

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 1



# **Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0**

Trabajo de diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

**Autor:** Fidel Alejandro Sánchez Fernández

**Tutores:**

MsC. PA. Yurisbel Vega Ortiz

Ing. Miosotis Toledo Rodríguez

Ing. Sueyaile Toledo Ruiz

**LA HABANA, 7 DE DICIEMBRE DEL 2022**

**“AÑO 64 DE LA REVOLUCIÓN”**

## DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma **Fidel Alejandro Sánchez Fernández**, CI: **97090115248** con título "**Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0**" concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los \_\_ días del mes de \_\_\_\_\_ del año 2022.

---

**Fidel Alejandro Sánchez Fernández**

Firma del Autor

---

**Ing. Miosotis Toledo Rodríguez**

Firma del Tutor

---

**Ing. Sueyaile Toledo Ruiz**

Firma del Tutor

---

**MsC. PA. Yurisbel Vega Ortiz**

Firma del Tutor

## **DATOS DE CONTACTO**

Ing. Miosotis Toledo Rodríguez

Universidad de las Ciencias Informáticas, La Habana, Cuba

Correo: mtoledo@uci.cu

Ing. Sueyaile Toledo Ruiz

Universidad de las Ciencias Informáticas, La Habana, Cuba

Correo: stoledo@uci.cu

MsC. PA. Yurisbel Vega Ortiz

Universidad de las Ciencias Informáticas, La Habana, Cuba

Correo: yurisbelv@uci.cu

## **AGRADECIMIENTOS**

En primer lugar quiero agradecer a mi familia, sin la cual no estaría hoy aquí, a mis padres y mi hermana por tanto cariño, devoción, esmero, tanto ánimo, tantos regaños, consejos y perseverancia que me ayudó a lograr este título, los amo con la vida. A mami, quien me enseñó a leer y escribir, sin ella no estaría leyendo estas líneas, mi viejita bella. A mi novia, por tanto apoyo e insistencia y por ser mi refugio entre tanto estrés. A mis amigos, los HP por los increíbles momentos y recuerdos juntos en la uci, nunca los olvidaré y ojalá nos encontremos de nuevo como siempre dijimos, a Jennifer mi madrina, por pelearme tanto para que estudiara. A mis compañeros de aula Hanier, Heidi, Daylin por las horas de estudio y por ser un equipo increíble. A Yandry, por las madrugadas tomando café y ayudándome con la programación en C, a mis profesores durante estos años de carrera, al profe Menéndez por tantas reflexiones y sus clases magistrales. A mis tutores maravillosos por las horas que me dedicaron, por tanta ayuda y paciencia, a Aldy, Juan, Yileni, Junquera por aportar su granito de arena al ayudarme con la tesis o la programación. Se que siempre me faltarán nombres de personas que de una manera u otra estuvieron presentes durante todo este proceso, a todos ustedes muchísimas gracias.

**DEDICATORIA**

A mis padres, son mi razón de ser.

## RESUMEN

Los fondos de pantalla constituyen una de las formas en las cuales el usuario interactúa con su escritorio y se sienta más cómodo a la hora de sentarse frente al ordenador, ya que, al personalizarlo a su gusto con imágenes, crea un ambiente agradable y más cercano al mismo. Los usuarios de la Distribución Cubana GNU/Linux Nova, tienen la posibilidad de elegir entre varios fondos de pantalla, sin embargo, no pueden elegir varias imágenes a la vez. Por tal motivo se hace necesario crear un componente que permita al usuario personalizar su ordenador para una mayor satisfacción en su entorno de trabajo. La presente investigación tuvo como objetivo desarrollar un componente para configurar los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0, con el fin de facilitar la configuración completa de su fondo de pantalla y lograr una experiencia. La metodología de desarrollo de software empleada fue la Variación de AUP para la UCI, se utilizaron las herramientas Visual Paradigm, Glade, el Lenguaje de Modelado Unificado, editor de código fuente Visual Studio Code, la biblioteca gráfica GTK y el lenguaje de programación C. Además se tuvo en cuenta la arquitectura 3-Capas. Con las pruebas aplicadas se comprobó el correcto funcionamiento del Componente, demostrando que el mismo satisface las necesidades del cliente. Se desarrolló un componente que configura los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0 permitiendo seleccionar una carpeta de fotos del usuario y configurarla como transición de diapositivas, cumpliendo las restricciones del diseño elaborado.

**Palabras Clave:** componente, fondo de pantalla, GNU/Linux, Nova.

## ÍNDICE

INTRODUCCIÓN .....	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA SOBRE EL PROCESO DE CONFIGURACIÓN DE LOS FONDOS DE PANTALLA.....	6
Introducción .....	6
I.1.1 Componente informático .....	6
I.1.2 Centro de Control .....	6
I.1.3 Fondo de Pantalla .....	6
I.2 Configuración de los fondos de pantalla en los Entornos de Escritorio.....	7
I.2.1 Comparación de homólogos .....	9
I.2.2 Aplicaciones para la configuración de fondos de pantalla en GNU/Linux .....	9
I.3 Configuración de fondos de pantalla de la Distribución Cubana GNU/Linux Nova .....	10
I.4 Metodología de desarrollo de software .....	11
I.4.1 Metodología Variación de AUP para la UCI .....	12
I.5 Herramientas y lenguajes para el modelado.....	13
Lenguaje de Modelado Unificado .....	13
I.5.1 Herramienta Visual Paradigm 8.0 para UML.....	13
I.5.2 Herramienta Glade .....	13
I.6 Herramientas y lenguajes de implementación: GTK.....	14
I.6.1 Lenguajes de Programación .....	14
I.6.1.1 Lenguaje de programación C .....	15
I.6.2 Editor de código fuente.....	15
I.6.2.1 Visual Studio Code .....	15
I.7 Herramienta de control de versiones .....	15
Conclusiones del capítulo.....	16
CAPÍTULO II: ANÁLISIS Y DISEÑO DEL COMPONENTE PARA LA CONFIGURACIÓN DE LOS FONDOS DE PANTALLA.....	17
Introducción .....	17
II.1 Descripción del contexto de la propuesta de solución desarrollada .....	17
II.1.1 Modelo conceptual .....	17
II.2 Requisitos .....	19
II.2.1 Fuentes para la obtención de requisitos .....	19
II.2.2 Técnicas de identificación de requisitos.....	19
II.2.3 Especificación de requisitos de software .....	21
II.2.4 Descripción de requisitos de software mediante Historias de Usuario .....	23
II.2.5 Validación de requisitos de software .....	25
II.2.6 Análisis y diseño .....	26
II.2.7 Diseño de clases .....	27
II.2.8 Diseño arquitectónico .....	30
Conclusiones del capítulo.....	32
CAPÍTULO III: IMPLEMENTACIÓN Y EVALUACIÓN DEL COMPONENTE PARA LA CONFIGURACIÓN DE LOS FONDOS DE PANTALLA .....	33
Introducción .....	33

III.1 Implementación .....	33
III.1.1 Estándares de codificación .....	33
III.2 Diagrama de componentes .....	35
III.2.1 Diagrama de Despliegue .....	36
III.3 Ejemplo de interfaz gráfica de usuario .....	37
III.3.1 Pruebas de software.....	38
III.3.1.1 Tipos de pruebas de software .....	38
III.3.1.2 Métodos de prueba.....	38
III.3.1.3 Técnicas de prueba .....	39
III.3.1.4 Aplicación de las pruebas .....	40
III.3.2 Evaluación de la satisfacción del cliente sobre la propuesta de solución.....	46
Conclusiones del capítulo.....	49
CONCLUSIONES GENERALES.....	50
RECOMENDACIONES .....	51
REFERENCIAS BIBLIOGRÁFICAS.....	52
ANEXOS .....	55



## ÍNDICE DE FIGURAS

Figura 1 Modelo conceptual .....	18
Figura 2 Prototipo de interfaz de usuario correspondiente al RF1. Definir el tiempo de la transición de diapositivas .....	24
Figura 3 Diagrama de clases del diseño correspondiente al RF2. Seleccionar carpeta de fotos de los archivos del usuario.....	27
Figura 4 Patrones de diseño GRASP .....	28
Figura 5 Patrón del diseño GRASP: Creador .....	29
Figura 6 Patrón arquitectónico 3-Capas.....	31
Figura 7 Estándares de codificación .....	35
Figura 8 Diagrama de componentes.....	36
Figura 9 Diagrama de Despliegue.....	37
Figura 10 Interfaz gráfica de usuario para configuración de los fondos de pantalla .....	38
Figura 11 Función get_fullname() correspondiente al RF3 Seleccionar carpeta de fotos de los archivos del usuario. ....	42
Figura 12 Grafo de flujo .....	42
Figura 13 Resultados de las pruebas funcionales .....	45

## ÍNDICE DE TABLAS

Tabla 1 Comparación entre los Entornos de Escritorios estudiados .....	9
Tabla 2 Listado de requisitos funcionales de la propuesta de solución .....	21
Tabla 3 Listado de requisitos no funcionales de la propuesta de solución .....	23
Tabla 4 Descripción de la historia de usuario del RF.1 Definir el tiempo de la transición de diapositivas .....	23
Tabla 5 Diseño de caso de prueba correspondiente al RF1. Definir el tiempo de la transición de diapositivas. ....	25
Tabla 6 Descripción de las variables correspondiente al caso de prueba del RF1. Definir el tiempo de la transición de diapositivas .....	26
Tabla 7 Caso de prueba para el camino 2 del método get_fullname().....	43
Tabla 8 Diseño de caso de prueba para la Prueba de Integración .....	46
Tabla 9 Cuadro lógico de ladov.....	47
Tabla 10 Resultados obtenidos de los encuestados .....	48

## **INTRODUCCIÓN**

Un fondo de pantalla es una imagen que suele tener formato JPG<sup>1</sup>, PNG<sup>2</sup> o GIF<sup>3</sup> y se utiliza para darle un aspecto agradable a la interfaz de cualquier dispositivo inteligente. Se ubica en el fondo de los diferentes iconos que dan personalidad y funcionalidad al ordenador. Anteriormente solo las computadoras tenían esta configuración, no obstante, el nacimiento de nuevos mecanismos tecnológicos y plataformas le dieron mayor integración a esta configuración gráfica permitiendo darle personalidad propia y crear un ambiente grato dentro de la interfaz gráfica [1].

El término entorno de escritorio comprende un conjunto de aplicaciones que combinadas ofrecen al usuario del ordenador una interfaz gráfica similar a un escritorio tradicional. Los entornos de escritorio contemporáneos suelen extender esta interfaz con otros componentes como paneles, menús, áreas de notificaciones, fondos de pantallas y otros que a pesar de no estar comprendidos dentro del concepto original de un escritorio aportan nuevas funcionalidades al sistema. Constituye, para el usuario, un medio de acceso a las diferentes funcionalidades y datos en el ordenador. Permite la gestión de aplicaciones y su integración con el resto del sistema y utiliza Componentes de hardware como: dispositivos de almacenamiento externos, periféricos, dispositivos de interconexión y otros desde el espacio de usuario<sup>4</sup> [2]. El elemento visual más importante de un entorno de escritorio es su papel tapiz o fondo de pantalla. Este puede ser configurado fácilmente en la mayoría de las distribuciones GNU/Linux [3].

Existen diferentes entornos de escritorio en los sistemas Linux, entre los que se encuentran

---

<sup>1</sup> JPG: Joint Photographic Experts Group

<sup>2</sup> PNG: Portable Network Graphics (Gráficos de Red Portátiles)

<sup>3</sup>GIF: Graphics Interchange Format (Formato de Intercambio de Gráficos)

<sup>4</sup> Espacio de usuario: espacio de aplicación externo al núcleo (aplicaciones utilizadas por el usuario final).

GNOME<sup>5</sup>, KDE<sup>6</sup>, XFCE<sup>7</sup>, LXDE<sup>8</sup>, por mencionar algunos. En general, cada entorno de escritorio se distingue por su aspecto y comportamiento particulares, aunque algunos tienden a imitar características de escritorios ya existentes [4].

