorar la Usabilidad, la Organización y la Ayuda del sistema. Para ello se debe proveer a los usuarios de una serie de requisitos que han provocado que las distribuciones estudiadas hayan sido las más utilizadas hasta la actualidad, los cuales se mencionan a continuación:

Usabilidad

- Agregar los botones maximizar y minimizar en la barra de título, dado que este es un elemento que tienen en cuenta la mayoría de las distribuciones investigadas. Además facilita en gran medida la interacción de los usuarios con el sistema cuando están trabajando con más de una aplicación a la vez.
- Disminuir el tamaño de la iconografía y la tipografía para que el Entorno de escritorio sea más amigable y agradable para los usuarios.
- Cambiar el ícono de Accesibilidad en el Panel de menú para que muestre de una forma más clara la funcionalidad que representa y para que el usuario sepa a donde dirigirse para configurar su PC (del inglés *Personal Computer*, traducido al español como Computadora Personal) de acuerdo a la discapacidad que posee.

- Cambiar el ícono de Aplicaciones que se muestra en el menú Favoritos por uno que sea más intuitivo.
- Cambiar el botón del menú de aplicaciones “Actividades” por el logo de Nova, dado que así es como lo manejan la mayoría de las distribuciones estudiadas para representar su menú de aplicaciones y para darle más identidad al sistema.
- Adicionar nueva forma para agregar y eliminar las aplicaciones a Favoritos para que los usuarios tengan la posibilidad de elegir como hacerlo y para que el menú tenga una mayor facilidad de uso.

Organización

- El área de notificaciones debe permitir a los usuarios elegir las aplicaciones de las que ellos deseen recibir notificaciones, así como los íconos que se deben mostrar en ella.
- El menú debe permitir cambiar las aplicaciones de lugar para que los usuarios lo puedan organizar a su gusto y consideración.
- Para que la sección de Todas las aplicaciones se vea más organizada y sea más fácil para los usuarios encontrar las aplicaciones que van a utilizar, estas se deben mostrar también agrupadas por categorías.

Ayuda

- Crear una ayuda que se muestre al dar clic derecho encima de las aplicaciones que trae por defecto el sistema, para que los usuarios tengan una vía más rápida y cómoda de saber para que se utiliza la misma.
- Crear una ayuda que se muestre en el primer inicio del sistema, que le explique al usuario los elementos básicos para interactuar con el mismo.
- Realizar un paseo por Nova en forma de video o multimedia donde se muestren a los usuarios las principales configuraciones, atajos de teclado y demás elementos del sistema.

1.5 Evolución de la AI de Nova Escritorio 3.0 a Nova Escritorio 4.0.

En este punto se realiza un análisis de la evolución de la AI del Entorno de escritorio de Nova Escritorio 3.0 a Nova Escritorio 4.0.

Elementos	Nova 3.0	Nova 4.0
Panel	Panel de menú en la parte inferior del escritorio.	Panel de menú en la parte superior del escritorio. El panel inferior contiene el área de notificaciones.
Menú	El menú de aplicaciones contiene en la parte izquierda el menú de aplicaciones Favoritas (las aplicaciones se pueden cambiar de lugar en el orden de preferencia), al dar clic en el botón Más	Menú Actividades (permite mover las aplicaciones favoritas de lugar excepto aplicaciones y si se arrastra alguna hacia Aplicaciones muestra la papelera de reciclaje para eliminarlas).

	<p>Aplicaciones muestra una nueva ventana con el resto de las aplicaciones agrupadas por categorías y en la parte derecha el menú de acciones.</p> <p>Al dar clic derecho en el escritorio muestra un menú con opciones como crear una carpeta, crear un lanzador, cambiar el fondo del escritorio, etc.</p> <p>Al dar clic derecho encima de las aplicaciones muestra un menú con Iniciar aplicación, añadir o eliminar de favoritos y añadir o eliminar de programas de inicio.</p> <p>Menú de ventana (clic derecho en el borde de la ventana).</p>	<p>Submenú aplicaciones.</p> <p>Menú de ventana (clic derecho en el borde de la ventana).</p> <p>Menú de Acción (se encuentra en la parte derecha del panel de menú).</p> <p>Al dar clic derecho en el escritorio muestra un menú con opciones como crear una carpeta, cambiar el fondo del escritorio, entre otras.</p> <p>Al dar clic derecho encima de las aplicaciones se muestra un menú, que cambia, si la aplicación está cerrada muestra Ventana nueva, añadir o quitar de los favoritos y si está abierta incluye en el menú el nombre de la aplicación.</p>
Ventanas	<p>Posee la opción de minimizar, maximizar y cerrar en la parte derecha de la ventana.</p> <p>Permite personalizar la barra de herramientas de las ventanas al dar clic derecho encima de ella.</p>	<p>No poseen la opción minimizar y maximizar (se puede configurar).</p> <p>Cuenta con la barra de título y al dar clic derecho encima de ella muestra un menú que contiene las opciones de minimizar y maximizar.</p> <p>Muestra el botón de cerrar en la parte derecha de la barra de título.</p>
Área de trabajo	<p>Una vez que se activaban las áreas de trabajo en (Clic derecho en el panel/Añadir al panel/Selector de áreas de trabajo) el sistema da la facilidad de trabajar con cuatro áreas de trabajo diferentes a las que se acceden por el Panel de menú.</p>	<p>Contiene el aplique conmutador de áreas de trabajo en la parte izquierda del escritorio. (No están restringidas las áreas de trabajo con las que se desea interactuar).</p>
Gestor de archivos	<p>El gestor de archivo se encuentra en el menú Nova/Administrador de archivos.</p>	<p>El gestor de archivo se encuentra en el menú Actividades/archivos.</p>
Preferencias del entorno	<p>Se accede al centro de control por el menú de acciones.</p>	<p>Se accede a la configuración del sistema por el menú de acciones</p>
Área de notificaciones	<p>Se encuentra en la parte inferior derecha del escritorio a continuación de la barra de tareas y acepta ser personalizada.</p>	<p>La muestra en el panel inferior.</p>
Documento de la Arquitectura de la Información	<p>No existe.</p>	<p>No existe.</p>

Tabla 2. Evolución de la AI del Entorno de escritorio de Nova Escritorio.

Después de estudiar la evolución que ha tenido la distribución Nova Escritorio, es válido destacar el crecimiento que se observa de una versión a otra. Aunque se han corregido muchas deficiencias persisten otras que atentan contra la completa aceptación de la distribución por parte de los usuarios finales. Algunos de estos problemas se describen a continuación:

- Al instalar el sistema en una computadora portátil el botón de apagado que debe mostrarse en el Panel de menú es sustituido por el ícono de la batería, por lo que el usuario a simple vista no sabe hacia dónde dirigirse para apagar su PC.
- En el menú no se muestra el nombre completo de las aplicaciones, por lo que si el usuario no reconoce su ícono tampoco será capaz de identificarla por el nombre.
- El método de la red cableada no está implementado, por lo que el usuario no puede saber el estado en el que esta se encuentra cuando tiene la PC conectada por cable. Además no se muestra en el menú de acciones sus opciones de configuración.
- En el menú de acciones aparecen las opciones de apagado del sistema, por lo que no se deben tratar en el submenú de la red los términos Encender y Apagar porque esto puede causar confusión en los usuarios.
- Cuando existen más de 10 tareas programadas en el menú del calendario la ventana que lo contiene comienza a agrandar su tamaño y así sucesivamente hasta que alcanza la altura del escritorio, en lugar de mostrar una lista con un *scroll* o algo similar para que el menú mantenga su tamaño original.
- La guía de usuario que aparece al presionar F1 muestra las especificaciones de Ubuntu.

Además de los problemas que se mencionan, la versión actual del sistema presenta elementos que favorecieron a su aceptación. Uno de los que más impacto causó fue el Alt+Tab por ello se decide mantenerlo en la próxima versión.

1.6 Tecnologías para desarrollar soluciones de la AI.

1.6.1 Metodología de desarrollo

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.

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. (Fuentes Rodríguez, 2012)

Se selecciona la metodología SXP para el desarrollo del siguiente trabajo porque los elementos que la caracterizan se adaptan 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.

1.6.2 Lenguajes de Programación

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. Soporta gran parte de la sintaxis de programación estructurada a partir de C (por ejemplo, las declaraciones *if*, los bucles *while* y las declaraciones *switch*). 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. (Mozilla Developer Network, 2013)

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 utilizado ampliamente en otros software que no están vinculados a 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.

1.6.3 Bibliotecas fundamentales a utilizar

Gobject Introspection

Las principales bibliotecas 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*, y *JavaScript*. 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 esto surge la biblioteca *Gobject Introspection*, la cual 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 biblioteca *Gobject*, usando anotaciones dentro de los comentarios que son interpretados luego. También crea una base de datos llamada *Gobject Introspection Repository (gir)* en la que se guarda 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 biblioteca. Al ser compiladas estas bibliotecas, un escáner revisa el código fuente en busca de las anotaciones en los comentarios y crea un fichero *gir* que es compilado y situado en el repositorio (*/usr/lib/girepository-1.0*), donde podrá ser utilizado por otras. (Fuentes Rodríguez, 2012)

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. (Fuentes Rodríguez, 2012)

GLib

La biblioteca *GLib* es una de las más importantes que existen en GNOME. Esta biblioteca es, junto a la biblioteca *GTK+*, el pilar sobre el que se sustentan todas las aplicaciones. Dentro de *GLib* está implementada una serie de tipos de datos que nos hace más fácil, si cabe, el tratamiento de los datos y además tiene la propiedad de mejorar la portabilidad de nuestros programas. Los tipos que se usan en *GLib* no son muy diferentes de los que usamos en el estándar de C. (The GNOME Foundation, 2002)

GLib es gratuita y licenciada bajo *LGPL (Lesser GPL)*, lo cual significa en la práctica que puede ser utilizada tanto en programas de código abierto como cerrado. *GLib* no provee soporte gráfico.

Frecuentemente suele emparejarse con la biblioteca gráfica *GTK+* ya que la misma utiliza *GLib* internamente como soporte base. Sin embargo *GLib* puede ser utilizado para aplicaciones embebidas o servidor sin dependencia alguna del sistema gráfico. (Wikilibros, 2012)

GTK

GTK+ es la biblioteca gráfica (*toolkit*) sobre la que se sustenta toda la interfaz gráfica de GNOME y sigue el modelo de programación orientado a objetos. Es una biblioteca que contiene todo lo necesario para el desarrollo de interfaces gráficas, permitiendo la posibilidad de crear todo tipo de *widgets*, desde los más básicos, como botones, etiquetas, cajas de texto, hasta cosas más complejas como selectores de ficheros, colores, fuentes y cajas de texto multilínea con soporte para todo tipo de efectos sobre el texto.

Forman parte de *GTK+*:

- *GDK* implementa el nivel más bajo de la arquitectura, forma una capa sobre la implementación gráfica real y se reescribe para cada plataforma.
- *gdk-pixbuf* permite el tratamiento de imágenes gráficas. (Gonzalez, 2011)

1.6.4 Herramientas

Geany 1.23

Es un editor de texto ligero basado en *Scintilla* con características básicas de entorno de desarrollo integrado (IDE). Está disponible para distintos Sistemas Operativos, como GNU/Linux, *Mac OS X*, *BSD*, *Solaris* y *Microsoft Windows*. Es distribuido como software libre bajo la Licencia Pública General de GNU. Tiene soporte para muchos lenguajes de programación, como *C*, *C++*, *Java*, *JavaScript*, *PHP*, *HTML*, *CSS*, *Python*, *Perl*, *Ruby*, *Fortran*, *Pascal*, *Haskell* entre tantos otros. Algunas de sus características más destacadas son:

- Autocompletado
- Soporte multidocumento
- Soporte de proyectos
- Coloreado de sintaxis
- Emulador de terminal incrustado

Algunas de las utilidades más conocidas son:

- Compatible con la mayoría de lenguajes
- Varios paneles para acceder mejor a los datos
- Herramientas para compilar
- Buscador integrado

Lo más sorprendente quizás sea que este sencillo programa permite:

- Compilar y ejecutar directamente desde el entorno (en todos los lenguajes orientados a esta labor). Aunque parezca increíble, es capaz de llamar al compilador y luego ejecutar el programa compilado directamente a través de una consola que se te integra en el programa.

- Descomponer y representar las clases y estructuras del código. Geany interpreta el código y en la barra izquierda representa las estructuras y clases que aparecen en él.
- Ampliar funcionalidades mediante complementos. (Tröger, 2008)

Conclusiones Parciales

- El análisis crítico de las teorías asociadas a la investigación arrojó como resultado que existe una correspondencia entre las principales potencialidades que tienen los Entornos de escritorio más populares y las deficiencias que tienen las versiones de Nova Escritorio.
- El estudio de las tecnologías adecuadas para el desarrollo de Entornos de escritorios determinó que para realizar la investigación se debe utilizar la Metodología de desarrollo SXP, el IDE de programación o editor de texto *Geany 1.23*, el Lenguaje de programación *JavaScript* y como herramienta de modelado *Visual Paradigm*.
- No existe un documento formal que represente la AI del Entorno de escritorio de ninguna de las versiones de Nova Escritorio.
- El análisis de las distribuciones permitió determinar los requisitos que debe cumplir la AI del Entorno de escritorio de Nova Escritorio 5.0.

Capítulo 2: Propuesta de cambios para la AI del Entorno de escritorio de Nova Escritorio 5.0.

Introducción

En el proceso de desarrollo de software uno de los errores que se comete en muchas ocasiones es la no documentación de los cambios que se realizan. En el presente capítulo se pretende organizar y documentar cada una de las soluciones que se desarrollen para evitar esta mala práctica, además de brindar el procedimiento a realizar y los componentes a modificar para facilitar el trabajo a las personas que en un futuro vayan a reutilizar o modificar dichas soluciones.