Con el fin de asegurar la soberanía tecnológica e impulsar la innovación nacional, se lleva a cabo la migración paulatina hacia Software Libre y las aplicaciones de código abierto, debido a la necesidad que esto representa para la sociedad cubana en los aspectos políticos, económicos, sociales y tecnológicos. Con el objetivo de apoyar la informatización del país, en la Universidad de las Ciencias Informáticas (UCI), fue creado el Centro de Software Libre (CESOL), líder en los procesos de migración hacia tecnologías libres y de código abierto en Cuba [5].

Nova es una distribución de GNU/Linux creada para satisfacer las necesidades de la migración a plataformas de código abierto que experimenta Cuba como parte del proceso de informatización de la sociedad. Su proceso de construcción, distribución y mantenimiento está enfocado a alcanzar niveles de excelencia guiándose por cuatro principios: sostenibilidad, socio adaptabilidad, soberanía tecnológica, seguridad. El modelo de desarrollo colaborativo y el acceso al código utilizado permite un proceso flexible y versátil, en constante innovación y consonancia con las nuevas tendencias tecnológicas internacionales [6].

Nova Escritorio es una variante de Nova diseñado para el trabajo en estaciones de trabajo con prestaciones relativamente modernas, por lo que, haciendo uso de las posibilidades gráficas de las computadoras, puede brindar un ambiente que aproveche las ventajas de las nuevas tendencias en la experiencia de usuario. Propone un ambiente de usuario limpio y libre de distracciones, enfocado en aumentar la productividad del usuario, exhibiendo un entorno familiar a usuarios de Windows con el objetivo de reducir la resistencia al cambio en el proceso de migración [7].

La Distribución Cubana GNU/Linux Nova en sus versiones anteriores permitía la configuración de los fondos de pantalla, tanto de escritorio como la pantalla de bloqueo, actualmente las

---

<sup>5</sup>GNOME: GNU Network Object Model Environment

<sup>6</sup>KDE: KDesktop Environment

<sup>7</sup>XFCE: XForms Common Environment

<sup>8</sup>LXDE: Lightweight X11 Desktop Environment

únicas opciones de configuración que ofrece son las siguientes:

- Seleccionar una imagen de la biblioteca de fondos predeterminados de Nova.
- Añadir una imagen de los archivos del usuario.
- Seleccionar como fondo de pantalla la imagen añadida.

Con el objetivo de medir el índice de satisfacción de los usuarios respecto al grado de personalización y configuración de la Distribución Cubana GNU/Linux Nova 8.0, se aplicó una encuesta a 15 usuarios lo que permite recopilar información para la investigación. Como resultado se obtuvo que, el 70% de ellos se encuentra inconforme con la configuración actual de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0. La mayor petición de dichos usuarios es el poder añadir varios fondos de pantalla a la vez al ordenador. La ausencia de esta funcionalidad conlleva en muchos casos que hagan uso de programas de terceros como Variety y Wallch, lo cual no es óptimo, debido a estos programas no vienen instalados con el sistema y requieren de un usuario avanzado, con conocimientos de comandos de Linux para su instalación.

De acuerdo a lo planteado previamente, se ha identificado como **problema de investigación** ¿Cómo contribuir a la configuración de los fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova 8.0? Para dar respuesta a la problemática expuesta se asume como **objeto de estudio**: el proceso de configuración de fondos de pantalla en los Entornos de Escritorio, definiendo el **campo de acción**: el proceso de configuración de los fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova 8.0. Para brindarle una solución efectiva al problema, se plantea como **objetivo general**: desarrollar un componente que permita mejorar la configuración de los fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova 8.0.

A partir del objetivo general se determinan los **objetivos específicos** siguientes:

1. Elaborar el marco teórico de la investigación sobre el proceso de configuración de los fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova.
2. Diseñar un componente que permita mejorar la configuración de los fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova 8.0.
3. Implementar un componente que permita mejorar la configuración de los fondos de pantalla

en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova 8.0.

4. Evaluar el componente que permita mejorar la configuración de los fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova 8.0.

Para dar cumplimiento a los objetivos específicos planteados se definen las siguientes preguntas científicas:

1. ¿Cuáles son los fundamentos teóricos que sustentan la investigación sobre el proceso de configuración de los fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova?
2. ¿Cuáles son los aspectos a tener en cuenta para realizar el diseño de un componente para la configuración de los fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova 8.0?
3. ¿Qué elementos son necesarios implementar en el desarrollo de un componente para la configuración de los fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova 8.0?
4. ¿Qué pruebas de software aplicar para la evaluación del componente para la configuración de los fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova 8.0?

Para el desarrollo de esta investigación se utilizarán los siguientes **métodos de investigación científica**:

#### **Métodos teóricos**

- Analítico - Sintético: permite descomponer el problema en sus partes para poder realizar un mejor análisis, tomando información de distintas fuentes bibliográficas y la extracción de los elementos más importantes que se relacionan con el proceso de configuración de fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova 8.0.
- Inductivo-Deductivo: permite integrar los métodos y funcionalidades, así como sus patrones de diseño para resolver problemas particulares de la solución en desarrollo, facilitando el análisis de los elementos generales a elementos más específicos.
- Histórico-Lógico: permite hacer un análisis de la evolución de las Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova, seleccionando las más recientes, para así obtener

las características e información necesaria para el diseño de la propuesta de solución.

### **Métodos Empíricos**

- Entrevista: permite obtener información respecto a la configuración de los fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova 8.0, así como requisitos funcionales y características de la propuesta de solución.
- Observación: se utiliza para monitorear cómo se realiza en la actualidad la configuración de los fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova.
- Encuesta: se aplica para evaluar la propuesta de solución desarrollada y conocer el índice de satisfacción de sus usuarios potenciales.

El presente trabajo está estructurado en introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos. La estructura capitular se describe a continuación:

### **Capítulo 1: Fundamentación teórica sobre el proceso de configuración de los fondos de pantalla.**

En este capítulo se hace referencia a todos los elementos teóricos a tener en cuenta en la investigación del sistema a implementar, haciendo énfasis en la forma de configuración de fondos de pantallas actualmente. Además, se analizan mecanismos, técnicas, herramientas y lenguajes para darle solución a la problemática planteada.

### **Capítulo 2: Análisis y diseño del componente para la configuración de los fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova 8.0.**

Se hace alusión a la solución propuesta, planteándose la forma en la que se va a desarrollar la misma, así como el trabajo con las herramientas que se utilizarán para su implementación. Además, se describe el proceso ágil basado en la metodología Variación AUP para la UCI.

### **Capítulo 3: Implementación y evaluación del componente para la configuración de los fondos de pantalla.**

Este capítulo describe las tareas relacionadas con las disciplinas de implementación y pruebas de las funcionalidades.

## **CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA SOBRE EL PROCESO DE CONFIGURACIÓN DE LOS FONDOS DE PANTALLA**

### **Introducción**

A fin de facilitar la comprensión del alcance de la presente investigación, en este capítulo se realiza un estudio sobre el proceso de obtención de información asociada a la configuración de los fondos de pantallas en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova 8.0. Por último, se caracterizan la metodología, tecnologías y herramientas utilizadas durante el ciclo de desarrollo de la solución que se propone.

#### **I.1.1 Componente informático**

Un componente es una unidad modular de un programa software con interfaces y dependencias bien definidas que permiten ofertar o solicitar un conjunto de servicios o funcionalidades y es capaz de comunicarse con otros componentes [8].

#### **I.1.2 Centro de Control**

Es una interfaz gráfica que nos permite configurar diversos elementos de hardware (periféricos, gestión de energía, color, pantalla, sonido, teclado, etc.) y del sistema (notificaciones, compartir, fecha y hora, usuarios, privacidad, idioma) [9].

#### **I.1.3 Fondo de Pantalla**

Un fondo de pantalla o papel tapiz, incluye tanto el fondo de pantalla como la pantalla de bloqueo, también se conoce como escritorio, imagen de escritorio o fondo de escritorio. es una imagen que se coloca en el escritorio del entorno gráfico del sistema operativo como decoración. El fondo no está limitado a una imagen estática, existen fondos de escritorio animados e interactivos y generalmente se ajustan a la resolución de pantalla disponible. (SO)[1]. Sobre ese fondo se colocan todos los íconos del escritorio, además de los menús y otros elementos. También son empleados como imágenes en los fondos de dispositivos móviles como celulares [10].

## I.2 Configuración de los fondos de pantalla en los Entornos de Escritorio.

El Entorno de escritorio es una solución completa de interfaz gráfica de usuario, provee al usuario de iconos, barras de herramientas, aplicaciones e integración entre aplicaciones con habilidades como arrastrar y soltar que hacen un manejo más amigable del sistema operativo [4].

Entre los entornos de escritorios más utilizados se encuentran GNOME, KDE, Cinnamon, XFCE [11].

- **GNOME:** es un entorno de escritorio amigable que permite a los usuarios una configuración fácil de sus sistemas, incluye un panel (para iniciar las aplicaciones y mostrar su estado), un escritorio (donde pueden colocarse los datos y las aplicaciones), un conjunto de herramientas y aplicaciones de escritorio estándar y un conjunto de convenciones para facilitar la creación de aplicaciones que sean consistentes y colaboren unas con otras [12]. El objetivo de este entorno es crear un sistema de escritorio para el usuario final que sea completo, libre y fácil de usar, usa las bibliotecas gráficas GTK y está bajo licencia GPL<sup>9</sup> [13].

En el caso del entorno de escritorio GNOME, la configuración de los fondos de pantalla es muy pobre, permitiendo solamente elegir entre las imágenes predeterminadas del sistema y añadir imágenes del archivo del usuario.

Gnome usa un archivo XML para contener la información de configuración necesaria para ejecutar los fondos como una presentación de diapositivas. El archivo XML se puede encontrar en `/usr/share/backgrounds/contest/` y contiene toda la información que requiere Gnome. La duración se muestra en segundos (la etiqueta `<duration>`), por lo que el 1795.0 especificado para una imagen (en la etiqueta `<static>`) sería de 29 minutos y 55 segundos. Durante una transición de cinco segundos (la etiqueta `<transition>`), la imagen anterior cambia a la imagen actual. Cuando esta imagen se ha mostrado durante 1795 segundos, pasa al siguiente archivo. En el archivo XML, el último archivo vuelve al primer archivo de la lista.[14].

- **KDE:** se trata de un entorno de escritorio creado para diferentes versiones del sistema UNIX. El objetivo al crear este entorno fue el tener algo que permitiese a los usuarios no

---

<sup>9</sup>GPL: General Public License, es una licencia de derechos de autor ampliamente usada en el mundo del software libre y código abierto



solo mirar líneas de órdenes e interactuar de forma gráfica con todos los componentes del sistema. De la misma manera, se pretendía que esta, fuera una herramienta fácil de usar. Se basa en el la biblioteca gráfica Qt y actualmente está traducido en más de 108 idiomas. Este entorno se basa en la personalización, y de la misma manera que GNOME prácticamente cualquier aspecto puede ser configurado por el usuario de la manera que él desee [15].

KDE le proporciona al usuario la funcionalidad de elegir entre una imagen, un color sólido o una presentación para su fondo de pantalla, definir el posicionamiento de la misma, el orden de transición y el tiempo en horas, minutos y segundos.

➤ **Cinnamon:** es un escritorio Linux con características innovadoras avanzadas y una experiencia de usuario tradicional. La disposición del escritorio es similar a la de Gnome 2, la tecnología interna proviene de Gnome Shell y la prioridad es hacer que los usuarios se sientan cómodos con su escritorio[16]. Cinnamon ofrece un aspecto inteligente y limpio que está menos hinchado que los entornos de escritorio alternativos y se centra en la velocidad y flexibilidad[17].

Cinnamon permite elegir una o varias carpetas de fotos de los archivos del usuario y configurarlas como fondo de pantalla. Además, permite definir un tiempo determinado para la presentación de diapositivas, así como el orden de las mismas, el aspecto de la imagen y el color de fondo.

➤ **Xfce:** es un entorno de escritorio ligero para sistemas tipo UNIX. Su objetivo es ser rápido y usar pocos recursos del sistema, sin dejar de ser visualmente atractivo y fácil de usar. Xfce encarna la filosofía tradicional de UNIX de modularidad y reutilización. Está formado por una serie de componentes que proporcionan toda la funcionalidad que se puede esperar de un entorno de escritorio moderno. Se empaquetan por separado y se puede elegir entre los paquetes disponibles para crear el entorno de trabajo personal óptimo[18].

Permite elegir una imagen predeterminada del sistema, de los archivos del usuario o una carpeta de imágenes para configurar como fondo, así como un color sólido para el mismo, el estilo del fondo, y el tiempo en el que las diapositivas cambian. Además, permite definir la misma configuración para todas las áreas de trabajo (Las áreas de trabajo se refieren a la

agrupación de ventanas en su escritorio. Puede crear varias áreas de trabajo, que actúan como escritorios virtuales)[19].

### I.2.1 Comparación de homólogos

En la siguiente tabla se evidencia la comparación entre los Entornos de Escritorios estudiados:

Tabla 1 Comparación entre los Entornos de Escritorios estudiados

Fuente: elaboración propia

Criterios de análisis	Entornos de Escritorios de GNU/Linux			
	Gnome	KDE	Cinnamon	XFCE
Configurar un color sólido como fondo de pantalla	NO	SI	SI	SI
Configurar y ajustar el tamaño de la imagen de fondo	NO	SI	SI	SI
Configurar el tiempo de transición de las imágenes de fondo	NO	SI	SI	SI
Configurar una imagen de los archivos del usuario para el fondo de pantalla	SI	SI	SI	SI
Configurar una imagen de las predeterminadas del sistema como fondo	SI	SI	SI	SI
Configurar una carpeta de imágenes del usuario como fondo de pantalla	NO	SI	SI	SI
Configurar el fondo de la pantalla de bloqueo	SI	SI	SI	SI
Definir la misma configuración de fondo para todas las áreas de trabajo	NO	NO	NO	SI
Instalado por defecto en la Distribución Cubana GNU/Linux Nova 8.0	SI	NO	NO	NO

### I.2.2 Aplicaciones para la configuración de fondos de pantalla en GNU/Linux

Entre los gestores de fondos de pantalla de código abierto para GNU/Linux más populares encontramos Variety, Wallch y Shotwell [20].

- Variety es un Gestor (Administrador) de Fondos de pantalla de código abierto para Linux. Es una gran aplicación que incluye grandes características, en un pequeño tamaño y con una interfaz fácil de usar, puede utilizar imágenes locales o descargar automáticamente fondos de pantalla de varias fuentes en línea. Además, permite rotarlos en un intervalo regular, elegido por el usuario [21].
- Wallch es un cambiador de fondos de escritorio que busca facilitarle al usuario el tener una imagen bonita y elegante en su pantalla a diario. Desde las preferencias de Wallch, puede establecer que la aplicación muestre notificaciones en cada cambio de escritorio, mantener un historial de los fondos de escritorio cambiados, elegir empezar Wallch en el inicio, elegir un atajo de teclado para cambiar la imagen y muchas más opciones[22].
- Shotwell es un visor y organizador de imágenes Open Source desarrollado por el equipo de Gnome y por lo tanto adaptado a dicho escritorio, aunque esto no quiere decir que no funcione en otros escritorios. Algunas de sus funciones son rotar y recortar imágenes, eliminar los ojos rojos, ajustar niveles y balance de color [23]. Shotwell propone ser una alternativa real a gestores de imágenes más conocidos, como Picasa. Algo que destaca a este programa es la posibilidad de subir imágenes a Flickr, Facebook o Picasa. Además, permite la edición y organización de las carpetas a importar. Cuenta con varios idiomas disponibles, entre ellos el español[24].

### **I.3 Configuración de fondos de pantalla de la Distribución Cubana GNU/Linux Nova**

En un estudio realizado a la configuración de fondos de pantalla de la Distribución Cubana GNU/Linux Nova en sus versiones 6.0, 7.0 y 8.0 se observaron los siguientes aspectos:

#### **Nova 6.0:**

- Permite la configuración del fondo de pantalla y de la pantalla de bloqueo
- Permite seleccionar imágenes predeterminadas que proporciona el sistema operativo
- Permite añadir imágenes del usuario
- Permite seleccionar un color para el fondo.

#### **Nova 7.0:**

- Permite la configuración del fondo de pantalla y de la pantalla de bloqueo
- Permite seleccionar imágenes predeterminadas que proporciona el sistema operativo, seleccionando una imagen a la vez para el fondo

- Permite añadir imágenes del usuario, seleccionando una imagen a la vez para el fondo
- Permite seleccionar un color para el fondo.

#### **Nova 8.0:**

- Permite seleccionar imágenes predeterminadas que proporciona el sistema operativo, seleccionando una imagen a la vez para el fondo
- Permite añadir imágenes del usuario, seleccionando una imagen a la vez para el fondo

A partir del estudio realizado a las tres últimas versiones de la Distribución Cubana GNU/Linux Nova se demuestra la falta de opciones de configuración de los fondos de pantalla a medida que se renuevan las versiones, donde ninguna de las versiones analizadas de la distribución permite la configuración de los fondos de pantalla seleccionando varias imágenes como fondo. Utiliza como entorno de escritorio GNOME por defecto que, de los entornos de escritorios analizados es el único que no permite la configuración de los fondos de pantallas de manera simple. Además, se investiga la opción del archivo .xml y la aplicación Shotwell, incorporadas con GNOME, así como el uso de aplicaciones como Variety y Wallch. Sin embargo, estas opciones no son óptimas debido a que tanto los usuarios de GNU/Linux como los que migran de Microsoft Windows están acostumbrados a configurar su fondo de pantalla mediante el Centro de Control ubicado en su escritorio y para la instalación de las aplicaciones de terceros requieren de conocimientos avanzados de comandos de GNU/Linux.

El autor de la investigación propone la implementación de un componente integrado al Centro de Control del propio sistema operativo, de manera que al acceder a la configuración del escritorio de la Distribución Cubana GNU/Linux Nova 8.0, permita seleccionar la carpeta del usuario donde estarán contenidos los fondos de pantalla, elegir la frecuencia con la que se realicen las transiciones y el tamaño de las imágenes, prescindiendo así del uso de aplicaciones para la configuración de los fondos de pantalla.

#### **I.4 Metodología de desarrollo de software**

Una metodología de desarrollo de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de un sistema informático. Es un conjunto de procedimientos, técnicas, artefactos y herramientas que guían a los desarrolladores en la realización de software[25]. Está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de software a elegir las técnicas más

apropiadas en cada momento del proyecto; así como a planificarlo, gestionarlo, controlarlo y evaluarlo[26].

#### **I.4.1 Metodología Variación de AUP para la UCI**

La metodología de desarrollo de software Variación AUP para la UCI es una variación de AUP (Agile Unified Process, Proceso Unificado Ágil) que logra estandarizar el proceso de desarrollo de software en los proyectos productivos de la universidad, además de convertirse en la metodología rectora de su desarrollo productivo[27].

Esta metodología cuenta con 3 fases:

- Inicio: durante esta fase se efectúan las actividades relacionadas con la planeación del proyecto. Se realiza un estudio inicial de la organización o cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
- Ejecución: en esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto teniendo en cuenta los requisitos y la arquitectura. Durante el desarrollo se obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
- Cierre: a través de esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

La presente investigación se desarrolla en la fase de Ejecución, pues la misma propicia que se ejecuten las actividades necesarias para realizar el software; además aborda las disciplinas: Requisitos, Análisis y diseño, Implementación, Pruebas internas y Pruebas de aceptación (RODRÍGUEZ, 2015). De los 4 escenarios que propone la metodología para la descripción de requisitos se emplea el cuarto ya que se aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una HU no debe poseer demasiada información[27].

## **I.5 Herramientas y lenguajes para el modelado**

### **Lenguaje de Modelado Unificado**

El Lenguaje de Modelado Unificado (UML<sup>10</sup>), por sus siglas en inglés ayuda a especificar, visualizar y documentar modelos de sistemas de software, incluida su estructura y diseño, de una manera que cumpla con todos estos requisitos. Se puede usar UML para el modelado de negocios y el modelado de otros sistemas que no son de software. Usando cualquiera de las herramientas basadas en UML en el mercado, se puede analizar los requisitos de su futura aplicación y diseñar una solución que los cumpla, representar los resultados utilizando el diagrama estándar de UML 2.0[28]. Se utilizó la versión 2.5.1.

#### **I.5.1 Herramienta Visual Paradigm 8.0 para UML**

Visual Paradigm es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Presenta todos los diagramas UML y herramientas ERD esencialmente en el diseño de sistemas y bases de datos. Las innovadoras herramientas de modelado como Resource Catalog, Transitor y Nicknamer hacen que el modelado de sistemas sea fácil y rentable. Se caracteriza por su disponibilidad en múltiples plataformas (Windows, Linux), un diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad, el uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación, capacidades de ingeniería directa e inversa, modelo y código que permanece sincronizado en todo el ciclo de desarrollo, así como ser de licencia gratuita y comercial[29]. Se utilizó la herramienta en su versión 16.3.

#### **I.5.2 Herramienta Glade**

Glade es una herramienta RAD<sup>11</sup> (Desarrollo Rápido de Aplicaciones), que permite el desarrollo rápido y sencillo de interfaces de usuario para el kit de herramientas GTK y el entorno de escritorio GNOME.

---

<sup>10</sup> UML: Unified Modeling Language

<sup>11</sup>RAD: Rapid Application Development

Las interfaces de usuario diseñadas en Glade se guardan como XML y, mediante el uso del objeto GtkBuilder GTK, las aplicaciones pueden cargarlas dinámicamente según sea necesario. Los archivos XML se pueden usar en numerosos lenguajes de programación, incluidos C, C ++, C #, Vala, Java, Perl, Python y otros. Glade es software libre publicado bajo la licencia GNU GPL[30]. Se utilizó la herramienta en su versión 3.22.2.

## **I.6 Herramientas y lenguajes de implementación: GTK**

GTK es un proyecto gratuito y de código abierto mantenido por GNOME y una comunidad activa de colaboradores. Se publica bajo los términos de la Licencia pública general reducida de GNU. GTK tiene todas las características que debe tener un kit de herramientas de widgets. Estas características lo convierten en el conjunto de herramientas más confiable para desarrollar aplicaciones Linux, está escrito en C, pero ha sido diseñado para soportar una amplia gama de lenguajes como Python, JavaScript, C++, Rust y muchos más.

GTK ofrece características atractivas y un rendimiento excelente que se suma a sus aplicaciones. Tiene una colección completa de widgets<sup>12</sup> básicos como Botones, Windows, Barras de herramientas para usar en su aplicación, posee una API (Application Programming Interfaces) fácil de usar que ayuda a disminuir su tiempo de desarrollo y lo ayuda a lograr mejores resultados [31]. Se utilizó la biblioteca gráfica en su versión 3.6.7.

### **I.6.1 Lenguajes de Programación**

Los lenguajes de programación son un sistema estructurado de comunicación, el cual está conformado por conjuntos de símbolos, palabras claves, reglas semánticas y sintácticas que permiten el entendimiento entre un programador y una máquina. Es un lenguaje formal (o artificial, es decir, un lenguaje con reglas gramaticales bien definidas) que le proporciona a una persona, en este caso el programador, la capacidad de escribir (o programar) una serie de instrucciones o secuencias de órdenes en forma de algoritmos con el fin de controlar el comportamiento físico o lógico de un sistema informático, de manera que se puedan obtener diversas clases de datos o ejecutar determinadas tareas. A todo este conjunto de órdenes escritas mediante un lenguaje de programación se le denomina programa informático.

---

<sup>12</sup> Widgets: En términos de ingeniería del software, un widget es un componente software visible y personalizable.