2.1 Descripción de la solución propuesta para la AI del Entorno de escritorio de Nova Escritorio 5.0.

Para cumplir con el objetivo planteado se propone desarrollar cambios para la AI del Entorno de escritorio de Nova Escritorio 5.0 para contribuir a la calidad de la distribución. Estos cambios tienen base en la decisión del proyecto de mantenerse con la versión más actualizada de Gnome para su Entorno de escritorio. En el momento de la investigación Gnome se encuentra en su versión 3.10. La presentación de la AI se realiza mediante tres modelos fundamentales, Diagramas de organización, de funcionamiento y de presentación; además de una explicación detallada de los elementos que componen el Entorno de escritorio. Todos estos elementos están contenidos en la plantilla Arquitectura de la Información del Entorno de escritorio de Nova Escritorio 5.0. En la Figura 2 se puede apreciar que las modificaciones se realizarán en el componente Panel (Gnome-Shell) del Entorno de escritorio.

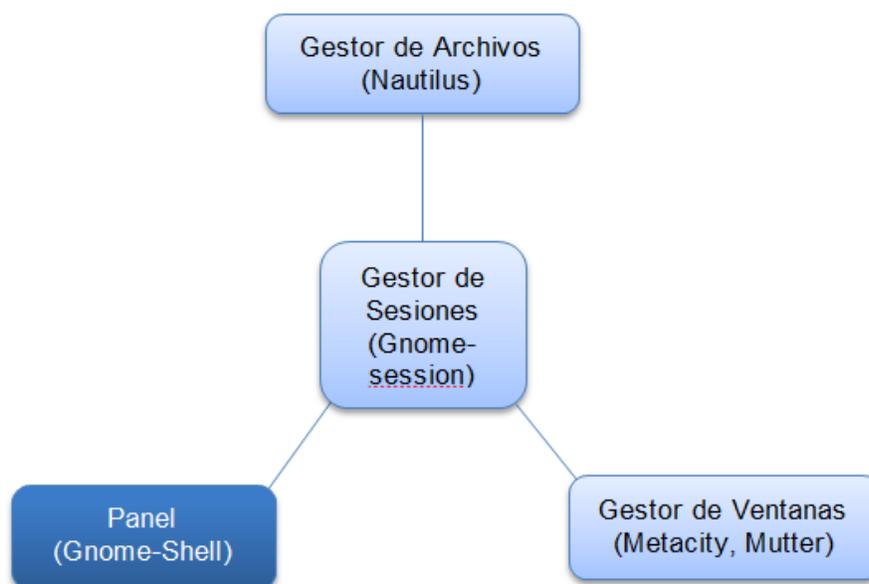


Figura 2. Entorno de escritorio de Nova Escritorio 5.0.(Fuentes Rodríguez, 2012)

En la Figura 3 se muestra el diagrama de funcionamiento del Entorno de escritorio Gnome. Los elementos resaltados en azul son los que se modifican para la AI del Entorno de escritorio de Nova Escritorio

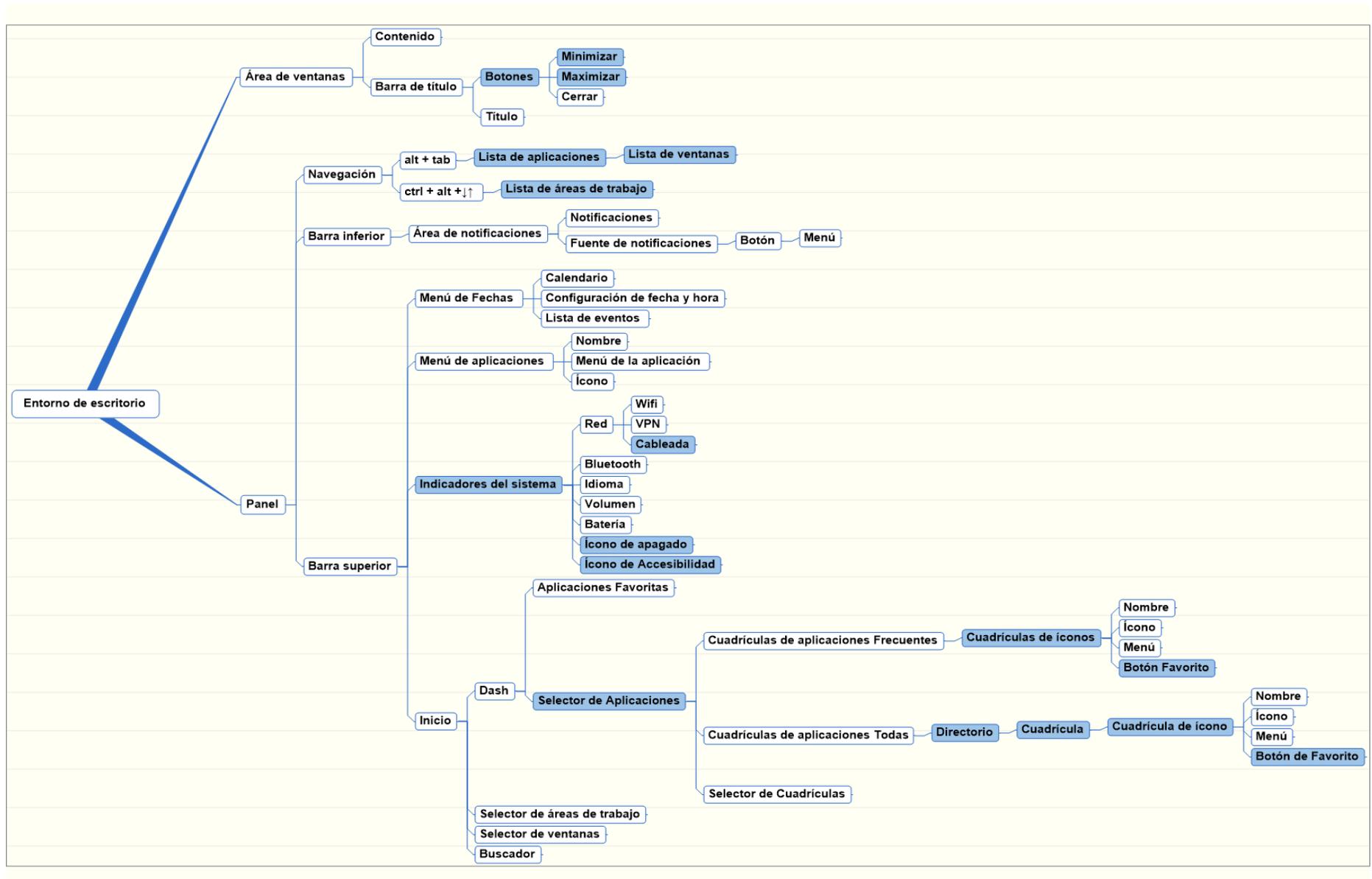


Figura 3. Diagrama de funcionamiento del Entorno de escritorio Gnome.

2.2 Requisitos funcionales.

Prioridad Muy Alta

1. Soporte para el método de la red cableada.
2. Mostrar la sección de todas las aplicaciones en el menú agrupadas por categorías.
3. Adicionar el botón de apagado del sistema al Panel de menú para las computadoras portátiles.
4. Agregar los botones maximizar y minimizar en la barra de título.
5. Modificar el botón Actividades del sistema.

Prioridad Alta

6. Adicionar nueva forma para agregar y eliminar las aplicaciones a Favoritos.
7. Modificar el ícono Accesibilidad del sistema.
8. Modificar el Alt+Tab.

Prioridad Media

9. Mostrar el nombre completo de las aplicaciones.
10. Crear ayuda para el primer inicio del sistema.
11. Modificar la ayuda del sistema.
12. Realizar un paseo por el sistema.

Prioridad Baja

13. Modificar el área de notificaciones.
14. Crear una ayuda para las aplicaciones.
15. Disminuir el tamaño de la iconografía y la tipografía
16. Permitir al usuario cambiar el orden de las aplicaciones en el menú.
17. Modificar el menú del calendario.
18. Modificar el ícono de Aplicaciones.

2.3 Historias de usuarios (HU)

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Agregar los botones maximizar y minimizar en la barra de título.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Katileidys Zerquera Canales Iván Romay Aragón	Iteración Asignada: 1
Prioridad en Negocio: Muy alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1/5
Descripción: Se incluirán los botones maximizar y minimizar en la barra de título de las ventanas.	
Observaciones:	

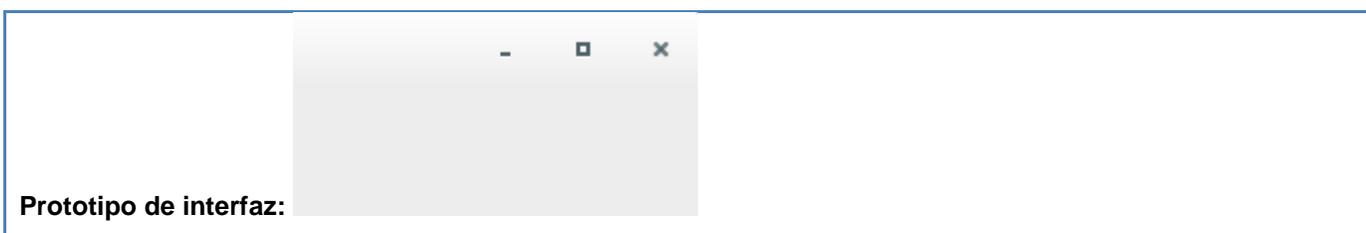


Tabla 3. HU_1 agregar los botones maximizar y minimizar en la barra de título.

Para la realización de la HU_1 se define la tarea de ingeniería 1, para realizar un estudio sobre dónde y cómo se crean los botones de la barra de título, la 2 para llevar a cabo la modificación al método donde se crean los botones de la misma y la 3 para documentar el cambio así como los artefactos, diagramas y prototipos necesarios.

Historia de Usuario	
Número: HU_2	Nombre Historia de Usuario: Mostrar la sección de todas las aplicaciones en el menú agrupadas por categorías.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Katileidys Zerquera Canales Iván Romay Aragón	Iteración Asignada: 1
Prioridad en Negocio: Muy alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1 ^{1/5}
Descripción: El menú muestra Todas las aplicaciones agrupadas en categorías dando la opción de escoger las aplicaciones que se van a mostrar en Favoritos.	
Observaciones:	
Prototipo de interfaz:	

Tabla 4. HU_2 mostrar la sección de todas las aplicaciones en el menú agrupadas por categorías.

Para la realización de la HU_2 se define la tarea de ingeniería 4, para realizar un estudio de los principales elementos que presentan los menús de aplicaciones de diferentes sistemas operativos, la 5

para llevar a cabo la programación de los cambios que se proponen para el menú de aplicaciones y la 6 para documentar el cambio así como los artefactos, diagramas y prototipos necesarios.

Historia de Usuario	
Número: HU_3	Nombre Historia de Usuario: Modificar el botón Actividades del sistema.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Katileidys Zerquera Canales Iván Romay Aragón	Iteración Asignada: 1
Prioridad en Negocio: Muy alta	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 4/5
Descripción: Cambiar el botón Actividades por el logo de Nova, es decir, la estrella.	
Observaciones:	
Prototipo de interfaz:	

Tabla 5. HU_3 Modificar el botón Actividades del sistema.

Para la realización de la HU_3 se define la tarea de ingeniería 7, para realizar un estudio en los diferentes Sistemas Operativos sobre cómo manejan el botón del menú de las aplicaciones, la 8 para modificar el método que muestra el botón Actividades, para que en lugar de este se muestre la estrella de Nova y la 9 para documentar el cambio así como los artefactos, diagramas y prototipos necesarios.

Historia de Usuario	
Número: HU_4	Nombre Historia de Usuario: Adicionar el botón de apagado del sistema al Panel de menú para las computadoras portátiles.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Katileidys Zerquera Canales Iván Romay Aragón	Iteración Asignada: 1
Prioridad en Negocio: Muy alta	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 3/5
Descripción: Si el dispositivo contiene batería se muestra el ícono de apagado y de la batería respectivamente, sino contiene batería se mantiene el ícono de apagado del sistema y se oculta el ícono de la batería.	

Observaciones:	
Prototipo de interfaz:	

Tabla 6. HU_4 adicionar el botón de apagado del sistema al Panel de menú para las computadoras portátiles.

Para la realización de la HU_4 se define la tarea de ingeniería 10, para adicionar el botón de apagado en el Panel de menú en las computadoras portátiles y la 11 para documentar el cambio así como los artefactos, diagramas y prototipos necesarios.

Historia de Usuario

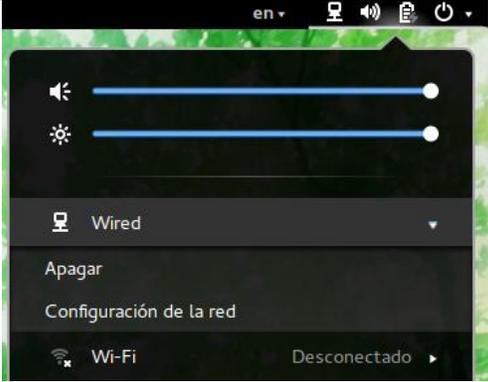
Número: HU_5	Nombre Historia de Usuario: Soporte para el método de la red cableada.		
Modificación de Historia de Usuario Número: Ninguna			
Usuario: Katileidys Zerquera Canales Iván Romay Aragón	Iteración Asignada: 1		
Prioridad en Negocio: Muy alta	Puntos Estimados: 1		
Riesgo en Desarrollo: Alta	Puntos Reales: 1 ^{4/5}		
Descripción: Mostrar el ícono de la red cableada cuando se establezca este tipo de conexión, así como las opciones de configuración en la sección de red del menú de acciones.			
Observaciones:			
Prototipo de interfaz:			

Tabla 7. HU_5 soporte para el método de la red cableada.

Para la realización de la HU_5 se define la tarea de ingeniería 12, para realizar un estudio sobre el comportamiento de la red cableada en versiones anteriores de Gnome, la 13 para realizar la

implementación del método de la red cableada a partir de los resultados obtenidos del estudio realizado y la 14 para documentar el cambio así como los artefactos, diagramas y prototipos necesarios.