### **I.6.1.1 Lenguaje de programación C**

El Lenguaje de programación C también conocido como “Lenguaje de programación de sistemas” es un sistema operativo multiplataforma, sus instrucciones son muy parecidas a otros lenguajes incluyendo sentencias como if, else, for, do y while. C es un lenguaje de alto nivel (puesto que es estructurado y posee sentencias y funciones que simplifican su funcionamiento) tenemos la posibilidad de programar a bajo nivel. Su última versión es la 18. Para simplificar el funcionamiento del lenguaje C tiene incluidas librerías de funciones que pueden ser incluidas haciendo referencia a la librería que las incluye [32]. Debido a que el Centro de Control de la Distribución Cubana GNU/Linux está implementado en este lenguaje, se emplea en la investigación para garantizar la integración de la propuesta de solución.

### **I.6.2 Editor de código fuente**

Los editores de código permiten editar código fuente en diversos lenguajes de programación y ofrecen múltiples herramientas para facilitar el trabajo y aumentar la productividad. Debemos distinguir entre editores de código e IDE. Los editores generalmente son programas ligeros, que ofrecen lo necesario para poder ser productivos y tener una experiencia de desarrollo adecuada, pero sin complicaciones. Sin embargo, los editores actuales se pueden extender tanto como se quiera, por medio de complementos que los pueden hacer llegar a ser tan avanzados como los IDE[33].

#### **I.6.2.1 Visual Studio Code**

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C++, C#, Java, Python, PHP, Go, .NET)[34]. Se utilizó la aplicación en su versión 1.67.2.

### **I.7 Herramienta de control de versiones**

Git es un sistema de control de versiones distribuido gratuito y de código abierto diseñado para manejar todo, desde proyectos pequeños hasta proyectos muy grandes, con rapidez y eficiencia. Es fácil de aprender y ocupa poco espacio con un rendimiento **ultrarrápido**[35]. El



diseño de Git resulta de la experiencia del diseñador de Linux, Linus Torvalds, manteniendo una enorme cantidad de código distribuida y gestionada por mucha gente, que incide en numerosos detalles de rendimiento, y de la necesidad de rapidez en una primera implementación[36]. Se utilizó la aplicación en su versión 15.2.

### **Conclusiones del capítulo**

El análisis del marco teórico sobre el proceso de configuración de los fondos de pantalla y el estudio de los principales conceptos asociados al problema planteado, permitió establecer las bases para el desarrollo de la investigación y conocer las características del objeto de estudio. A partir de la caracterización y comparación de algunos de los Entornos de Escritorio más usados actualmente, así como el estudio de las versiones más recientes de la Distribución Cubana GNU/Linux Nova y las aplicaciones existentes para configurar los fondos de pantalla en GNU/Linux, se demostró la necesidad de desarrollar un componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0 que se integra al sistema operativo y aproveche las mejores funcionalidades que ofrecen los sistemas analizados. En el desarrollo de la propuesta de solución se definió la utilización de la metodología de desarrollo de software Variación de AUP para la UCI, el Lenguaje de Modelado Unificado, la herramienta Visual Paradigm, la herramienta Glade, la biblioteca gráfica GTK y el lenguaje de programación C.

## **CAPÍTULO II: ANÁLISIS Y DISEÑO DEL COMPONENTE PARA LA CONFIGURACIÓN DE LOS FONDOS DE PANTALLA**

### **Introducción**

El presente capítulo tiene como objetivo fundamental describir la propuesta de solución y las tareas de ingeniería realizadas y los productos de trabajo obtenidos durante la ejecución de las disciplinas de Requisitos, Análisis y diseño del componente para la configuración de los fondos de pantalla de la Distribución Cubana de GNU/Linux Nova 8.0. Contiene el modelo conceptual realizado para comprender el contexto del negocio a informatizar. Se definen los requisitos a través de la especificación de requisitos de software, su descripción mediante historias de usuario, además de realizar la validación de los mismos. Se realiza el diagrama de clases del diseño, en el que se representan las restricciones de implementación de las funcionalidades del componente a partir de los patrones del diseño. Se presenta el diseño arquitectónico definido para la propuesta de solución.

### **II.1 Descripción del contexto de la propuesta de solución desarrollada**

En la descripción de la propuesta de solución del presente epígrafe se elaboró un modelo conceptual, permitiendo una mejor comprensión del funcionamiento del negocio a informatizar.

#### **II.1.1 Modelo conceptual**

Un modelo conceptual contiene la descripción de cómo se relacionan los conceptos que intervienen en el negocio y sirve para representar el problema de manera gráfica. Los conceptos que en este modelo aparecen, son una representación de elementos del mundo real y no de componentes de software[26]. A continuación, se presenta el modelo conceptual que corresponde a la descripción del contexto del negocio de la propuesta de solución:

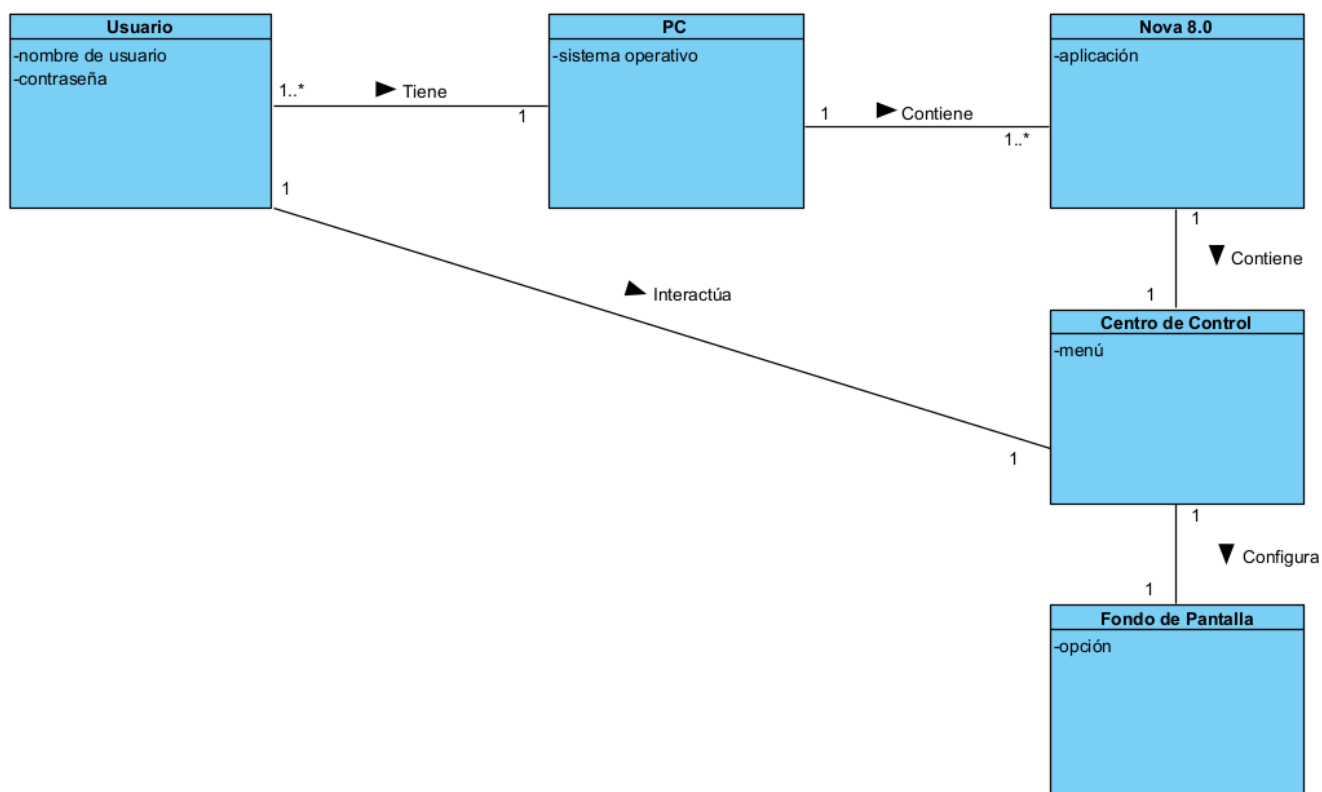


Figura 1 Modelo conceptual

Fuente: elaboración propia

Las definiciones que se muestran a continuación permiten comprender el significado que tienen los conceptos representados en el modelo conceptual del contexto del negocio informatizado.

**Conceptos del contexto del negocio:**

- **Nova 8.0:** es el sistema operativo instalado en la PC, que contiene un Centro de Control.
- **Centro de Control:** es una interfaz donde se realiza la configuración de la Distribución Cubana de GNU/Linux Nova, específicamente los fondos de pantallas.
- **Fondo de Pantalla:** es la interfaz donde se configura los fondos de pantalla por el usuario a través del Centro de Control.
- **PC:** es un ordenador que contiene la Distribución Cubana de GNU/Linux Nova en su versión 8.0, que cuenta con un Centro de Control; con este ordenador se pueden configurar los fondos de pantallas.

- **Usuario:** es el cliente final del sistema, que tiene una PC con la Distribución Cubana de GNU/Linux Nova 8.0 e interactúa con el Centro de Control para poder configurar los fondos de pantallas.

## II.2 Requisitos

El esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto. La forma de encapsular los requisitos serán las Historias de usuario (HU), agrupados en cuatro escenarios condicionados por el Modelado de negocio[27]. A continuación, se presenta las actividades desarrolladas en esta disciplina, así como los productos de trabajo elaborados en el desarrollo de la investigación.

### II.2.1 Fuentes para la obtención de requisitos

Uno de los elementos más importantes del proceso de desarrollo del software es la obtención de los requisitos, debido a que ayuda a conciliar conflictos de intereses entre los involucrados, y determinar qué tipo de software se desea desarrollar[26]. En este proceso intervienen diferentes fuentes que permiten identificar los requisitos que forman parte de una aplicación informática. Durante esta etapa de la investigación se tuvieron en cuenta como fuentes de obtención de requisitos:

- Especialistas del Centro de Software Libre (CESOL) de la Universidad de las Ciencias Informáticas (UCI).
- Aplicaciones estudiadas para la configuración de fondos de pantallas.
- Entornos de Escritorios estudiados.
- Modelo conceptual.

A continuación, se definen técnicas utilizadas en el levantamiento de requisitos del Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0:

### II.2.2 Técnicas de identificación de requisitos

En la obtención de los requisitos del componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0 se emplearon como técnicas de

identificación de requisitos: la entrevista, guía de observación, la tormenta de ideas y el desarrollo de prototipos. Su definición y forma de aplicación se describen a continuación.

- **Entrevista:** La entrevista es de gran utilidad para obtener información cualitativa como opiniones, o descripciones subjetivas de actividades. Es una técnica muy utilizada y requiere una mayor preparación y experiencia por parte del analista. La entrevista se puede definir como un “intento sistemático de recoger información de otra persona” a través de una comunicación interpersonal que se lleva a cabo por medio de una conversación estructurada[25].

La técnica se aplicó a través de una entrevista realizada a especialistas del Centro de Software Libre (CESOL) de la Universidad de las Ciencias Informáticas (UCI) (ver Anexo 1) para conocer cuáles son las características y funcionalidades necesarias para la implementación de la propuesta de solución, así como el lenguaje de programación en el que se deberá desarrollar y la compatibilidad necesaria.

- **Guía de Observación:** La técnica de observación facilita la obtención de información sobre la forma en que se efectúan las actividades. Permite observar la manera en que se llevan a cabo los procesos en una organización y verificar que realmente se sigan todos los pasos especificados en su funcionamiento[25].

La observación realizada (Anexo 2) al proceso de configuración de los fondos de pantalla en la Distribución Cubana GNU/Linux Nova 8.0, permitió conocer: los elementos fundamentales sobre los Entornos de Escritorios de GNU/Linux, las características de la Distribución Cubana GNU/Linux Nova 8.0 y las aplicaciones existentes para configurar los fondos de pantalla.

- **Tormenta de Ideas:** Consiste en reuniones entre un grupo de personas donde como primer paso sugieren toda clase de ideas sin juzgar su validez, y después de recopilar todas las ideas se realiza un análisis detallado de cada propuesta[26]. Se realizó una tormenta de ideas con los miembros del equipo de desarrollo de GNU/Linux Nova, con el objetivo de concretar cuáles son las funcionalidades que no deben faltar en la propuesta de solución, ya que se habían identificado un primer conjunto de requisitos en donde no se habían tenido en cuenta todas las necesidades a cubrir. En ella se tuvieron en cuenta todos los planteamientos realizados.