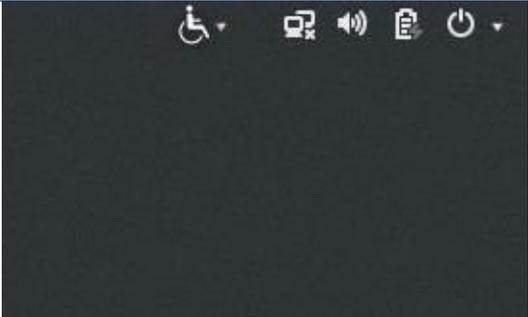
Historia de Usuario	
Número: HU_6	Nombre Historia de Usuario: Modificar el ícono de Accesibilidad.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Katileidys Zerquera Canales Iván Romay Aragón	Iteración Asignada: 1
Prioridad en Negocio: Muy alta	Puntos Estimados: 3/5
Riesgo en Desarrollo: Alta	Puntos Reales: 2/5
Descripción: Modificar el ícono de Accesibilidad que se muestra en el Panel de menú.	
Observaciones:	
Prototipo de interfaz:	

Tabla 8. HU_6 modificar el ícono de Accesibilidad.

Para la realización de la HU_6 se define la tarea de ingeniería 15, para cambiar el ícono de Accesibilidad actual del Panel de menú por otro que sea más intuitivo, la 16 para documentar el cambio así como los artefactos, diagramas y prototipos necesarios.

Historia de Usuario	
Número: HU_7	Nombre Historia de Usuario: Adicionar nueva forma para agregar y eliminar las aplicaciones a Favoritos.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Katileidys Zerquera Canales Iván Romay Aragón	Iteración Asignada: 1
Prioridad en Negocio: Muy alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alta	Puntos Reales: 1 ^{1/5}
Descripción: Se muestra una estrella en la parte superior derecha del ícono de las aplicaciones que permitirá adicionar y eliminar las mismas a Favoritos. (Si la aplicación no es favorita la estrella que se muestra es transparente y en caso contrario blanca)	
Observaciones:	

Prototipo de interfaz:



Tabla 9. HU_7 adicionar nueva forma para agregar y eliminar las aplicaciones a Favoritos.

Para la realización de la HU_7 se define la tarea de ingeniería 17, para realizar un estudio en los Sistemas Operativos sobre los diferentes métodos que existen para agregar y eliminar las aplicaciones a Favoritos, la 18 para agregar un método que debe mostrar una estrella en la parte superior derecha del ícono de las aplicaciones del menú que al dar clic encima de ella la aplicación se agregue a Favoritos si no pertenece a este o se elimine si pertenece y la 19 para documentar el cambio así como los artefactos, diagramas y prototipos necesarios.

Historia de Usuario	
Número: HU_8	Nombre Historia de Usuario: Modificar el Alt+Tab.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Katileidys Zerquera Canales Iván Romay Aragón	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alta	Puntos Reales: 4/5
Descripción: Se debe implementar el Alt+Tab como se muestra en la versión 4.0 de Nova Escritorio.	
Observaciones:	
Prototipo de interfaz:	

Tabla 10. HU_8 modificar el Alt+Tab.

Para la realización de la HU_8 se define la tarea de ingeniería 20, para realizar un estudio de los diferentes sistemas operativos sobre como manejan el Alt+Tab, la 21 para realizar la implementación un Alt+Tab igual al que presenta Nova Escritorio 4.0 y la 22 para documentar el cambio así como los artefactos, diagramas y prototipos necesarios.

2.4 Implementación de los requisitos

2.4.1 Vistas de implementación

Los archivos modificados durante la realización de los cambios al Entorno de escritorio se encuentran en `/usr/share/gnome-shell/js/ui` y `/usr/share/gnome-shell/js/ui/status` y las librerías utilizadas para la implementación se encuentran en `/usr/lib/girepository-1.0`.

Brindar soporte para el método de la red cableada

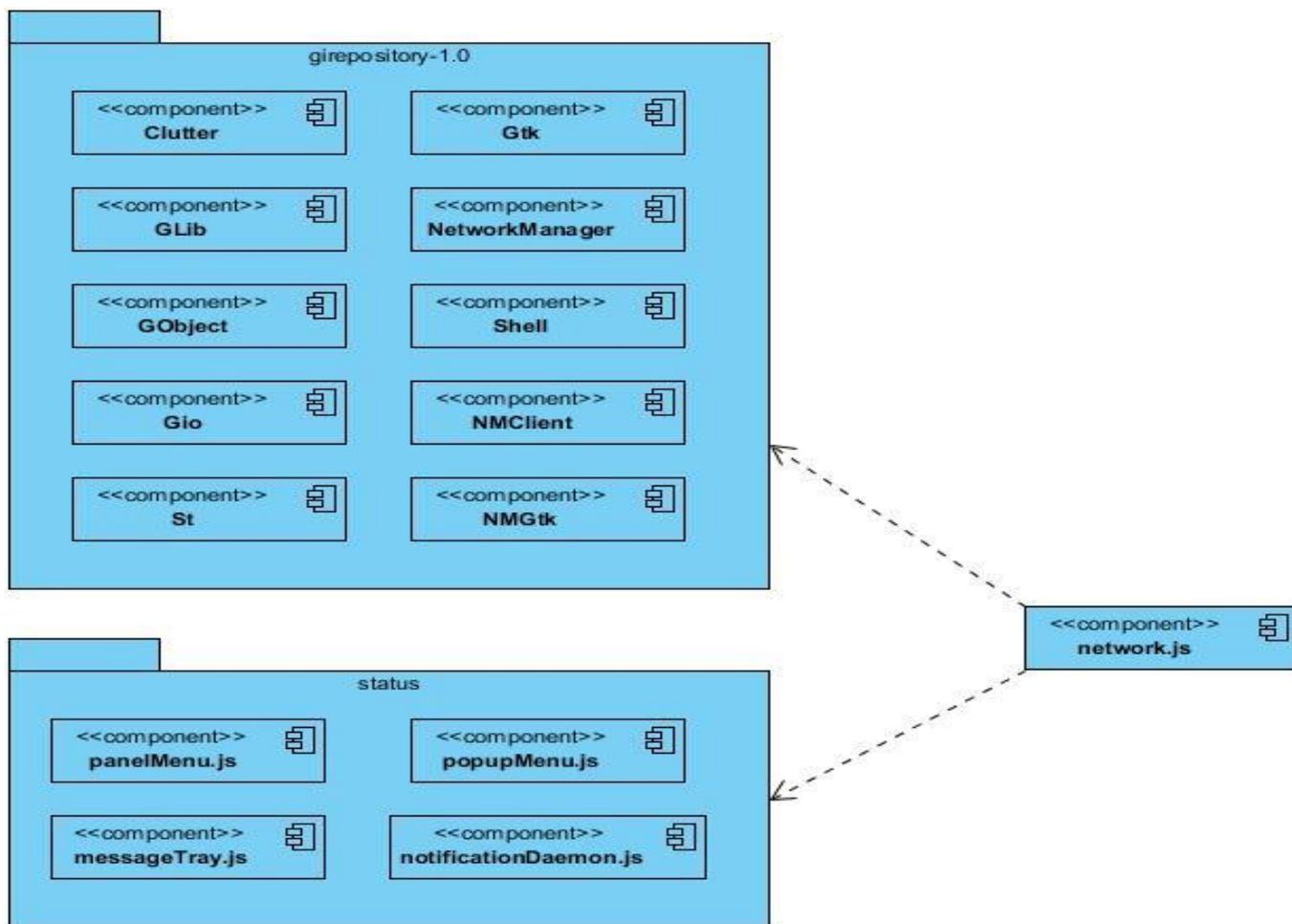


Figura 4. Diagrama de componentes de soporte para el método de la red cableada.

Para dar soporte al método de la red cableada se hace uso de bibliotecas como *Clutter*, *St*, *Gtk* y *NMGtk* para el desarrollo de interfaces gráficas, *Glib* para el tratamiento de los datos, *GObject* para poder interactuar con bibliotecas desarrolladas en diferentes lenguajes de programación, *NetworkManager* y *NMClient* para la gestión de red, *Shell* para el manejo de los ficheros y *Gio* para la gestión de Entradas y Salidas.

Además durante la implementación se utilizan funcionalidades de clases que se encuentran definidas en varios ficheros, como es el caso de la función *popupMenuItem* que pertenece al fichero *popupMenu.js*, el cual es llamado por la función *createSettingsAction* para crear los *label* en el menú de la parte derecha del Panel de menú. Otro caso son los ficheros *messageTray.js* y *notificationDaemon.js* que se encargan de la gestión de los mensajes y notificaciones que lanza el sistema. Además se utiliza la función *SystemIndicator* que pertenece al fichero *panelMenu.js* para gestionar el espacio en el Panel de menú donde se muestra el ícono de la red. Los ficheros que se mencionan se han representado en el diagrama de componentes asociado al cambio realizado (Ver Figura 4).

Se agrega la categoría *WIRED* a la constante *NMConnectionCategory* para que esta sea reconocida posteriormente por los métodos o clases que requieran su uso. Se crean las funciones *ensureActiveConnectionProps*, la cual garantiza que se utilice un dispositivo a la vez y *createSettingsAction* para crear el *label* con el nombre del dispositivo que se está utilizando, en la sección de la red del menú. Luego se crea la clase *NMDeviceWired*. Esta verifica que el dispositivo activado sea el de la red cableada y si es el caso, gestiona los estados por los que va transitando el proceso de conexión y muestra en el Panel de menú los íconos correspondientes a cada uno. Por último, en la clase *NMApplet* que es la que maneja el comportamiento de todos los dispositivos de conexión, se realizan modificaciones en las funciones *tryLatedInit*, *updatedIcon* y *getMainConnection*.

El algoritmo devuelve el ícono correspondiente al tipo de dispositivo que este activo o el ícono de desconexión en caso de que no haya algún dispositivo activo.

Adicionar el botón de apagado del sistema al Panel de menú para las computadoras portátiles.

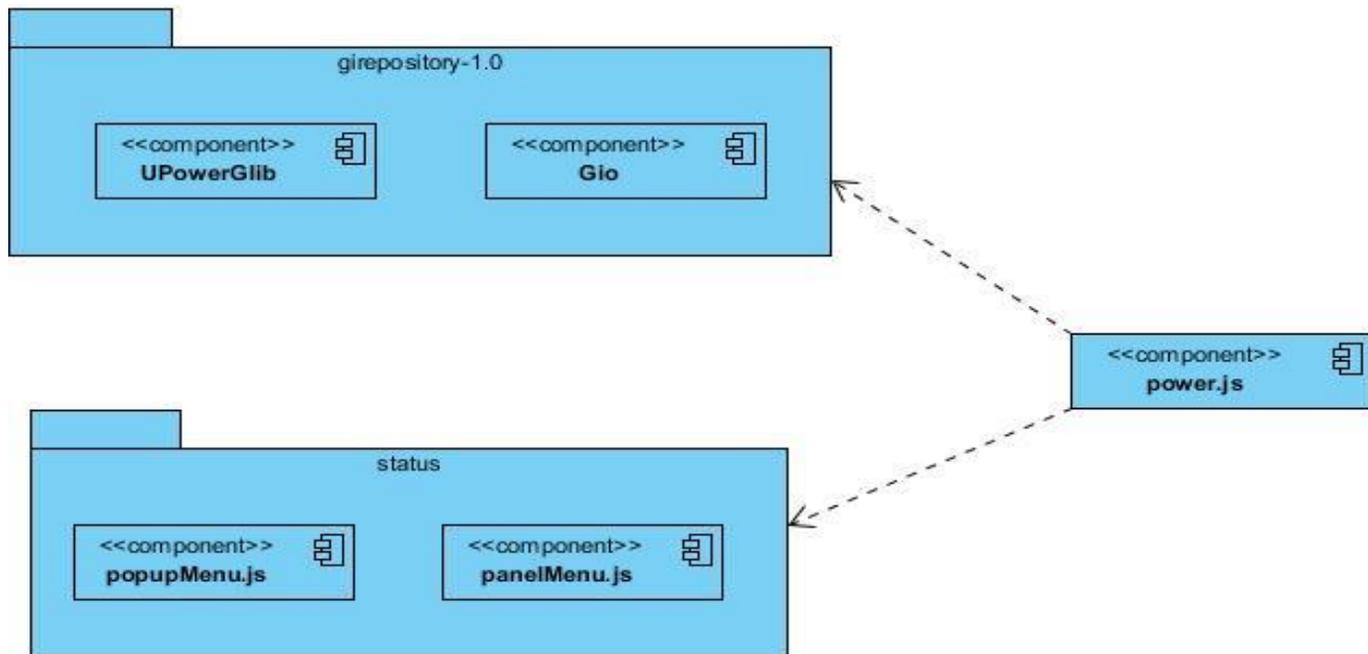


Figura 5. Diagrama de componentes de adicionar el botón de apagado del sistema al Panel de menú para las computadoras portátiles.

Para que se muestre el botón de apagado del sistema se hace uso de bibliotecas como *UPowerGlib* y *Gio*. Estas permiten gestionar todo lo relacionado con el estado en que se encuentra la batería de las laptops y los íconos que se van a mostrar en el Panel de menú respectivamente. Además durante la implementación se utiliza la clase *PopupSubMenuMenuItem* que se encuentra definida en el fichero *popupMenu.js*. Los ficheros que se mencionan se han representado en el diagrama de componentes asociado al cambio realizado (Ver Figura 5).

Se procede a declarar una variable que va a contener un espacio en el Panel de menú con la ayuda de las funcionalidades que brinda la clase *SystemIndicator* que pertenece al fichero *panelMenu.js*, a la cual se le pasa el ícono de apagado del sistema. Luego se analiza si el dispositivo contiene batería y en dependencia del resultado que se obtenga, se mantiene el ícono de apagado del sistema y se oculta o no el ícono de la batería.

Mostrar la sección de todas las aplicaciones en el menú agrupadas por categorías.