➤ **Desarrollo de prototipos:** Los prototipos suelen consistir en versiones reducidas, demos o conjuntos de pantallas (que no son totalmente operativos) de la aplicación pedida. Fomentan el desarrollo de ideas que desembocan en requerimientos, además de permitir a los usuarios mejorar las especificaciones de requerimientos. En algunos casos también se utiliza como un medio para formalizar la aceptación previa del cliente de los requisitos del proyecto[26]. Se aplicó la técnica de desarrollo de prototipos para conocer cuáles son las características que debían tener las interfaces del componente desarrollado. Los prototipos elaborados fueron revisados por miembros del equipo de desarrollo, quienes consideraron que se debía contar con estos prototipos para la representación visual de la propuesta de solución.

La aplicación de las técnicas descritas anteriormente permitió comprender con profundidad el problema de la investigación y facilitó la identificación de las funcionalidades y características de la propuesta de solución, definidas en el siguiente epígrafe.

### II.2.3 Especificación de requisitos de software

Describe el comportamiento del sistema. También tiene que estar reflejados los requisitos no funcionales en la ERS, estos imponen restricciones en el diseño o en la implementación. La ERS va enfocada tanto al equipo de desarrollo como al cliente. Para que sea comprensible por todas las partes se debe utilizar un lenguaje informal[37].

#### Requisitos Funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares[27]. A continuación, se presenta el listado de los requisitos funcionales de la propuesta de solución:

*Tabla 2 Listado de requisitos funcionales de la propuesta de solución*

*Fuente: elaboración propia*

Número	Requisito	Descripción	Complejidad
RF 1	Definir el tiempo de la transición de diapositivas	El componente permite al usuario definir el tiempo en segundos o minutos (30s, 1 min, 10min, 20min, 30min) en el	Alta

		que se realice la transición de una diapositiva a otra.	
RF 2	Seleccionar carpeta de fotos de los archivos del usuario	El componente permite seleccionar una carpeta de imágenes de los archivos del usuario y configurarla como fondo de pantalla en una transición de diapositivas.	Media
RF 3	Definir el tamaño de las imágenes	El componente permite al usuario definir la propiedad de tamaño de las imágenes seleccionadas como fondo de pantalla, pudiendo seleccionar entre varias opciones (Zoom, Centrado, Cuadrícula, Aumentado).	Alta
RF 4	Generar Transición de Diapositivas	El componente permite al usuario configurar la transición de diapositivas.	Alta

### Requisitos No Funcionales

Los requisitos no funcionales son cualidades y restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo sobre el proceso de desarrollo. Estas restricciones a menudo se aplican al sistema en su totalidad, normalmente apenas se aplican a las características o servicios individuales del sistema. La definición de los requisitos no funcionales de la propuesta de solución se realizó mediante el estándar de calidad ISO-25010<sup>13</sup>, propuesto en el producto de trabajo “Especificación de requisitos de software” del expediente de proyecto utilizado en la actividad productiva de la universidad[26]. A continuación, se presenta el listado de los requisitos no funcionales para la propuesta de solución:

*Tabla 3 Listado de requisitos no funcionales de la propuesta de solución*

*Fuente: elaboración propia*

<sup>13</sup> ISO: International Organization for Standardization, Organización Internacional de Normalización

Número	Requisito	Descripción
RNF 1	Requisito de Funcionalidad	El Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0 debe realizar todas las operaciones indicadas por el usuario: seleccionar carpeta.
RNF 2	Requisito de interoperabilidad	El Componente para la configuración de los fondos de pantalla debe ser compatible con la Distribución Cubana GNU/Linux Nova en su versión 8.0.
RNF 3	Requisito de usabilidad	El Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0 debe mostrar sus interfaces en idioma español; mostrando en sus componentes una secuencia lógica de pasos a seguir para el usuario.

Para describir las características de los requisitos funcionales definidos en el epígrafe 2.2, así como sus restricciones en el contexto del negocio informatizado, se utilizó el producto de trabajo Historia de Usuario. En el epígrafe 2.4 se presenta un ejemplo de este artefacto de la ingeniería de software.

#### II.2.4 Descripción de requisitos de software mediante Historias de Usuario

En la descripción de los requisitos se definió una Historia de Usuario por cada requisito. En la tabla 6 se muestra la descripción de la historia de usuario correspondiente al RF 1: Definir el tiempo de la transición de diapositivas.

Tabla 4 Descripción de la historia de usuario del RF.1 Definir el tiempo de la transición de diapositivas

Fuente: elaboración propia

## HISTORIA DE USUARIOS



<b>Número:</b> RF1	<b>Nombre del requisito:</b> Definir el tiempo de la transición de diapositivas
<b>Programador:</b> Fidel Alejandro Sánchez Fernández	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 25
<b>Riesgo en Desarrollo:</b> No aplica	<b>Tiempo Real:</b> 25 horas

**Descripción:**

El componente permite al usuario definir el tiempo (segundos o minutos) en el que se realice la transición de diapositivas. El usuario selecciona el tiempo en segundos (30 segundos) o minutos (entre 1 minuto y 30 min) en que se realice la transición. Si el usuario que no seleccione ninguna opción, la opción predeterminada sería 30 segundos como tiempo definido para la transición de diapositivas.

**Observaciones:**

**Prototipo elemental de interfaz gráfica de usuario:**



Figura 2 Prototipo de interfaz de usuario correspondiente al RF1. Definir el tiempo de la transición de diapositivas.

Fuente: elaboración propia

## II.2.5 Validación de requisitos de software

El objetivo de la validación de requisitos es asegurar que todos los requisitos del software que se han establecido se correspondan con las necesidades del negocio, de los clientes y de los usuarios, que se han detectado las inconsistencias, omisiones y errores y que estos han sido corregidos[26]. Para validar los requisitos obtenidos se aplicaron las técnicas de revisión técnica formal y diseños de caso de prueba, los cuales se describen a continuación.

### Diseño de casos de pruebas

Los diseños de casos de pruebas permiten crear un conjunto de entradas y salidas esperadas que sean efectivos para descubrir defectos en los programas y muestren que el sistema satisface sus requerimientos. Para su diseño, se selecciona una característica del sistema o componente que se está probando, un conjunto de entradas que ejecutan dicha característica, se documentan las salidas esperadas o rasgos de salida y donde sea posible se diseña una prueba automatizada que demuestre que las salidas reales y las esperadas son las mismas[25].

### Caso de prueba correspondiente al RF 1. Definir el tiempo de la transición de diapositivas.

**Descripción general:** El componente permite al usuario definir el tiempo en segundos o minutos (30s, 1 min, 10min, 20min, 30min) en el que se realice la transición de una diapositiva a otra.

**Precondiciones:** No aplica

Tabla 5 Diseño de caso de prueba correspondiente al RF1. Definir el tiempo de la transición de diapositivas.

Fuente: elaboración propia

Escenario	Descripción	Duración	Respuesta del Sistema	Flujo Central
EC1.1 Definir el tiempo de una transición de diapositivas	El usuario selecciona el tiempo en segundos o minutos de la casilla de opciones para configurar el tiempo de la transición de diapositivas	1 minuto	Se modifica la propiedad <duration> en el archivo gnome-slider-properties.xml y	1. Seleccionar opción de tiempo dentro de la casilla de opciones

			define el tiempo de transición		
EC1.2	No seleccionar ninguna opción	En el caso de que el usuario no seleccione ninguna de las opciones de la casilla de opciones para configurar el tiempo de la transición de diapositivas	30 segundos	Se define de manera predeterminada 30 segundos como el tiempo de transición para las diapositivas	1.No seleccionar ninguna opción

### Descripción de las variables

Tabla 6 Descripción de las variables correspondiente al caso de prueba del RF1. Definir el tiempo de la transición de diapositivas

Fuente: elaboración propia

No.	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	Duración	Campo de selección	No	En el campo de selección el usuario selecciona un solo valor

### II:2.6 Análisis y diseño

En esta disciplina los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Se modela el sistema, su forma y su arquitectura para que soporte todos los requisitos, incluyendo los requisitos no funcionales[26]. En los epígrafes siguientes se muestran las actividades desarrolladas en esta disciplina, así como los productos de trabajo elaborados en el proceso de desarrollo del componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0.

## II:2.7 Diseño de clases

En el desarrollo del diseño de clases del componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0 se tuvo en cuenta los patrones de diseño GRASP<sup>14</sup>. A continuación, se detallan los resultados obtenidos. g

### Diagrama de clases del diseño

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia y de uso [25]. A continuación, se presenta el diagrama de clases del diseño de la propuesta de solución.

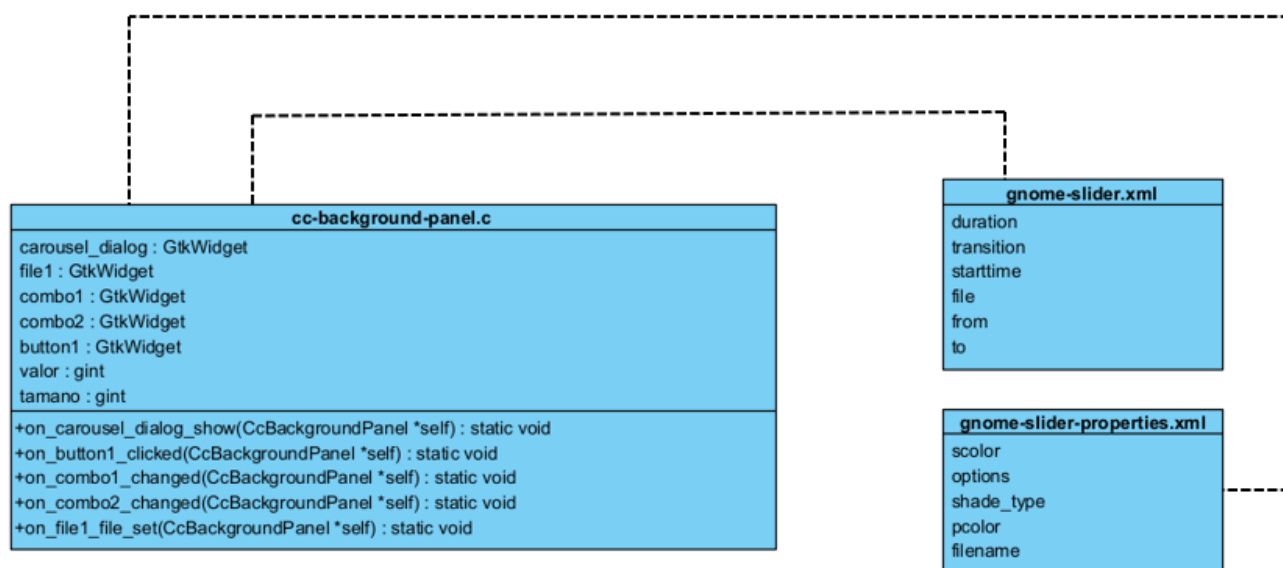


Figura 3 Diagrama de clases del diseño correspondiente al RF2. Seleccionar carpeta de fotos de los archivos del usuario

Fuente: elaboración propia

Descripción del diagrama de clases del diseño:

- **gnome-slider.xml**: es el archivo XML que contiene la estructura de datos que leerá `cc-background-panel.c` para configurar las imágenes de fondo de pantalla y su respectivo tiempo de transición, fecha de comienzo y la dirección donde se encuentran las imágenes.

<sup>14</sup>GRASP: General Responsibility Assignment Software Patterns, Patrones de Asignación de Responsabilidad

- gnome-slider-properties.xml: es el archivo XML que contiene la estructura de datos que leerá focal.xml para configurar las imágenes de fondo de pantalla según las propiedades definidas en el archivo.
- cc-background-panel.c: es un componente del Centro de Control que permite encapsular las funcionalidades necesarias para interactuar con las clases y estructuras definidas, solicita la información a los archivos XML y luego la muestra al usuario.