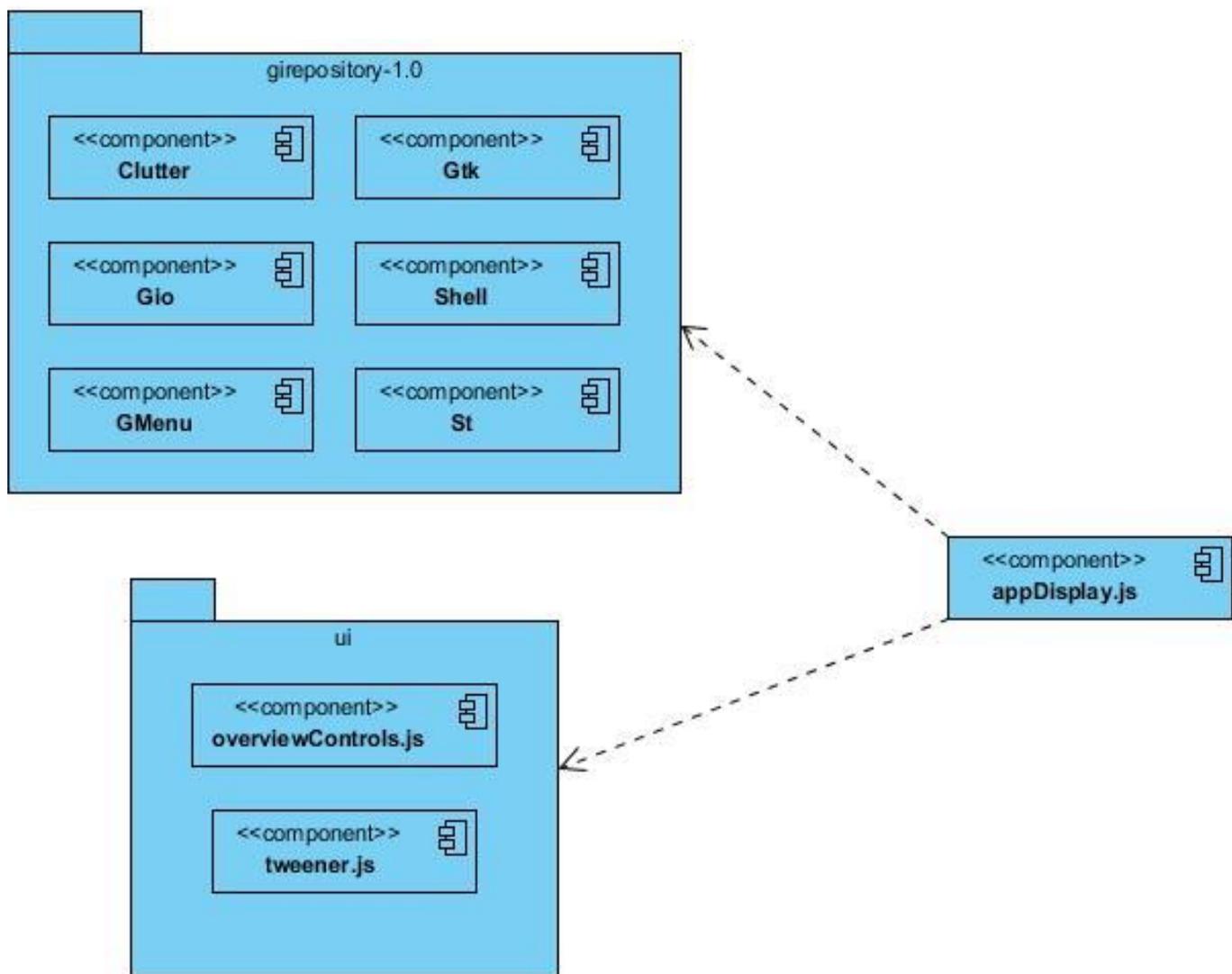


Figura 6. Diagrama de componentes de mostrar la sección de todas las aplicaciones en el menú agrupadas por categorías.

Para agrupar las aplicaciones en el menú por categorías se hace uso de bibliotecas como *Clutter*, *St* y *Gtk* para el desarrollo de interfaces gráficas, *Shell* para el manejo de los ficheros y *Gio* para la gestión de Entradas y Salidas. Además durante la implementación se utilizan funciones como *SIDE_CONTROLS_ANIMATION_TIME* donde se maneja la duración de las animaciones y *addTween* que adiciona las animaciones al menú, que se encuentran definidas en los ficheros *overviewControls.js* y *tweener.js* respectivamente. Los ficheros que se mencionan se han representado en el diagrama de componentes asociado al cambio realizado (Ver Figura 5).

Se procede a crear una función que recibe como parámetro un directorio, se inicializa una variable que se comporta como un contador y con la ayuda de la biblioteca *Gmenu* se analiza la cantidad de aplicaciones que contiene dicho directorio y se aumenta el contador en 1 por cada aplicación que se detecte. Luego otro método se encarga de analizar el contador y si este es mayor o igual a 1 se muestra la categoría en el menú, en caso contrario no se muestra.

Modificar el botón Actividades del sistema

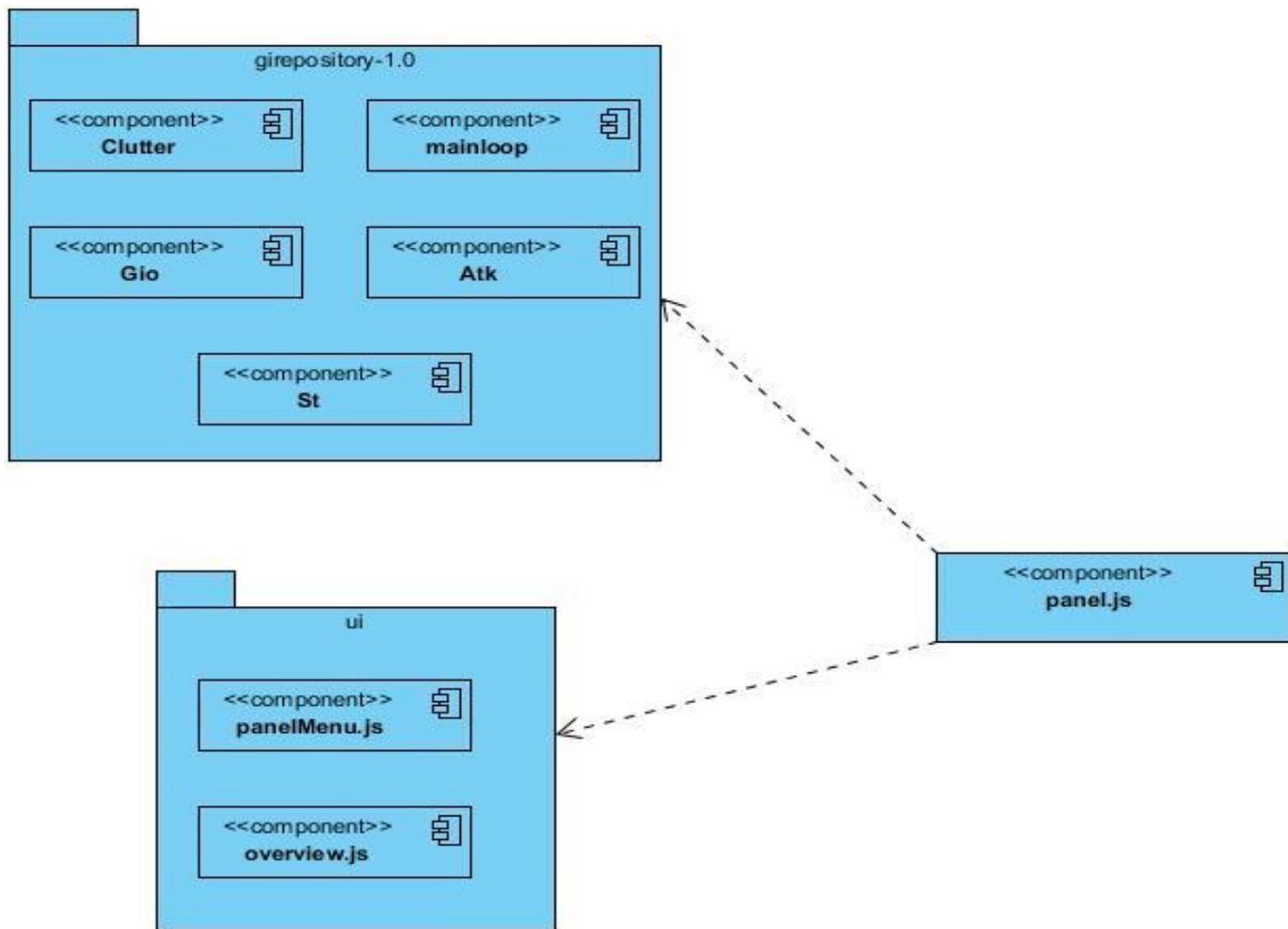


Figura 7. Diagrama de componentes de modificar el botón Actividades del sistema.

Para modificar el botón Actividades se hace uso de bibliotecas como *Clutter* y *St* para el desarrollo de interfaces gráficas, *Atk* para crear interfaces con grandes características de accesibilidad, *Gio* para la gestión de Entradas y Salidas y *MainLoop* permite que se ejecute el *shell*. Además durante la implementación se utilizan las funciones *shouldToggleByCornerOrButton* y *toggle* de la clase *OverView* que pertenece al fichero *overView.js* para el manejo de las animaciones y la clase *Button* del fichero *panelMenu.js* para crear y agregar funcionalidades al botón de menú del sistema. Los ficheros que se mencionan se han representado en el diagrama de componentes asociado al cambio realizado (Ver Figura 7).

Se procede a crear un ícono, al cual se le pasa la imagen que va a mostrar (logo de Nova). Este se posiciona en el Panel de menú donde se encontraba el *Label* que mostraba la palabra Actividades, manteniendo las mismas funcionalidades y estilos de clases.

Agregar los botones minimizar y maximizar

Para agregar los botones minimizar y maximizar en la barra de título de las ventanas se procede a agregar las palabras *maximize* y *minimize* en el método *button-layout* que se encuentra ubicado en el fichero */usr/share/glib-2.0/schemas/org.gnome.shell.gschema.xml*.

Modificar el ícono Accesibilidad del sistema.

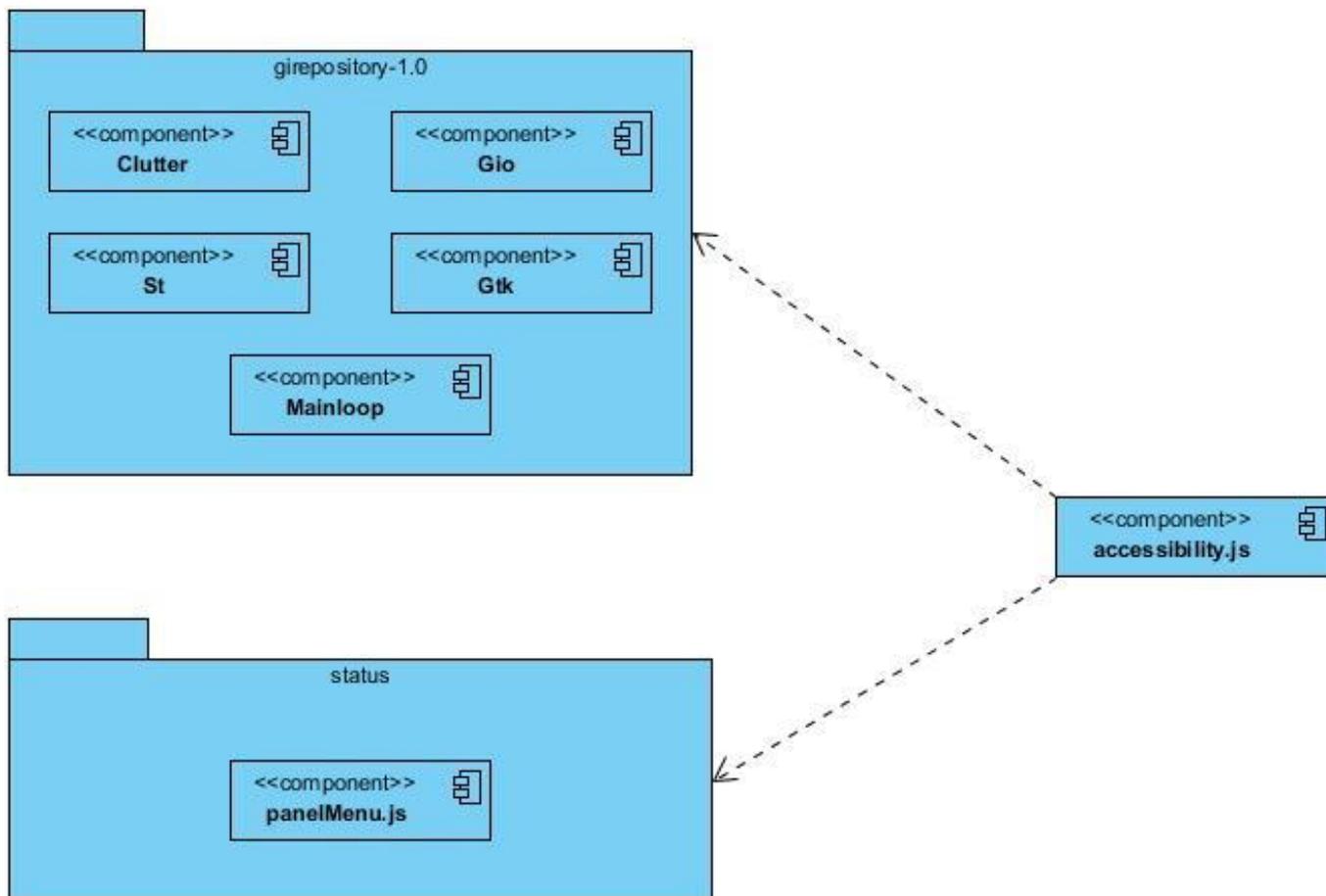


Figura 8. Diagrama de componentes modificar el ícono Accesibilidad del sistema.