### Patrones de diseño

Los patrones de diseño brindan un esquema que facilita el trabajo y aportan mayor organización y claridad en la estructura de la aplicación[25]. Provee un esquema para refinar componentes de un sistema de software y la forma en que se relacionan entre sí. Describe una estructura generalmente recurrente de comunicación de componentes que resuelve un problema de diseño general dentro de un contexto particular[26]. En el diseño de la propuesta de solución se utilizaron los patrones GRASP.

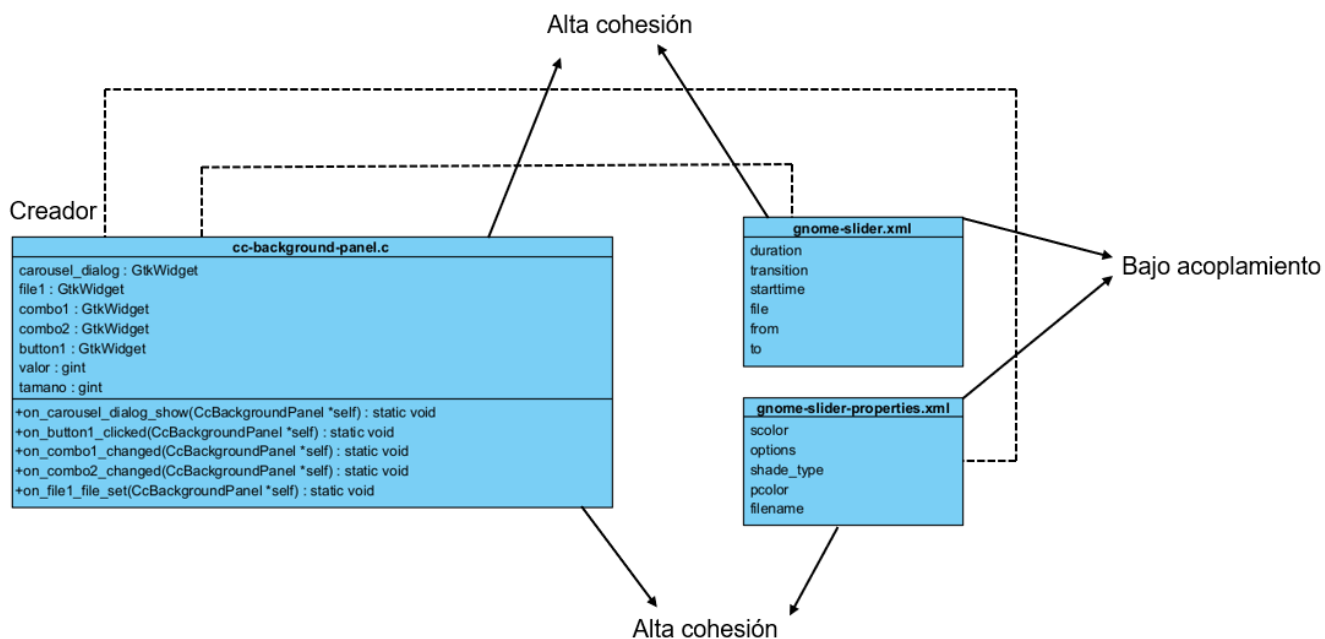


Figura 4 Patrones de diseño GRASP

Fuente: elaboración propia

### Patrones GRASP:

Los patrones GRASP describen responsabilidades que están relacionadas con las obligaciones de un objeto en cuanto a su comportamiento. Entre estos patrones se encuentran el experto, el creador, el controlador, bajo acoplamiento, alta cohesión. A continuación, se describen los patrones que fueron empleados en el diseño de la propuesta de solución.

**Creador:** ayuda a identificar qué clase debe ser responsable de la creación o instanciación de nuevos objetos o clases. Este patrón se evidencia en el método `cc_background_panel_class_init()` y llama al método `gtk_widget_class_bind_template_child()` para construir los widgets. En la figura 5 se presenta el uso de este patrón en estos métodos.

```
static void
cc_background_panel_class_init(CcBackgroundPanelClass *klass)
{
    GObjectClass *object_class = G_OBJECT_CLASS(klass);
    CcPanelClass *panel_class = CC_PANEL_CLASS(klass);
    GtkWidgetClass *widget_class = GTK_WIDGET_CLASS(klass);

    g_type_ensure(CC_TYPE_BACKGROUND_CHOOSER);
    g_type_ensure(CC_TYPE_BACKGROUND_PREVIEW);

    panel_class->get_help_uri = cc_background_panel_get_help_uri;

    object_class->constructed = cc_background_panel_constructed;
    object_class->dispose = cc_background_panel_dispose;
    object_class->finalize = cc_background_panel_finalize;

    gtk_widget_class_set_template_from_resource(widget_class, "/org/gnome/control-center/background/cc-background-panel.ui");

    gtk_widget_class_bind_template_child(widget_class, CcBackgroundPanel, add_picture_button);
    gtk_widget_class_bind_template_child(widget_class, CcBackgroundPanel, add_carousel_button);
    gtk_widget_class_bind_template_child(widget_class, CcBackgroundPanel, background_chooser);
    gtk_widget_class_bind_template_child(widget_class, CcBackgroundPanel, desktop_preview);
    gtk_widget_class_bind_template_child(widget_class, CcBackgroundPanel, carousel_dialog);
    gtk_widget_class_bind_template_child(widget_class, CcBackgroundPanel, file1);
    gtk_widget_class_bind_template_child(widget_class, CcBackgroundPanel, combo1);
    gtk_widget_class_bind_template_child(widget_class, CcBackgroundPanel, combo2);
    gtk_widget_class_bind_template_child(widget_class, CcBackgroundPanel, button1);
}
```

Figura 5 Patrón del diseño GRASP: Creador

Fuente: elaboración propia

**Alta cohesión:** el objetivo de este patrón es lograr la menor dependencia entre clases, de manera que las modificaciones realizadas en algunas de ellas afecten lo menos posible el funcionamiento del sistema. Asigna responsabilidades de control de flujo del sistema a clases específicas. Un ejemplo donde se evidencia este patrón es entre los archivos `cc-background-panel.c` y `gnome-slider.xml`, `gnome-slider-properties.xml` con `cc-background-panel.c`, pues al

existir una sola relación entre ambas, los cambios en una tienen poca influencia en el funcionamiento de la otra.

**Bajo acoplamiento:** a través de este patrón se asegura la necesidad de mantener el sistema con un bajo nivel de complejidad, lo que quiere decir que el sistema debe realizar solo las operaciones necesarias para cumplir una tarea dentro de un área funcional. Este patrón queda evidenciado en cada una de las clases del diseño, ya que cada una responde con las funcionalidades que debe cumplir para realizar los trabajos correspondientes a cada clase. Un ejemplo se encuentra en los archivos `gnome-slider.xml` y `gnome-slider-properties.xml` pues estos manejan todas sus responsabilidades en función de realizar la tarea del acceso a los datos en este caso de direcciones de las imágenes de fondo, así como su tiempo de transición y propiedades.

### II.2.8 Diseño arquitectónico

El diseño arquitectónico es el proceso creativo que intenta establecer una organización del sistema informático que satisfaga los requerimientos funcionales y no funcionales. Su objetivo es representar componentes que interactúen entre ellos y tener asignadas tareas específicas, ser flexible y extensible y representar las relaciones de control entre las partes[26]. En la propuesta de solución se utiliza el patrón arquitectónico N-Capas, en su variante 3-Capas, descrito a continuación.

#### Patrón arquitectónico 3-Capas

La programación por capas es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten). El diseño más utilizado actualmente es el diseño en tres niveles (o en tres capas)[38].

- La capa de Presentación es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio. En este caso está representada por el archivo `cc-background-panel.ui`, es un archivo de la interfaz de usuario que almacena datos en formato XML.

- La capa de Negocio es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. Está representada por los archivos cc-background-panel.c y cc-background-panel.h, estos contienen el código fuente desarrollado y escrito en el lenguaje de programación C y las declaraciones directas de clases, subrutinas, variables u otros identificadores, respectivamente.
- La capa de Datos es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. En esta capa se encuentran los archivos de configuración settings, que contienen toda la configuración de los argumentos(parámetros) y la configuración inicial del sistema y los gnome-slider.xml y gnome-slider-properties.xml donde se encuentra toda la configuración de los fondos de pantalla.

En la siguiente figura se evidencia la estructura de la separación en capas propuesta por el patrón 3-Capas.



Figura 6 Patrón arquitectónico 3-Capas

Fuente: ([https://youtu.be/M\\_M8T8G4Og8](https://youtu.be/M_M8T8G4Og8))



### **Conclusiones del capítulo**

El análisis y diseño del componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0 permitió identificar 4 requisitos funcionales y 3 no funcionales, de acuerdo a las necesidades del cliente. El empleo del patrón arquitectónico 3-Capas posibilitó validar la organización y la comunicación entre los diferentes componentes de la propuesta de solución. El diagrama de clases propició modelar las diferentes clases, sus atributos y relaciones y el empleo del patrón del diseño GRASP brindó una mayor calidad al software desarrollado.

## **CAPÍTULO III: IMPLEMENTACIÓN Y EVALUACIÓN DEL COMPONENTE PARA LA CONFIGURACIÓN DE LOS FONDOS DE PANTALLA**

### **Introducción**

El capítulo contiene los productos de trabajos elaborados en las disciplinas de Implementación y pruebas, así como la evaluación del índice de satisfacción grupal sobre el Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0. En él se presenta la especificación de los estándares de codificación utilizados durante la implementación de la solución; así como el diagrama de despliegue, diagrama de componentes y un ejemplo de la interfaz gráfica de usuario. Además, se realizan pruebas de software Unitarias, Funcionales e Integración con el objetivo de descubrir y corregir los posibles errores de Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0.

### **III.1 Implementación**

El objetivo de esta disciplina es construir el sistema informático, a partir de los resultados del Análisis y Diseño[26]. Una implementación es la realización de una especificación técnica o algoritmos como un programa, componente de software, u otro sistema de cómputo. También son una manifestación en el mundo real de las funciones de procesamiento y estructuras de la información. Muchas implementaciones son realizadas según una especificación o un estándar[39]. Durante la implementación se utilizaron los estándares de codificación descritos a continuación.

#### **III.1.1 Estándares de codificación**

Los estándares de codificación incorporan principios de ingeniería sólidos para la programación en sus respectivos lenguajes y forman la base de cualquier enfoque preventivo[40]. A continuación, se presentan los detalles:

##### **Indentación**

- Se utilizan 4 espacios sin caracteres de tabulación o espacios en blanco.

##### **Nomenclatura de archivos**

- Los caracteres que se pueden utilizar para nombrar archivos y directorios son[A-Z, a-z, 0-9, ., \_, -].
- Las extensiones de archivo usan letras minúsculas, como .c, .h.
- Los bloques de programa deben escribirse con sangría, el número de espacios con sangría es 4.
- Entre bloques de programa relativamente independientes y después de la descripción de la variable tiene que agregar una línea en blanco.

#### **Tabuladores o Espacios**

- Las estructuras de control deben tener un espacio entre la palabra reservada de la estructura y el signo de apertura de paréntesis para distinguir entre las llamadas de las funciones.
- Las funciones son llamadas sin espacio entre el nombre de la función, el signo del paréntesis y el primer parámetro; espacios entre cada coma por parámetro y sin espacios entre el ultimo paréntesis, el signo de paréntesis cerrado y el signo de punto y coma (;).

#### **Estándares de nomenclatura**

- Se utilizó *Snake\_Case* (palabras en minúscula y separadas por barra baja (underscore)) para los nombres de las variables y funciones.
- Se utilizó *Upper Camel Case* (primera letra de cada palabra en mayúscula) para el nombre de las clases.
- Se utilizó *Upper Snake Case* (todas las letras de cada palabra en mayúscula y separadas por barra baja (underscore)) para el nombre de los tipos de datos.