Para modificar el ícono de Accesibilidad se hace uso de bibliotecas como *Clutter*, *St* y *Gtk* para el desarrollo de interfaces gráficas, *Gio* para la gestión de Entradas y Salidas y *MainLoop* permite que se ejecute el *shell*. Además durante la implementación se utilizan las funciones de la clase *Button* del fichero *panelMenu.js* para crear y agregar funcionalidades al botón Accesibilidad del sistema. Los ficheros que se mencionan se han representado en el diagrama de componentes asociado al cambio realizado (Ver Figura 8).

Se modifica el ícono del botón Accesibilidad, manteniendo las mismas funcionalidades y estilo de clases.

Modificar el Alt+Tab

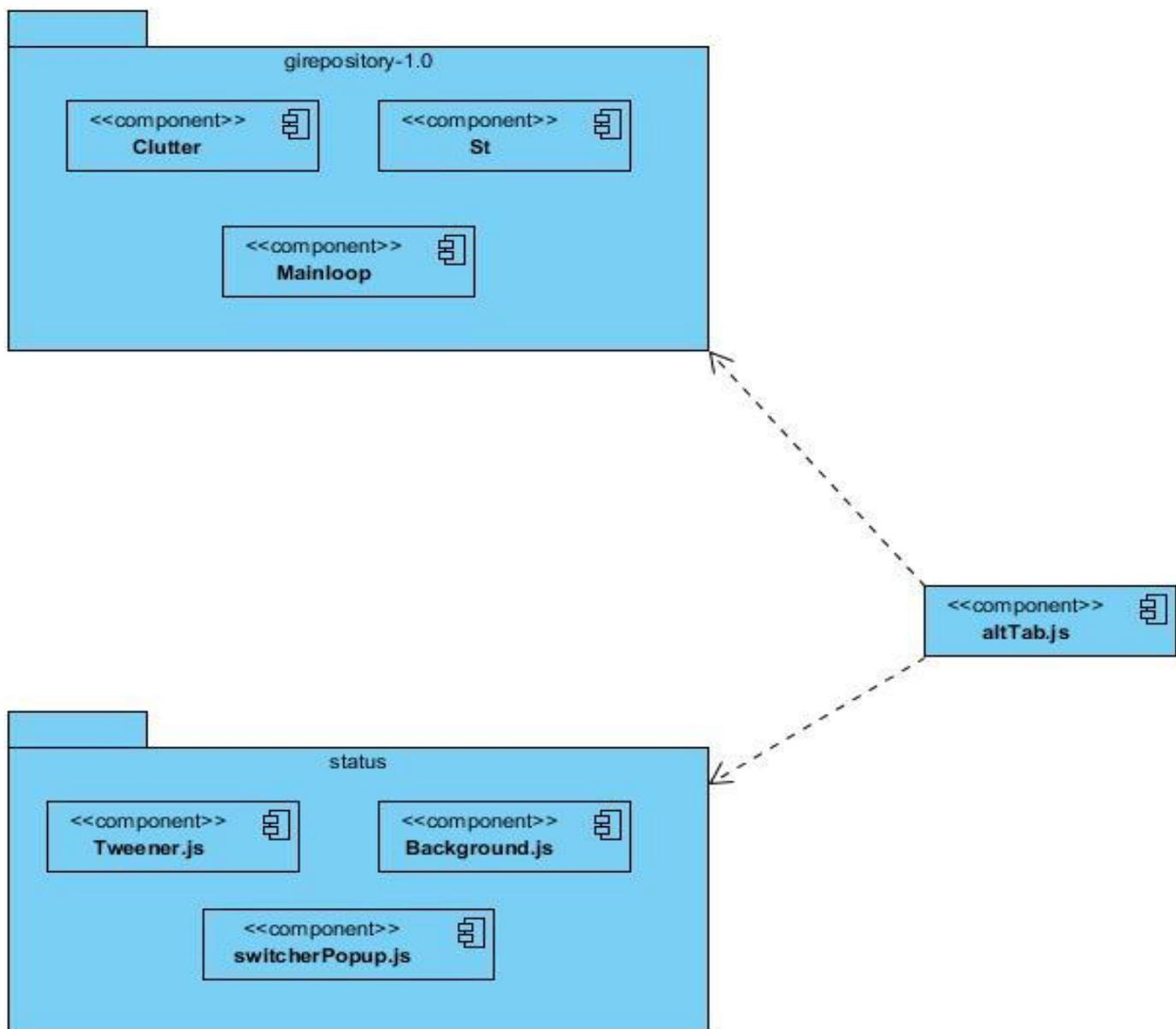


Figura 9. Diagrama de componentes modificar el Alt+Tab.

Para modificar el Alt+Tab se hace uso de bibliotecas como *Clutter* y *St* para el desarrollo de interfaces gráficas y *MainLoop* para que se ejecute el *shell*. Durante la implementación se utilizan las funciones de la clase *Switcher* del fichero *altTab.js* para construir el componente que hace rotar las aplicaciones. Además se utilizan las funcionalidades de la clase *Manager* del mismo fichero para inicializar funciones del *Switcher* que permiten pasar las aplicaciones a favor o en contra de las manecillas del reloj, mostrar todas las aplicaciones de una o todas las áreas de trabajo y todas las ventanas de una única aplicación. Los ficheros que se mencionan se han representado en el diagrama de componentes asociado al cambio realizado (Ver Figura 9).

Adicionar nueva forma para agregar y eliminar las aplicaciones a Favoritos.

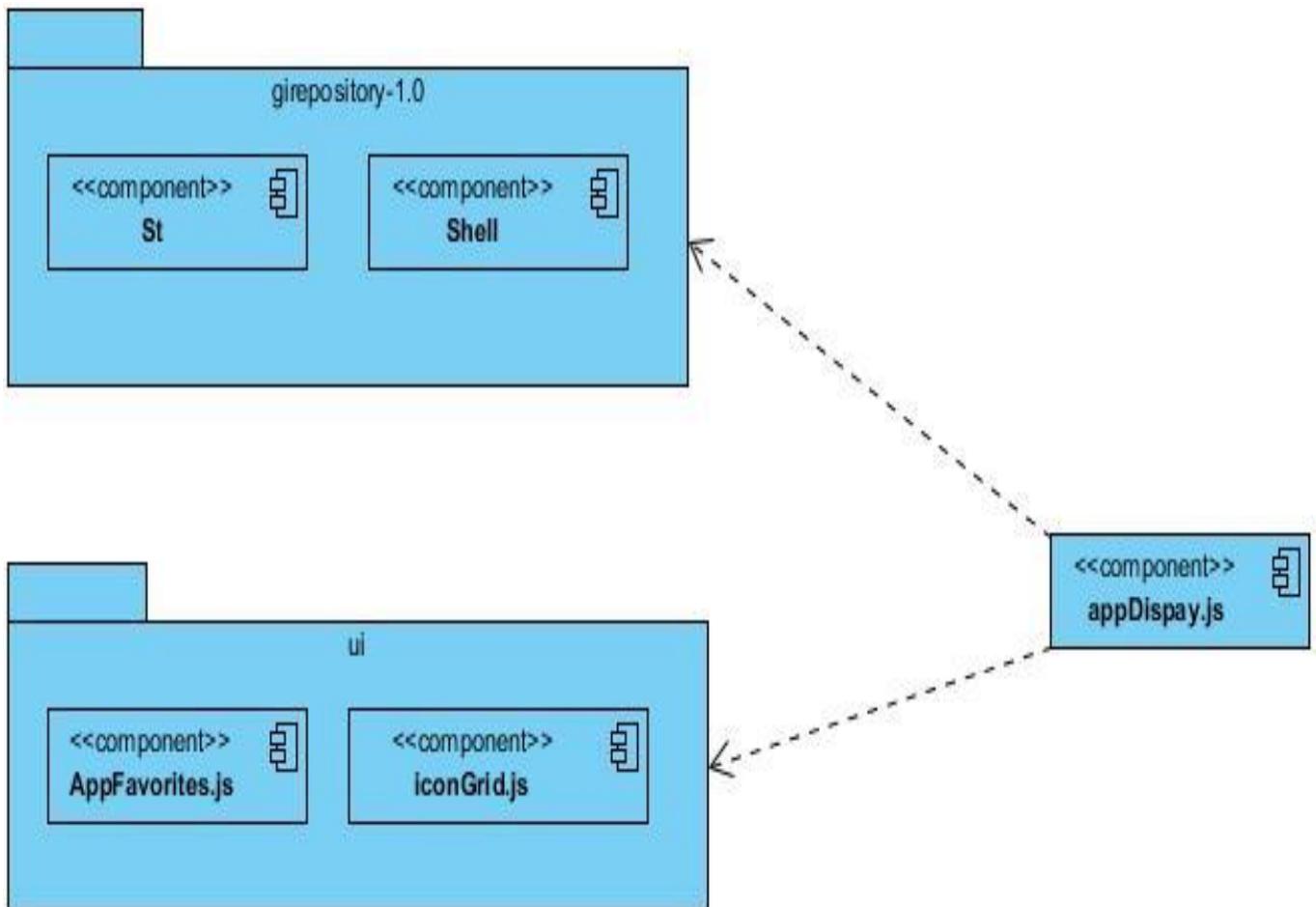


Figura 10. Diagrama de componentes adicionar nueva forma para agregar y eliminar las aplicaciones a Favoritos.

Para adicionar una nueva forma de agregar y eliminar las aplicaciones a Favoritos se utiliza *Shell* para el manejo de los ficheros y *St* para la creación de íconos y botones. Además se implementan métodos como *showStar* y *hideStar* en el fichero *iconGrid.js* y se utilizan funciones ya definidas como *addFavorite* del fichero *AppFavorites.js*.

Se procede a crear un botón al cual se le adiciona un ícono y se posiciona en la parte superior derecha del contenedor de la aplicación. Luego se crea una función para mostrar el ícono y otra para ocultarlo.

En el fichero *appDisplay.js* se crea el método *onStarPress* que al dar clic encima del ícono la aplicación se adicione a favoritos y cambie la forma del mismo, si la aplicación es favorita al dar clic encima del ícono esta se elimina de los favoritos y el ícono regresa a su estado inicial.

Se crea otra función para el manejo del evento *get_hover* es decir cuando el mouse pase por encima de la aplicación se especifica que ícono debe mostrarse si es favorita o no. Los ficheros que se mencionan se han representado en el diagrama de componentes asociado al cambio realizado (Ver Figura 10).

2.4.2 Programación por pares

Para dar solución a los requisitos funcionales propuestos en el presente trabajo, el equipo de desarrollo, entre sus tareas a realizar se propuso el estudio del lenguaje de programación *JavaScript* y de los paquetes de clases que conforman al Entorno de escritorio Gnome. Una vez adquiridos los conocimientos básicos en el tema, procede a la programación para dar cumplimiento a los requerimientos. En dependencia de la complejidad, en caso de ser Alta, el equipo trabaja en pareja desarrollando un mismo requisito y si esta es Media o baja lo hacen por separado desarrollando diferentes requisitos a la vez.

Conclusiones Parciales

- En este capítulo se desarrollaron los principales requisitos que tributan a los cambios de la AI del Entorno de escritorio de Nova Escritorio 5.0 y proveen al sistema de un mejor acabado.
- Se presenta por primera vez el documento de la AI del entorno de escritorio de Nova Escritorio, logrando tener una descripción detallada de los componentes del Entorno de escritorio para Nova escritorio 5.0.

Capítulo 3: Evaluación de los cambios para la AI del Entorno de escritorio de Nova Escritorio 5.0.

Introducción

En este capítulo se realizan las pruebas a los cambios que se desarrollan para la AI del Entorno de escritorio de Nova Escritorio 5.0, siendo esta una actividad en la cual el sistema es ejecutado bajo condiciones o requerimientos específicos. Para evaluar la calidad de las funcionalidades desarrolladas se utilizan los niveles de pruebas Unitarias, Funcionalidad y Aceptación. Se diseñan y describen los casos de pruebas correspondientes tanto al código como a la interfaz del sistema, los resultados obtenidos se observan y se registran. Además se rectifican los aspectos negativos detectados, llevando a cabo un seguimiento de los mismos.

3.1 Pruebas unitarias

Para realizar las pruebas unitarias se utiliza el método de **Caja Blanca**. Este se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado. La prueba de caja blanca, es un método de diseño de casos de prueba donde se usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que (1) garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo; (2) ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa; (3) ejecuten todos los bucles en sus límites y con sus límites operacionales y (4) ejerciten las estructuras internas de datos para asegurar su validez. (Pressman, 2001)

Existen varios tipos de pruebas de caja blanca, ellos son:

- **Prueba de condición:** Es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa.
- **Prueba de flujo de datos:** Se seleccionan caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.
- **Prueba de bucles:** Es una técnica de prueba de caja blanca que se centra exclusivamente en la validez de las construcciones de bucles.
- **Prueba del camino básico:** Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. (Martinez Furones y Figueredo Aguilar, 2012)

En este caso se va a aplicar la llamada prueba de camino básico con el objetivo de comprobar que cada línea de código se ejecuta al menos una vez.

A continuación se definen los pasos necesarios para realizar dicha prueba:

1. Generar grafo de flujos de datos.
2. Cálculo de complejidad ciclomática, $V(G)$.
3. Determinar los caminos independientes o básicos.
4. Generar un caso de prueba para cada camino de ejecución.

Un grafo de flujo está constituido por tres componentes fundamentales:

Nodo: Es usado para representar una o más secuencias procedimentales, es decir, representa cada línea del código que pertenece a la funcionalidad a evaluar.

Arista: Es la relación entre cada nodo, e implica el sentido del flujo de la ejecución de cada línea de código.

Regiones: Son las áreas cerradas delimitadas por las aristas, estas representan la cantidad de caminos independientes al ejecutarse un programa.

Método `getIndicatorIcon`

A continuación se muestra el código y selección de las áreas que representan cada nodo:

```
getIndicatorIcon: function() {  
  if (this._device.active_connection) {  
    let state = this._device.active_connection.state;  
    if (state == NetworkManager.ActiveConnectionState.ACTIVATING) {  
      return 'network-wired-acquiring-symbolic';  
    } else if (state == NetworkManager.ActiveConnectionState.ACTIVATED) {  
      if (this._canReachInternet())  
        return 'network-wired-symbolic';  
      else  
        return 'network-wired-no-route-symbolic';  
    } else {  
      return 'network-wired-disconnected-symbolic';  
    }  
  } else  
    return 'network-wired-disconnected-symbolic';  
}
```

Se construye un grafo que representa el flujo de la funcionalidad y cada arista indica todos los posibles caminos a seguir.