A continuación, se presenta un fragmento del código de la función `on_add_carousel_button_clicked_cb()` donde se evidencia el empleo de los estándares de codificación:

```

static void
cc_background_panel_class_init(CcBackgroundPanelClass *klass)
{
    GObjectClass *object_class = G_OBJECT_CLASS(klass);
    CcPanelClass *panel_class = CC_PANEL_CLASS(klass);
    GtkWidgetClass *widget_class = GTK_WIDGET_CLASS(klass);

    g_type_ensure(CC_TYPE_BACKGROUND_CHOOSER);
    g_type_ensure(CC_TYPE_BACKGROUND_PREVIEW);

    panel_class->get_help_uri = cc_background_panel_get_help_uri;

    object_class->constructed = cc_background_panel_constructed;
    object_class->dispose = cc_background_panel_dispose;
    object_class->finalize = cc_background_panel_finalize;
}
    
```

Cuatro espacios

Snake Case

Upper Camel Case

Upper Snake Case

Figura 7 Estándares de codificación

Fuente: elaboración propia

### III.2 Diagrama de componentes

Los diagramas de componentes representan las relaciones entre los componentes individuales del sistema mediante una vista de diseño estática, pueden ilustrar aspectos de modelado lógico y físico[41].

En la figura 8 se evidencia como fue empleado el patrón 3-Capas en el Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0:

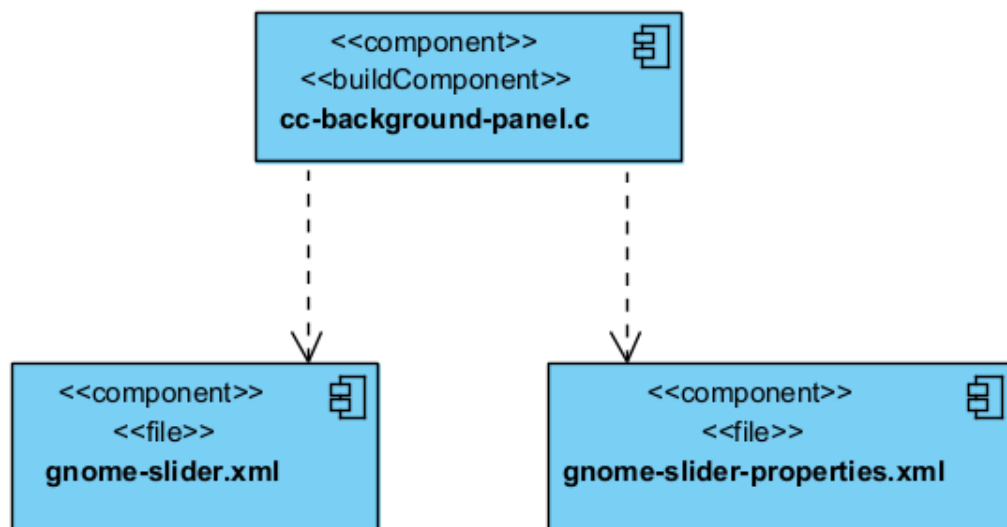


Figura 8 Diagrama de componentes

Fuente: elaboración propia

### III.2.1 Diagrama de Despliegue

El diagrama de despliegue es una estructura que muestra la arquitectura del sistema desde el punto de vista de la distribución de los artefactos del software en los destinos de despliegue. En él están representados nodos que corresponden con los dispositivos de hardware o con algún entorno de ejecución de software[26]. En la figura siguiente se presenta el diagrama de despliegue elaborado para la propuesta de solución:

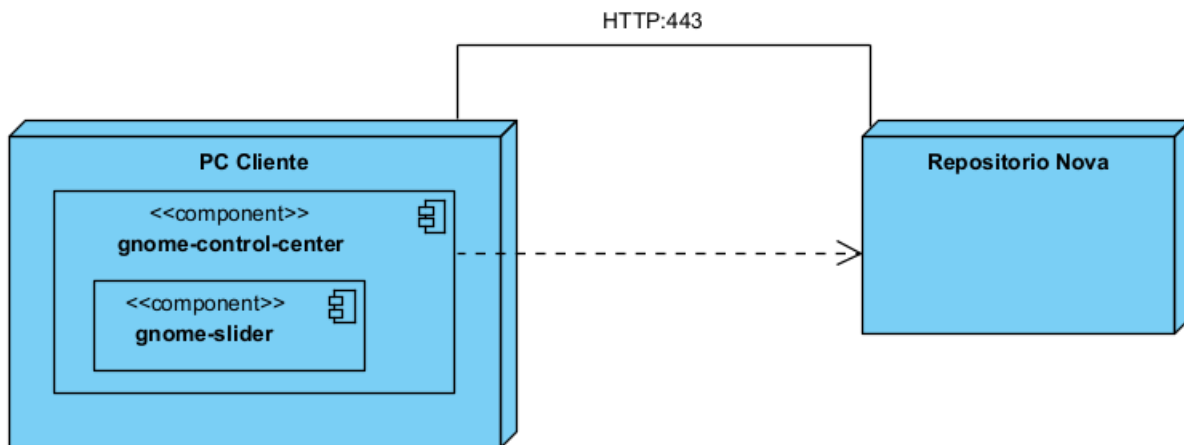


Figura 9 Diagrama de Despliegue

Fuente: elaboración propia

A continuación, se describen los nodos, componentes y protocolos de comunicación representados en el diagrama anterior.

#### Descripción de los nodos:

- PC Cliente: estación de trabajo desde la cual se accede al Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0, a través de Centro de Control.
- Repositorio Nova: servidor web que almacena los paquetes y programas de la distribución GNU/Linux Nova.

#### Descripción de los componentes:

- Centro de Control: interfaz del Entorno de Escritorio GNOME que permite configurar todo el sistema.

- gnome-slider: interfaz de la propuesta de solución que permite configurar los fondos de escritorio como una transición de diapositivas.

#### Descripción de protocolos:

- HTTP (Hypertext Transfer Protocol, Protocolo de Transferencia de Hipertexto): protocolo de transferencia de datos; utilizado en el Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0 para la transmisión de los datos entre los controladores y la vista.

### III.3 Ejemplo de interfaz gráfica de usuario

La interfaz de usuario es una parte de un programa que gestiona la interacción con el usuario basándose en relaciones visuales como iconos, menús, formularios o un puntero. Su principal objetivo es facilitar la comunicación con el ordenador[39].

El cumplimiento de estas reglas permite desarrollar interfaces flexibles, en las que el usuario no realice acciones innecesarias y no deseadas, así como garantizar la interrupción de una secuencia de acciones en el instante requerido por el usuario. Brinda la posibilidad de desglosar la información de forma progresiva a través de una estructura organizada, orientar al usuario en las tareas a desarrollar y que esas tareas se realicen en el contexto adecuado[39]. A continuación se muestra la interfaz de usuario correspondiente al Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0:

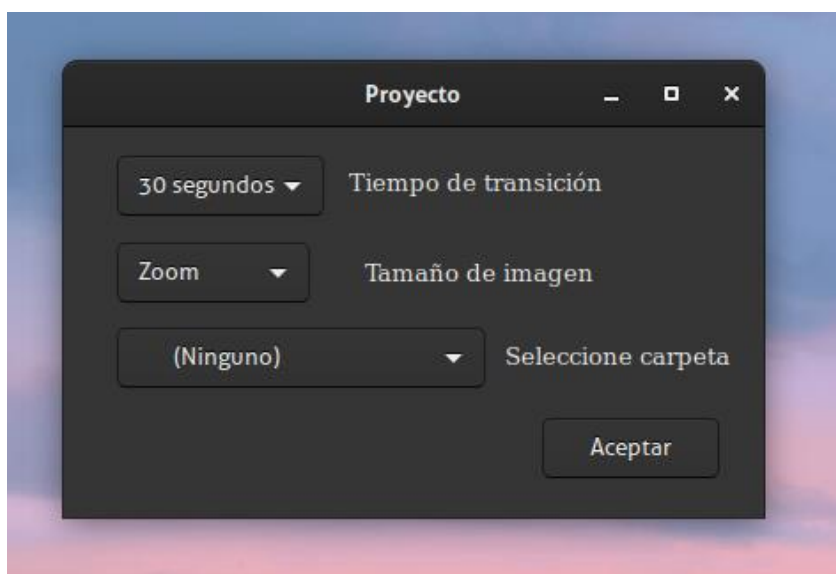


Figura 10 Interfaz gráfica de usuario para configuración de los fondos de pantalla

Fuente: elaboración propia

### **III.3.1 Pruebas de software**

Las pruebas de software son un conjunto de procesos con los que se pretende probar un sistema o aplicación en diferentes momentos para comprobar su correcto funcionamiento. Este tipo de pruebas abarca cualquier estadio del desarrollo del sistema, desde su creación hasta su puesta en producción[42].

#### **III.3.1.1 Tipos de pruebas de software**

##### **Pruebas Unitarias**

Las pruebas unitarias o de componente consisten en la ejecución de actividades que le permitan verificar al desarrollador que los componentes unitarios están codificados bajo condiciones de robustez, esto es, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada[26].

##### **Pruebas Funcionales**

Las pruebas funcionales se centran en comprobar que el sistema funcione acorde a los requisitos funcionales y no funcionales, detectando posibles defectos derivados de errores en la fase de programación; dichas pruebas se llevan a cabo a través de la interfaz gráfica de la aplicación. Mediante estas pruebas se demuestra que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, además de mantener íntegra la información externa del sistema[39].

##### **Pruebas de Integración**

Las pruebas de integración consisten en la comprobación de que los elementos del software que interactúan entre sí, funcionan de manera correcta. Uno de sus objetivos es identificar errores introducidos por la combinación de programas o componentes probados unitariamente, para asegurar que la comunicación, enlaces y los datos compartidos ocurran apropiadamente. Se diseñan para descubrir errores o completitud en las especificaciones de las interfaces[26].

#### **III.3.1.2 Métodos de prueba**

##### **Método de caja blanca**

Las pruebas de caja blanca o pruebas estructurales se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente[26]. Mediante este método de prueba, el ingeniero del software puede obtener casos de prueba que garanticen que se ejecuten por lo menos una vez todos los caminos independientes de cada módulo; que se ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa; que se realicen todos los bucles en sus límites y con sus límites operacionales; y efectúen las estructuras internas de datos para asegurar su validez[39].

### **Método de caja negra**

Las pruebas de caja negra o pruebas funcionales se aplican sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa. A través de estas pruebas se pretende demostrar que las funcionalidades del software son operativas; las entradas son aceptadas de manera correcta[26].

### **III.3.1.3 Técnicas de prueba**

#### **Camino básico**

La técnica del camino básico tiene como objetivo comprobar que cada camino se ejecute de manera independiente de un componente o programa, obteniéndose una medida de la complejidad lógica del diseño. Esta técnica debe ser utilizada para evaluar la efectividad de los métodos asociados a una clase, al tratar de confirmar que cada camino independiente sea ejecutado al menos una vez en el sistema[43].

#### **Partición de equivalencia**

La técnica de partición de equivalencia consiste en probar cada una de las funcionalidades del sistema, mediante la definición de casos de prueba que sean capaces de encontrar diferentes tipos de errores a nivel de interfaz. Es un conjunto de estados válidos y no válidos para las condiciones de entrada. Por lo general, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición booleana[39].

#### **Pruebas basadas en hilos**

Esta técnica fue aplicada durante las pruebas de integración y su objetivo es integrar el conjunto de clases requeridas, para responder una entrada o suceso al sistema. Cada hilo se integra y prueba individualmente[39].



#### **III.3.1.4 Aplicación de las pruebas**

En el presente epígrafe se presentan los resultados obtenidos de las diferentes pruebas de software anteriormente definidas y aplicadas empleando los métodos y técnicas durante la ejecución de las disciplinas Pruebas internas y Pruebas de aceptación propuestas por la metodología de desarrollo de software variación AUP para la UCI.

##### **Pruebas Internas**

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas[26]. Durante esta disciplina se aplicaron pruebas unitarias, funcionales y de integración, descritas a continuación.