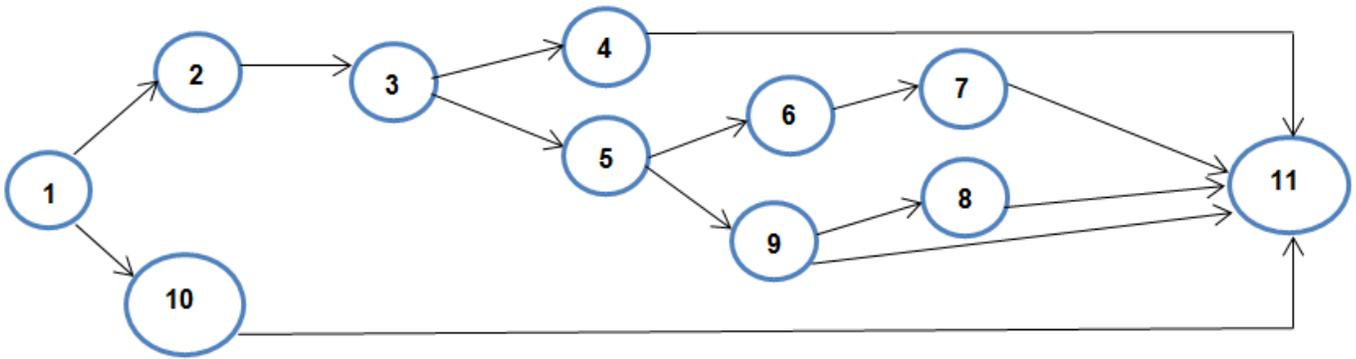


Figura 11. Gráfico de flujo de la funcionalidad getIndicatorIcon.

A partir de este nodo se obtiene la complejidad ciclomática, que está dada por las siguientes fórmulas que deben dar el mismo resultado:

1. $V(G) = (A - N) + 2$ Donde "A" es el número de aristas y "N" el de nodos.
2. $V(G) = P + 1$ Donde "P" es la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).
3. $V(G) = R + 1$ Donde "R" es la cantidad total de regiones.

Obteniendo como resultado:

1. $V(G) = (14 - 11) + 2 = 5$
2. $V(G) = 4 + 1 = 5$
3. $V(G) = 4 + 1 = 5$

Como se puede observar se obtienen los mismos resultados. A continuación se definen los caminos independientes:

- R1: 1-2-3-4-11
- R2: 1-2-3-5-9-11
- R3: 1-2-3-5-6-8-11
- R4: 1-2-3-5-6-7-11
- R5: 1-10-11

Luego de obtener los caminos básicos es necesario realizar los casos de prueba, debe obtenerse al menos uno por cada camino definido:

Caso de prueba para el camino básico 1:

R1: 1-2-3-4-11

Entrada: El actor debe conectar el cable de red.

Función del código: Se verifica si el dispositivo de red está conectado, se crea una variable para guardar el estado de la conexión. Si el estado de la red es 'activando' se muestra el ícono 'network-wired-acquiring-symbolic'.

Resultado: Luego de aplicar el caso de prueba, se pudo comprobar que el flujo de trabajo de la funcionalidad está correcto.

Caso de prueba para el camino básico 2:

R2: 1-2-3-5-9-11

Entrada: El actor debe conectar el cable de red.

Función del código: Se verifica si el dispositivo de la red está conectado, se crea una variable para guardar el estado de la conexión. Si el estado de la red no es 'activando' se verifica si ya está 'activada' en caso de que no esté activa se muestra el ícono '*network-wired-disconnected-symbolic*'.

Resultado: Luego de aplicar el caso de prueba, se pudo comprobar que el flujo de trabajo de la funcionalidad está correcto.

Caso de prueba para el camino básico 3:

R3: 1-2-3-5-6-8-11

Entrada: El actor debe conectar el cable de red.

Función del código: Se verifica si el dispositivo de red está conectado y se crea una variable para guardar el estado de la conexión. Si el estado de la red no es 'activando', se verifica si ya está 'activada', en caso de que no esté activa se muestra el ícono '*network-wired-disconnected-symbolic*'.

Resultado: Luego de aplicar el caso de prueba, se pudo comprobar que el flujo de trabajo de la funcionalidad está correcto.

Caso de prueba para el camino básico 4:

R4: 1-2-3-5-6-7-11

Entrada: El actor debe conectar el cable de red.

Función del código: Se verifica si el dispositivo de red está conectado y se crea una variable para guardar el estado de la conexión. Si el estado de la red no es 'activando', se verifica si ya está 'activada' y si existe una conexión, en ese caso muestra el ícono '*network-wired-symbolic*'.

Resultado: Luego de aplicar el caso de prueba, se pudo comprobar que el flujo de trabajo de la funcionalidad está correcto.

Caso de prueba para el camino básico 5:

R5: 1-10-11

Entrada: El actor debe conectar el cable de red.

Función del código: Se verifica si el dispositivo de red está conectado, en caso de no estar conectado se muestra el ícono '*network-wired-disconnected-symbolic*'.

Resultado: Luego de aplicar el caso de prueba, se pudo comprobar que el flujo de trabajo de la funcionalidad está correcto.

Método *_syncIcon*

A continuación se muestra el código y selección de las áreas que representan cada nodo:

```

_synclcon: function() {
  let icon = this._proxy.Icon; } 1
  if (icon) {
    let gicon = Gio.icon_new_for_string(icon);
    this._indicator.gicon = gicon;
    this._item.icon.gicon = gicon; } 2
  } else {
    this._indicator.hide(); } 3
  }, 4
}

```

Se construye un grafo que representa el flujo de la funcionalidad y cada arista indica todos los posibles caminos a seguir.

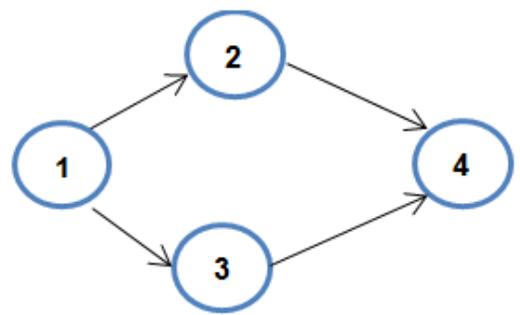


Figura 12. Gráfico de flujo de la funcionalidad synclcon.

A partir de este nodo se obtiene la complejidad ciclomática, que está dada por las siguientes fórmulas que deben dar el mismo resultado:

$$V(G) = (4 - 4) + 2 = 2$$

$$V(G) = 1 + 1 = 2$$

$$V(G) = 1 + 1 = 2$$

Como se puede observar se obtienen los mismos resultados, a continuación se definen los caminos independientes:

R1: 1-2-4

R2: 1-3-4

Luego de obtener los caminos básicos es necesario realizar los casos de prueba, debe obtenerse al menos uno por cada camino definido:

Caso de prueba para el camino básico 1:

R1: 1-2-4

Entrada: El actor debe encender la PC e iniciar el sistema.

Función del código: Se crea una variable donde se guarda si el dispositivo contiene o no batería, si tiene batería se muestra el ícono de esta y el de apagado del sistema en el Panel de menú.

Resultado: Luego de aplicar el caso de prueba, se pudo comprobar que el flujo de trabajo de la funcionalidad está correcto.

Caso de prueba para el camino básico 2:

R1: 1-3-4

Entrada: El actor debe encender la PC e iniciar el sistema.

Función del código: Se crea una variable donde se guarda si el dispositivo contiene o no batería, si no tiene batería solo se muestra el ícono de apagado del sistema en el Panel de menú.

Resultado: Luego de aplicar el caso de prueba, se pudo comprobar que el flujo de trabajo de la funcionalidad está correcto.

Método `_onStarPress`

A continuación se muestra el código y selección de las áreas que representan cada nodo:

```
_onStarPress: function(actor, event) {  
    let button = event.get_button();  
    if (button == 1) {  
        let favs = AppFavorites.getAppFavorites();  
        let isFavorite = favs.isFavorite(this.app.get_id());  
        if (isFavorite){  
            favs.removeFavorite(this.app.get_id());  
            this.icon.star.icon_name = 'non-starred-symbolic';  
        }else {  
            favs.addFavorite(this.app.get_id());  
            this.icon.star.icon_name = 'starred-symbolic';  
        }  
    }  
},
```

The diagram illustrates the mapping of code blocks to nodes in a flow graph. Node 1 is the initial function call. Node 2 is the conditional check for button 1. Node 3 is the conditional check for isFavorite. Node 4 is the conditional execution of removeFavorite or addFavorite. Node 5 is the final closing brace of the function.

Se construye un grafo el cual representa el flujo de la funcionalidad y cada arista indica todos los posibles caminos a seguir.

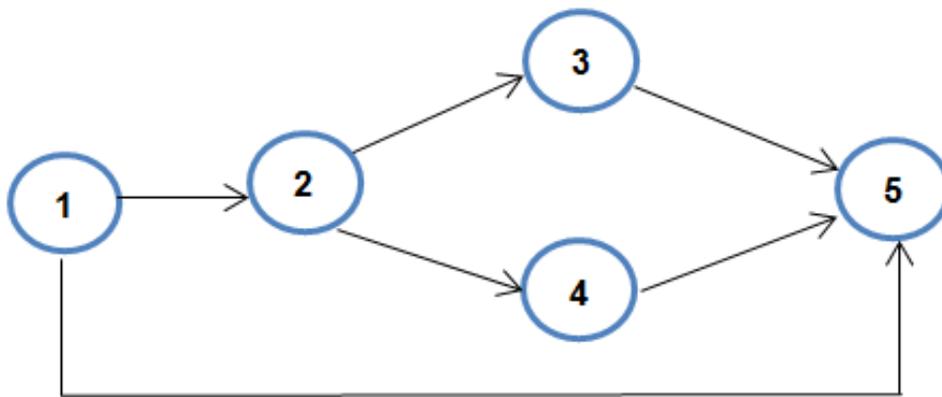


Figura 13. Gráfico de flujo de la funcionalidad onStarPress.

A partir de este nodo se obtiene la complejidad ciclomática, que está dada por las siguientes fórmulas que deben dar el mismo resultado:

$$V(G) = (6 - 5) + 2 = 3$$

$$V(G) = 2 + 1 = 3$$

$$V(G) = 2 + 1 = 3$$

Como se puede observar se obtienen los mismos resultados, a continuación se definen los caminos independientes:

R1: 1-2-3-5

R2: 1-2-4-5

R3: 1-5

Luego de obtener los caminos básicos es necesario realizar los casos de prueba, debe obtenerse al menos uno por cada camino definido:

Caso de prueba para el camino básico #1:

R1: 1-2-3-5

Entrada: Se recibe un actor y un evento.

Función del código: Se crea una variable donde se guarda el evento que se pasa por parámetro y se verifica si este es el clic primario, en caso de serlo se crean variables para verificar si la aplicación donde se produce el evento es favorita o no, si es favorita se elimina del menú de Favoritos y el ícono que recibe el evento cambia su forma.

Resultado: Luego de aplicar el caso de prueba, se pudo comprobar que el flujo de trabajo de la funcionalidad está correcto.

Caso de prueba para el camino básico 2:

R1: 1-2-4-5

Entrada: Se recibe un actor y un evento.

Función del código: Se crea una variable donde se guarda el evento que se pasa por parámetro y se verifica si este es el clic primario, en caso de serlo se crean variables para verificar si la aplicación donde se produce el evento es favorita o no, si no es favorita se adiciona al menú de Favoritos y el ícono que recibe el evento cambia su forma.

Resultado: Luego de aplicar el caso de prueba, se pudo comprobar que el flujo de trabajo de la funcionalidad está correcto.

Caso de prueba para el camino básico 3:

R1: 1-5

Entrada: Se recibe un actor y un evento.

Función del código: Se crea una variable donde se guarda el evento que se pasa por parámetro y se verifica si este es el clic primario, en caso de no serlo se termina el método.

Resultado: Luego de aplicar el caso de prueba, se pudo comprobar que el flujo de trabajo de la funcionalidad está correcto.

Método `_contCategory`

A continuación se muestra el código y selección de las áreas que representan cada nodo:

```
function _contCategory(dir){
```

```
    let cont = 0;
```

```
    let iter = dir.iter();
```

```
    let appSystem = Shell.AppSystem.get_default();
```

```
    let nextType;
```

```
    while ((nextType = iter.next()) != GMenu.TreeItemType.INVALID) {
```

```
        if (nextType == GMenu.TreeItemType.ENTRY) {
```

```
            let entry = iter.get_entry();
```

```
            let app = appSystem.lookup_app_by_tree_entry(entry);
```

```
            if (!entry.get_app_info().get_nodisplay())
```

```
                cont ++;
```

```
        } else if (nextType == GMenu.TreeItemType.DIRECTORY) {
```

```
            let itemDir = iter.get_directory();
```

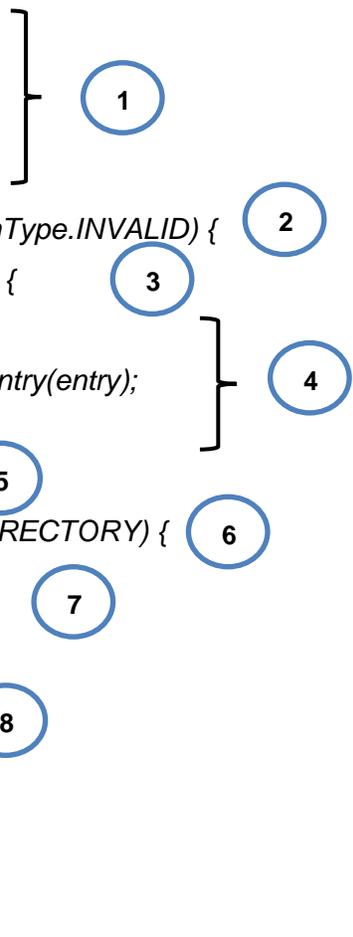
```
            if (!itemDir.get_is_nodisplay())
```

```
                cont += _contCategory(itemDir);
```

```
        }
```

```
    }  
    return cont;
```

```
};
```



Se construye un grafo el cual representa el flujo de la funcionalidad y cada arista indica todos los posibles caminos a seguir.

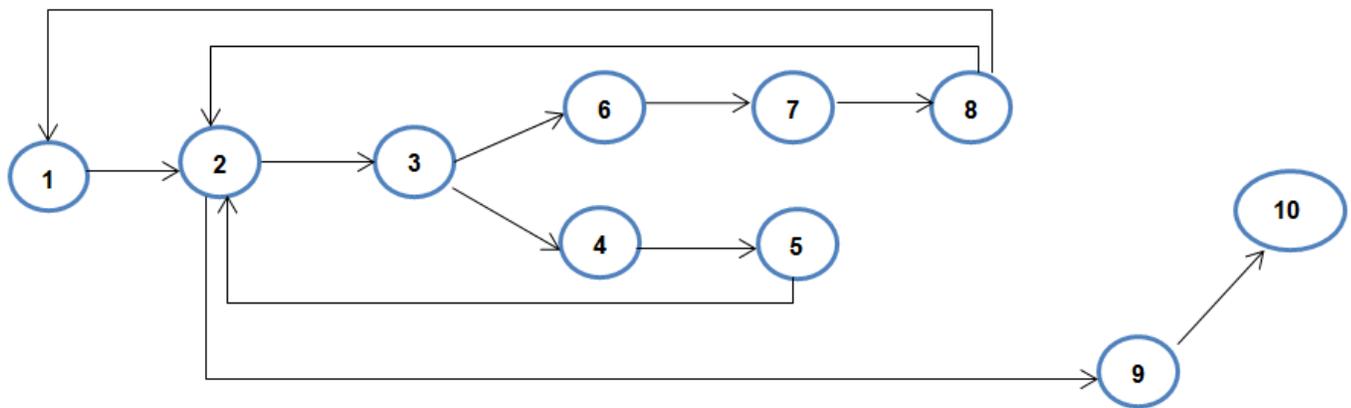


Figura 14. Gráfico de flujo de la funcionalidad contCategory.

A partir de este nodo se obtiene la complejidad ciclomática, que está dada por las siguientes fórmulas que deben dar el mismo resultado:

$$V(G) = (12 - 10) + 2 = 4$$

$$V(G) = 3 + 1 = 4$$

$$V(G) = 3 + 1 = 4$$

Como se puede observar se obtienen los mismos resultados, a continuación se definen los caminos independientes:

R1: 1-2-9-10

R2: 1-2-3-6-7-8-1

R3: 1-2-3-6-7-8-2

R4: 1-2-3-4-5-2

Luego de obtener los caminos básicos es necesario realizar los casos de prueba, debe obtenerse al menos uno por cada camino definido:

Caso de prueba para el camino básico 1:

R1: 1-2-9-10

Entrada: Se recibe un directorio.

Función del código: Se crean las variables necesarias, entre ellas un contador para guardar la cantidad de aplicaciones que contiene el directorio recibido por parámetro. Si el directorio pasado por parámetro es inválido se retorna el contador.

Resultado: Luego de aplicar el caso de prueba, se pudo comprobar que el flujo de trabajo de la funcionalidad está correcto.

Caso de prueba para el camino básico 2:

R2: 1-2-3-6-7-8-1

Entrada: Se recibe un directorio.

Función del código: Se crean las variables necesarias, entre ellas un contador para guardar la cantidad de aplicaciones que contiene el directorio recibido por parámetro. Mientras el directorio pasado por parámetro no sea inválido se verifica su tipo, si no es una aplicación y es un directorio, y si la información que contiene se puede mostrar se guarda este en una variable y se llama el método recursivamente.

Resultado: Luego de aplicar el caso de prueba, se pudo comprobar que el flujo de trabajo de la funcionalidad está correcto.

Caso de prueba para el camino básico #3:

R3: 1-2-3-6-7-8-2

Entrada: Se recibe un directorio.

Función del código: Se crean las variables necesarias, entre ellas un contador para guardar la cantidad de aplicaciones que contiene el directorio recibido por parámetro. Mientras el directorio pasado por parámetro no sea inválido se verifica su tipo, si no es una aplicación y es un directorio, y si la información que contiene no se puede mostrar se regresa al inicio del ciclo.

Resultado: Luego de aplicar el caso de prueba, se pudo comprobar que el flujo de trabajo de la funcionalidad está correcto.

Caso de prueba para el camino básico #4:

R4: 1-2-3-4-5-2

Entrada: Se recibe un directorio.

Función del código: Se crean las variables necesarias, entre ellas un contador para guardar la cantidad de aplicaciones que contiene el directorio recibido por parámetro. Mientras el directorio pasado por parámetro no sea inválido se verifica su tipo, si es una aplicación, y la información que contiene se puede mostrar se aumenta el contador en uno y se regresa al inicio del ciclo.

Resultado: Luego de aplicar el caso de prueba, se pudo comprobar que el flujo de trabajo de la funcionalidad está correcto.

3.2 Pruebas de funcionalidad

Para realizar las pruebas de funcionalidad se utiliza el método de **Caja Negra**. Estas se llevan a cabo sobre la interfaz del sistema. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto. Además examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software. (Pressman, 2001)

Diseño de caso de prueba de Mostrar la sección de todas las aplicaciones en el menú agrupadas por categorías.

Escenario	Descripción	Respuesta del sistema	Flujo central
EP: 1 Mostrar las aplicaciones en Categorías en el menú.	Muestra la sección de todas las aplicaciones en el menú agrupadas por categorías.	Se deben mostrar las aplicaciones agrupadas por categorías.	<ul style="list-style-type: none"> ➤ Se da clic en la estrella que se muestra en la parte izquierda del Panel de menú. ➤ Se da clic en el botón "Todas".
		Se deben mostrar las aplicaciones.	<ul style="list-style-type: none"> ➤ Se da clic en cualquiera de los contenedores para abrirlo y mostrar las aplicaciones que posee.
		Se debe ejecutar correctamente.	<ul style="list-style-type: none"> ➤ Se da clic en una de las aplicaciones que se muestran para ejecutarla.

Tabla 11. Diseño de caso de prueba de Mostrar la sección de todas las aplicaciones en el menú agrupadas por categorías.

Diseño de caso de prueba de Modificar el botón Actividades del sistema.

Escenario	Descripción	Respuesta del sistema	Flujo central
EP: 1 Mostrar el botón del menú de aplicaciones con la estrella de Nova.	Muestra el ícono con la estrella identificativa de Nova como botón del menú de aplicaciones.	Se debe mostrar el menú de aplicaciones y estas se deben ejecutar correctamente.	<ul style="list-style-type: none"> ➤ Se da clic en la estrella que se muestra en la parte izquierda del Panel de menú y se ejecutan algunas de las aplicaciones para confirmar si el menú funciona correctamente.

Tabla 12. Diseño de caso de prueba de Modificar el botón Actividades del sistema.

Adicionar el botón de apagado del sistema al Panel de menú para una computadora portátil.

Escenario	Descripción	Respuesta del sistema	Flujo central
EP: 1 Mostrar botón de apagado.	Se muestra el botón de apagado del sistema en la parte derecha del Panel de menú.	Se debe mostrar el botón de apagado y las opciones que se ejecuten lo deben hacer correctamente.	<ul style="list-style-type: none"> ➤ Se da clic en el botón de apagado que se muestra en la parte derecha del Panel de menú. ➤ Se da clic en las opciones de apagado que se muestran en el submenú que se despliega para verificar que el botón funciona correctamente.

Tabla 13. Adicionar el botón de apagado del sistema al Panel de menú para las computadoras portátiles.

Diseño de caso de prueba de Soporte para el método de la red cableada.

Escenario	Descripción	Respuesta del sistema	Flujo central
EP: 1 Red cableada activada	Debe conectarse el cable de red a la computadora.	Debe aparecer en la parte derecha del Panel de menú el ícono que muestra que la red cableada se está conectando y después debe cambiarse por el de conectado y además la PC debe quedar conectada a la red.	<ul style="list-style-type: none"> ➤ Conectar el cable de red. ➤ Dirigirse a la sección de red que se encuentra en el menú de la parte derecha del Panel de menú. ➤ Verificar en el submenú de la red cableada que esta se encuentre encendida.
EP: 2 Red cableada desactivada	Debe desconectarse el cable de red de la PC.	La PC debe quedar desconectada de la red y cambiar el ícono de conectado por el ícono de desconectado. (En caso que la PC esté conectada mediante otro dispositivo de red a la misma vez se debe cambiar el ícono por el del dispositivo activo. Ejemplo: VPN o WI-FI)	<ul style="list-style-type: none"> ➤ Desconectar el cable de red de la PC o apagar la red cableada en la sección de red en el menú de la parte derecha del Panel de menú.

Tabla 14. Diseño de caso de prueba de Soporte para el método de la red cableada.

Diseño de caso de prueba de Modificar el ícono Accesibilidad del sistema.

Escenario	Descripción	Respuesta del sistema	Flujo central
EP: 1 Mostrar el ícono de Accesibilidad.	Se muestra el ícono de Accesibilidad en el Panel de menú.	Se debe mostrar en la parte derecha del Panel de menú el ícono de Accesibilidad.	<ul style="list-style-type: none"> ➤ Se debe observar antes de iniciar la sesión en la parte derecha del Panel de menú.
		Se debe desplegar el menú con las opciones de configuración de Accesibilidad.	<ul style="list-style-type: none"> ➤ Se da clic en el mismo para desplegar el menú con las opciones de configuración. ➤ Se modifica alguna de las opciones para confirmar si el menú funciona correctamente.

Tabla 15. Diseño de caso de prueba de Modificar el ícono Accesibilidad del sistema.

Diseño de caso de prueba de Agregar los botones minimizar y maximizar.

Escenario	Descripción	Respuesta del sistema	Flujo central
EP: 1 Mostrar los botones minimizar y maximizar.	Se muestran los botones minimizar y maximizar en la barra de título.	Se debe mostrar en la parte derecha de la barra de título los botones y deben funcionar correctamente.	<ul style="list-style-type: none"> ➤ Se debe ejecutar cualquier aplicación. ➤ Se debe observar en la parte derecha de la barra de título junto al botón de cerrar los botones minimizar y maximizar. ➤ Se presionan los botones para verificar que funcionan correctamente.

Tabla 16. Diseño de caso de prueba de Agregar los botones minimizar y maximizar.

Diseño de caso de prueba Modificar el Alt+Tab.

Escenario	Descripción	Respuesta del sistema	Flujo central
EP: 1 Alt+Tab	Se muestran las ventanas de las aplicaciones que se están ejecutando en la misma área de trabajo.	Las ventanas deben girar en el sentido de las manecillas del reloj hasta llegar a la ventana que se desea visualizar.	<ul style="list-style-type: none"> ➤ Se deben ejecutar varias aplicaciones. ➤ Se presionan las teclas Alt+Tab hasta llegar a la ventana con la que se desea interactuar.

<p>EP: 2 Shift+Alt+Tab</p>		<p>Las ventanas deben girar en el sentido contrario de las manecillas del reloj hasta llegar a la ventana que se desea visualizar.</p>	<ul style="list-style-type: none"> ➤ Se deben ejecutar varias aplicaciones. ➤ Se presionan las teclas Shift+Alt+Tab hasta llegar a la ventana con la que se desea interactuar.
<p>EP: 3 Ctrl+Alt+Tab</p>	<p>Se muestran las ventanas de las aplicaciones que se están ejecutando en todas las áreas de trabajo.</p>	<p>Las ventanas deben girar en el sentido de las manecillas del reloj hasta llegar a la ventana que se desea visualizar.</p>	<ul style="list-style-type: none"> ➤ Se deben ejecutar varias aplicaciones en diferentes áreas de trabajo. ➤ Se presionan las teclas Ctrl+Alt+Tab hasta llegar a la ventana con la que se desea interactuar.
<p>EP: 4 Shift+Ctrl+Alt+Tab</p>	<p>Se muestran las ventanas de las aplicaciones que se están ejecutando en todas las áreas de trabajo.</p>	<p>Las ventanas deben girar en el sentido contrario de las manecillas del reloj hasta llegar a la ventana que se desea visualizar.</p>	<ul style="list-style-type: none"> ➤ Se deben ejecutar varias aplicaciones en diferentes áreas de trabajo. ➤ Se presionan las teclas Shift+Ctrl+Alt+Tab hasta llegar a la ventana con la que se desea interactuar.
<p>EP: 5 Alt+Dead_grave¹</p>	<p>Se muestran las ventanas de una misma aplicación que se están ejecutando en diferentes áreas de trabajo.</p>	<p>Las ventanas de una misma aplicación deben girar en el sentido de las manecillas del reloj hasta llegar a la ventana que se desea visualizar.</p>	<ul style="list-style-type: none"> ➤ Se deben abrir varias ventanas de una misma aplicación en diferentes áreas de trabajo. ➤ Se presionan las teclas Alt+Dead_grave hasta llegar a la ventana con la que se desea interactuar.
<p>EP: 6 Shift+Alt+Dead_grave</p>	<p>Se muestran las ventanas de una misma aplicación que se están ejecutando en diferentes áreas de trabajo.</p>	<p>Las ventanas de una misma aplicación deben girar en el sentido contrario de las manecillas del reloj hasta llegar a la ventana que se desea visualizar.</p>	<ul style="list-style-type: none"> ➤ Se deben abrir varias ventanas de una misma aplicación en diferentes áreas de trabajo. ➤ Se presionan las teclas Shift+Alt+Dead_grave hasta llegar a la ventana con la que se desea interactuar.

¹ Tecla que se posiciona debajo de la tecla Esc.

EP: 7 Ctrl+Alt+↑	Se muestran las áreas de escritorio que se están utilizando.	Se debe mover hacia arriba entre las diferentes áreas de escritorio que se estén utilizando hasta llegar a la que se quiere visualizar.	<ul style="list-style-type: none"> ➤ Se deben ejecutar varias aplicaciones en diferentes áreas de trabajo. ➤ Se presionan las teclas Ctrl+Alt+↑ hasta llegar al área de trabajo con la que se desea interactuar.
EP: 8 Ctrl+Alt+↓		Se debe mover hacia abajo entre las diferentes áreas de escritorio que se estén utilizando hasta llegar a la que se quiere visualizar.	<ul style="list-style-type: none"> ➤ Se deben ejecutar varias aplicaciones en diferentes áreas de trabajo. ➤ Se presionan las teclas Ctrl+Alt+↓ hasta llegar al área de trabajo con la que se desea interactuar.

Tabla 17. Diseño de caso de prueba Modificar el Alt+Tab.

Diseño de caso de prueba Adicionar nueva forma para agregar y eliminar las aplicaciones a Favoritos.

Escenario	Descripción	Respuesta del sistema	Flujo central
EP: 1 Agregar aplicaciones a Favoritos.	Debe agregarse la aplicación a Favoritos.	Debe aparecer en el menú Favoritos la aplicación agregada.	<ul style="list-style-type: none"> ➤ Dirigirse al menú de aplicaciones. ➤ Posicionarse encima de cualquier aplicación que muestre una estrella vacía. ➤ Dar clic encima de la estrella.
EP: 2 Eliminar aplicaciones de Favoritos.	Debe eliminarse la aplicación de Favoritos.	No debe aparecer en el menú Favoritos la aplicación que se elimina.	<ul style="list-style-type: none"> ➤ Dirigirse al menú de aplicaciones. ➤ Posicionarse encima de cualquier aplicación que muestre una estrella blanca. ➤ Dar clic encima de la estrella.

Tabla 18. Diseño de caso de prueba Adicionar nueva forma para agregar y eliminar las aplicaciones a Favoritos.

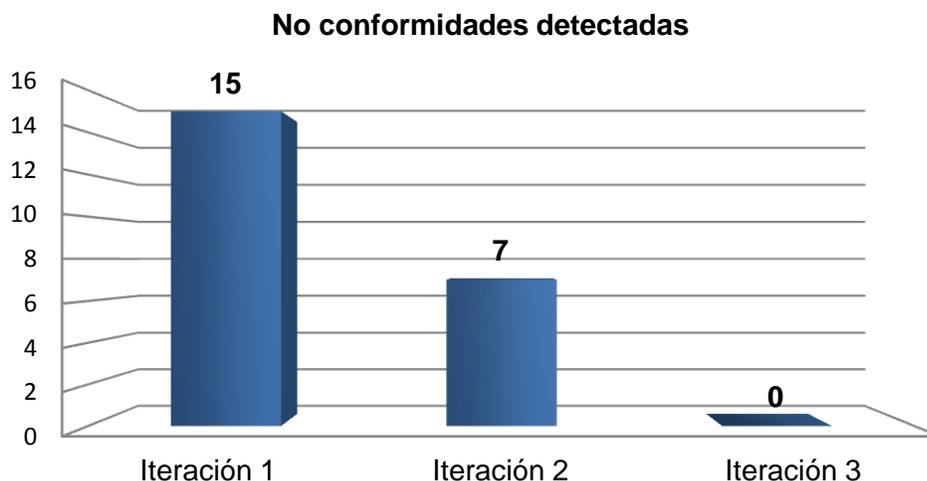


Figura 15. Gráfica representativa de los resultados de las pruebas de funcionalidad.

Resultado: El proceso de pruebas de funcionalidad se lleva a cabo en tres iteraciones y las No Conformidades que se detectan se relacionan principalmente, con la no correspondencia de lo que se muestra con lo que se programa, no se observa lo que se programa y se bloquea el sistema en varias ocasiones. En la primera iteración se detectan 15 no conformidades, que se corrigen en su totalidad. En la segunda iteración se detectan 7 no conformidades que también son corregidas y en la tercera no se detectan no conformidades. Estos resultados demuestran que las funcionalidades desarrolladas funcionan correctamente.

3.3 Pruebas de aceptación.

Las pruebas de aceptación se efectúan mediante una encuesta realizada a una muestra de 340 personas de la población que utiliza Nova Escritorio. Dicha población está formada por Directivos de la UCI, desarrolladores del Sistema Operativo Nova, profesores y asistentes de control de la Facultad 1 y parte de la comunidad HumanOS. Esta consiste en mostrarles a las personas encuestadas una vista de la versión actual del sistema y otra con los cambios realizados en el mismo, pudiendo ellos seleccionar la opción de su preferencia o estar en desacuerdo con las dos opciones (Se evidencia en la gráfica con la diferencia de valores entre el antes y el después). (Ver Anexo 2)

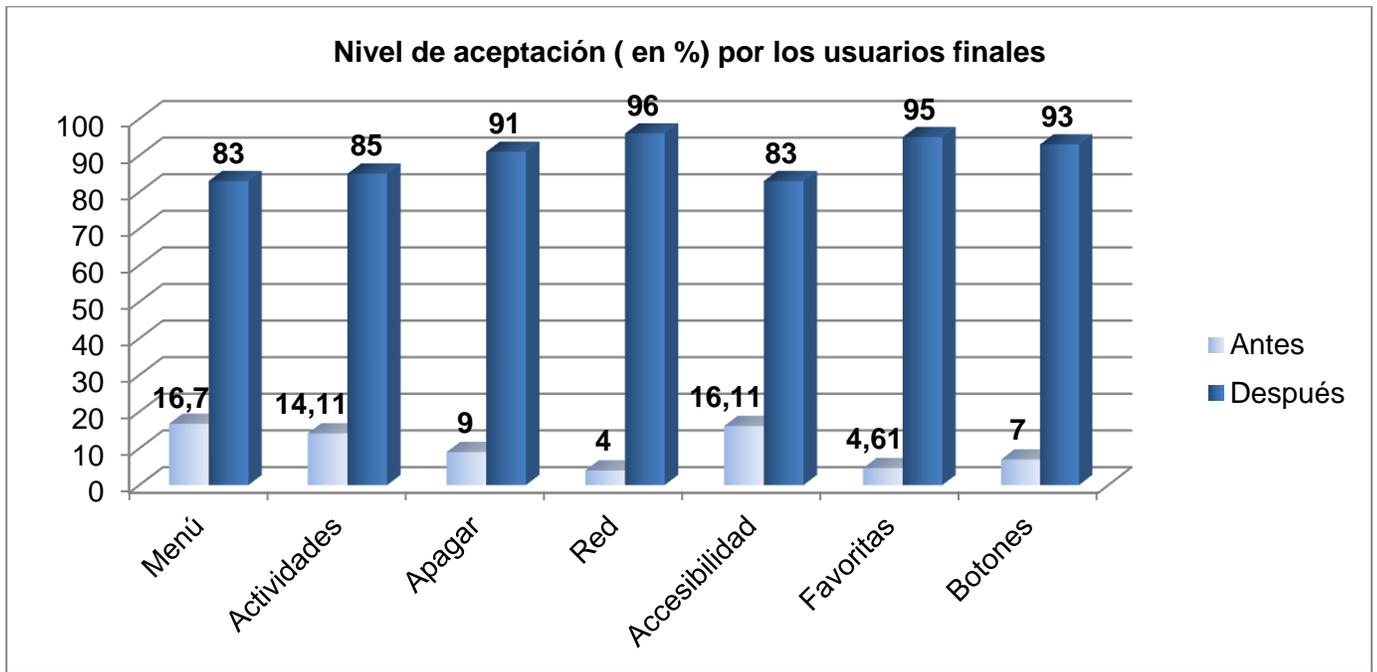


Figura 16. Gráfica representativa de los resultados de las pruebas de aceptación.

Resultados: Esta arroja como resultado que existe más de un 80% de aceptación por parte de los usuarios finales hacia cada uno de los cambios desarrollados para la AI del Entorno de escritorio de Nova Escritorio 5.0.

Conclusiones Parciales

- Las pruebas de Caja blanca y Caja negra realizadas demuestran que las funcionalidades desarrolladas están correctas y listas para ser utilizadas.
- Las pruebas de aceptación que se realizan demuestran que los cambios fueron bien acogidos por la muestra seleccionada.

- El estudio de la AI de las distribuciones más populares de los sistemas operativos evidenció que la versión actual de Nova Escritorio (4.0) es compleja en cuanto al uso y configuración, lo que trae consigo críticas y rechazo por parte de muchos de los usuarios finales.
- La definición de las necesidades de los usuarios finales y el análisis de la AI de las distribuciones más populares de los sistemas operativos permitieron identificar diecisiete requisitos nuevos a incorporar en el Entorno de escritorio de Nova Escritorio 5.0.
- La selección de las tecnologías necesarias permitió el desarrollo de los requisitos de mayor prioridad para la AI del Entorno de escritorio de Nova Escritorio 5.0.
- Las pruebas de aceptación realizadas en la muestra seleccionada, demostraron con un 89.4% de conformidad, que todas las modificaciones realizadas contribuyeron a la aceptación de la AI del Entorno de escritorio de Nova Escritorio 5.0 por parte de los usuarios finales.

- Profundizar en la AI de Nova Escritorio.
- Desarrollar los cambios propuestos de complejidad Media y Baja en la investigación para la distribución Nova Escritorio 5.0.
- Realizar la migración de los cambios desarrollados para Gnome 3.10 a Gnome 3.12.

- AGUILAR, C. 2011. Tipos y versiones de Windows. [en línea]. [Consulta: 14 enero 2014]. Disponible en: <http://tuwindows.com/tipos-y-versiones-de-windows.html>.
- BODNAR, L. 2011. DistroWatch. [en línea]. [Consulta: 22 febrero 2014]. Disponible en: <http://distrowatch.com/dwres.php?resource=major&language=ES>.
- BUSTAMANTE, J. 2003. La arquitectura de la información del siglo XX al XXI. El profesional de la información [en línea]. Vol. 12, no. 2, pp. 110–112. [Consulta: 12 noviembre 2013]. Disponible en: <http://elprofesionaldelainformacion.metapress.com/index/58va7wmf1p60ev0m.pdf>.
- COMMUNITY FOUNDATION, G. 2013. Informática Básica - Microsoft Windows y Apple Mac OS X. [en línea]. [Consulta: 22 enero 2014]. Disponible en: http://gcfaprendelibre.org/tecnologia/curso/informatica_basica/empezando_a_usar_un_computador/8.do.
- FUENTES RODRÍGUEZ, J.M. 2012. ENTORNO DE ESCRITORIO DE NOVA 4.0. S.l.: UCI.
- GONZALEZ, D., 2011. Programacion gráfica en entornos GNOME usando la biblioteca GTK + 2.0. 2011. S.l.: s.n.
- GOOGLE 2014. Acerca de los gráficos de Tendencias de búsqueda - Ayuda de Tendencias de búsqueda de Google. [en línea]. [Consulta: 4 junio 2014]. Disponible en: <https://support.google.com/trends/answer/4355164?hl=es&rd=1>.
- MACHÍN CASTILLO, J.L. y CORSO BOUZA, A., 2010. Descripción del proceso de construcción del sistema operativo base de la distribución cubana de GNU/Linux Nova. mayo 2010. S.l.: s.n.
- MARCOS MORA, M.C. y ROVIRA FONTANALS, C., 2005. Evaluación de la usabilidad en sistemas de información web. 2005. S.l.: s.n.
- MARTINEZ FURONES, O. y FIGUEREDO AGUILAR, E. 2012. Desarrollo de una herramienta informática para la identificación y reparación de inconsistencias de estructura y datos entre bases de datos PostgreSQL [en línea]. Ciudad de la Habana: UCI. Disponible en: <http://catalogoenlinea.uci.cu/cgi-bin/koha/opac-detail.pl?biblionumber=11994>.
- MOZILLA DEVELOPER NETWORK 2013. About JavaScript - JavaScript | MDN. [en línea]. [Consulta: 23 enero 2014]. Disponible en: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript?redirectlocale=en-US&redirectslug=JavaScript%2FAbout_JavaScript.
- ORACLE 2010. Capítulo 2 Visión general del escritorio de GNOME (Escritorio de GNOME 2.0 para el sistema operativo Solaris: Guía del usuario). [en línea]. [Consulta: 27 mayo 2014]. Disponible en: <https://docs.oracle.com/cd/E19683-01/817-4012/6mjilr9p8/index.html>.
- PRESSMAN, R.S. 2001. Ingeniería de Software. Un enfoque práctico. 5ta. Madrid, España: s.n.

ROMERO, Y.V. 2013. Herramienta para la publicación de metadatos bibliográficos como datos enlazados. La Habana: Ciencias Informáticas. TD-06942-13

RONDA LEÓN, R. 2005. La Arquitectura de la Información y las Ciencias de la Información. [en línea]. [Consulta: 14 octubre 2013]. Disponible en:
http://www.nosolousabilidad.com/articulos/ai_cc_informacion.htm.

ROSENFELD, L. y MORVILLE, P. 2002. Information Architecture for the World Wide Web. S.I.: O'Reilly Media, Inc. ISBN 9780596000356.

TENZER, S.M., 2008. Organización de la información y trabajo en computadora. marzo 2008. S.I.: s.n.

THE GNOME FOUNDATION 2002. GLib™. [en línea]. [Consulta: 23 enero 2014]. Disponible en:
<http://www.calcifer.org/documentos/librognome/glib.html>.

TOUB, S. 2000. AP RACTICAL Guide TO Assessing WE B Site Organization. [en línea]. [Consulta: 12 noviembre 2013]. Disponible en:
http://edu.enredo.org/IMG/pdf/evaluating_INFORMATION_ARCHITECTURE.pdf.

TRÖGER, E. 2008. Geany. Geany: Home Page [en línea]. [Consulta: 23 enero 2014]. Disponible en:
<http://www.geany.org/>.

WIKILIBROS 2012. Programación en C/Glib. [en línea]. [Consulta: 23 enero 2014]. Disponible en:
http://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C/Glib.