##### **Pruebas unitarias**

Las pruebas unitarias se centran en probar cada componente de código de un software de forma individual para asegurar que funcione de manera apropiada como unidad. Emplean técnicas de prueba que recorren caminos específicos en la estructura de control de los componentes[25]. El método de prueba utilizado para la realización de esta prueba es caja blanca y la técnica de prueba contenida en este método que se empleó fue la técnica del camino básico.

En la figura 11 se presenta la función `get_fullname()` correspondiente al RF3 Seleccionar carpeta de fotos de los archivos del usuario.

```

char *get_fullname(char *ruta, struct dirent *ent)
{
    char *nombrecompleto;
    int tmp;
    tmp = strlen(ruta);
    nombrecompleto = malloc(tmp + strlen(ent->d_name) + 2);
    if (ruta[tmp - 1] == '/')
        sprintf(nombrecompleto, "%s%s", ruta, ent->d_name);
    else
        sprintf(nombrecompleto, "%s/%s", ruta, ent->d_name);
    return nombrecompleto;
    free(nombrecompleto);
}

```

Figura 11 Función get\_fullname() correspondiente al RF3 Seleccionar carpeta de fotos de los archivos del usuario.

Fuente: elaboración propia

### 1. Dibujar el grafo de flujo de la funcionalidad

En la figura 12 se utilizó la notación de grafo de flujo para representar el código de la funcionalidad a probar:

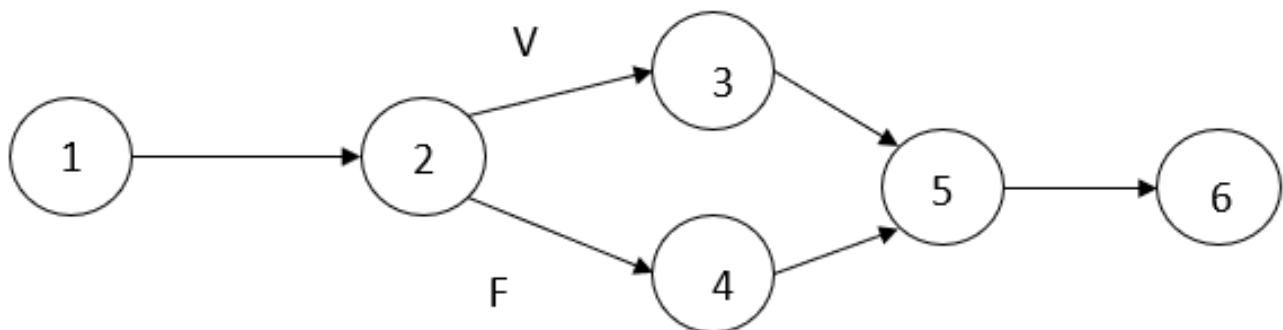


Figura 12 Grafo de flujo

Fuente: elaboración propia

## 2. Determinar la complejidad ciclomática

La complejidad ciclomática de un grafo  $V(G)$  se puede calcular de tres formas diferentes:

$V(G) = A - N + 2$  Donde A es el número de aristas del grafo de flujo y N es la cantidad de nodos del grafo

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

$$V(G) = 2$$

$V(G) = P + 1$  Donde P es el número de nodos predicados (nodos con más de una arista de salida) contenidos en el grafo

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

$$V(G) = 2$$

$V(G) = R$  Donde R es el número de regiones (áreas delimitadas por nodos y aristas en el grafo)

$$V(G) = 2$$

## 3. Determinar los caminos linealmente dependientes

Camino básico 1: 1,2,4,5,6

Camino básico 2: 1,2,3,5,6

## 4. Definir los casos de prueba para comprobar la ejecución de cada camino del conjunto básico

En el diseño de los casos de prueba se especifican los siguientes elementos:

- Descripción: contiene una descripción sobre las restricciones de los datos de entrada que debe tener el caso de prueba.
- Condición de ejecución: se especifican los parámetros que debe poseer el caso de prueba para que se cumpla una condición deseada como respuesta del funcionamiento del procedimiento.
- Entrada: se muestran los parámetros de entrada al procedimiento.
- Resultados esperados: se explica el resultado esperado de la ejecución del procedimiento

En la tabla se presenta el diseño de caso de prueba del camino 2 del conjunto de caminos básicos linealmente independientes correspondientes a la funcionalidad `get_fullname()`. En este camino se prueba que se define la estructura de la ruta de los archivos contenidos en la carpeta seleccionada por el usuario de forma exitosa.

Tabla 7 Caso de prueba para el camino 2 del método `get_fullname()`

Fuente: Elaboración propia

<b>Diseño de caso de prueba para el camino 2</b>	
<b>Descripción</b>	Define y guarda la estructura de la ruta completa de los archivos contenidos en la carpeta seleccionada
<b>Condición de ejecución</b>	El usuario selecciona una carpeta de imágenes para el fondo de pantalla
<b>Entrada</b>	ruta: /home/fide/imágenes/fondos/foto1.jpg
<b>Resultados esperados</b>	Devuelve la estructura de la ruta de los archivos contenidos en la carpeta seleccionada separando cada elemento con una barra diagonal

Tabla 8 Caso de prueba para el camino 1 del método get\_fullname()

Fuente: Elaboración propia

<b>Diseño de caso de prueba para el camino 2</b>	
<b>Descripción</b>	Define y guarda la estructura de la ruta completa de los archivos contenidos en la carpeta seleccionada
<b>Condición de ejecución</b>	El usuario selecciona una carpeta de imágenes para el fondo de pantalla
<b>Entrada</b>	ruta: /home/fide/imágenes/fondosfoto1.jpg
<b>Resultados esperados</b>	Devuelve la estructura de la ruta de los archivos contenidos en la carpeta seleccionada agregando a la estructura una barra diagonal (/) luego de la dirección del archivo y antes del nombre del mismo.

Luego de realizadas las pruebas unitarias y a partir de la aplicación de los casos de pruebas, se comprobó que el flujo de trabajo de las funcionalidades del Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0 es correcto; ya que cada sentencia fue ejecutada al menos una vez, cumpliéndose así las condiciones de las pruebas y los resultados esperados.

### Pruebas funcionales

En las pruebas funcionales se tuvieron cuenta el método de caja negra y la técnica de partición equivalente. Para la aplicación de la técnica de partición equivalente se retomaron los casos de prueba diseñados en el epígrafe 2.5 “Validación de requisitos de software” en que se definen las clases de equivalencia válidas y no válidas para cada campo de entrada del sistema. En la figura se presenta un gráfico con los resultados de las pruebas funcionales.

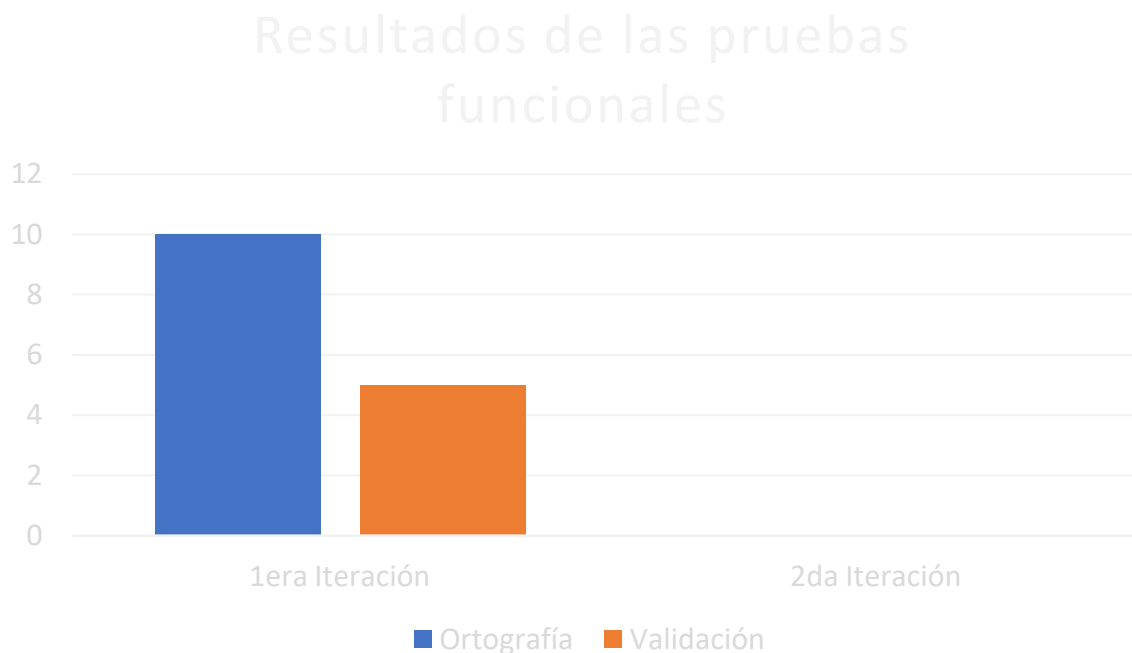


Figura 13 Resultados de las pruebas funcionales

Fuente: elaboración propia

Las pruebas funcionales se realizaron en 2 iteraciones; en la primera se identificaron 15 no conformidades de ellas 10 fueron de errores ortográficos y 5 de validación. Durante la segunda iteración no se identificaron no conformidades.

### Pruebas de Integración

Para la prueba de integración se realizó el diseño de caso de prueba presentado en la figura a continuación para comprobar la integración del Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0.

**Descripción general:** Permite integrar el Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0 al gnome-control-center (Centro de Control).

**Condiciones de ejecución:** El código fuente del Componente para la configuración de los fondos de pantalla debe encontrarse en la carpeta gnome-control-center/panels/backgrounds/ de la Distribución Cubana GNU/Linux Nova 8.0.

Tabla 9 Diseño de caso de prueba para la Prueba de Integración

Fuente: elaboración propia

Escenarios	Descripción	Respuesta del sistema	Flujo central
EC1.1 Seleccionar opción de Agregar Transición	El usuario selecciona el botón de Agregar Transición	Muestra un cuadro de diálogo con las opciones para la configuración de la transición	1. Seleccionar el botón de Agregar Transición
EC1.2 Seleccionar opción de Agregar Transición	El usuario selecciona el botón de Agregar Transición	No muestra el cuadro de diálogo con las opciones para la configuración de la transición y cierra el Centro de Control	1. Seleccionar el botón de Agregar Transición

Para integrar el Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0, al Centro de Control se debe garantizar que el código fuente se encuentre en la carpeta gnome-control-center/panels/backgrounds/ de la Distribución Cubana GNU/Linux Nova 8.0. En la tabla anterior se muestra el caso de prueba utilizado para corroborar la integración del componente al Centro de Control. Para desarrollar esta prueba se tuvieron en cuenta 2 escenarios que representan las posibles situaciones que se pueden

presentar cuando se proceda a integrar la solución desarrollada. Como resultado se obtuvo que el Componente para la configuración de los fondos de pantalla se integró de manera correcta a Nova 8.0.

### **Pruebas de aceptación**

En la disciplina de Prueba de aceptación se realizan las pruebas de software. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido[25]. El cliente realizó pruebas funcionales en las que no se detectaron no conformidades y emitió un acta de aceptación de los productos de trabajo en total conformidad con la solución desarrollada (ver Anexo 4).

### **III.3.2 Evaluación de la satisfacción del cliente sobre la propuesta de solución**

La satisfacción del cliente se ha convertido en uno de los objetivos más importantes dentro del proceso de desarrollo de software en la validación de cualquier investigación científica. La información brindada resulta muy útil para conocer las fortalezas y debilidades de la propuesta realizada[26]. Con la finalidad de determinar el índice de satisfacción personal y grupal de los usuarios sobre el Componente para la configuración de los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0, se aplicó la Técnica de V. A. Iadov.

La técnica de V.A. Iadov está conformada por cinco preguntas: cuatro cerradas y una abierta, las cuales se relacionan a través de lo que se denomina el "Cuadro Lógico de Iadov". Constituye una vía para el estudio de la satisfacción en dependencia del campo donde se aplique, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas que se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce. Estas preguntas permiten ubicar al o los encuestados en una escala de satisfacción tomando valores del 1 al 6 de la siguiente forma:

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definida
4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictoria

Para conocer el grado de satisfacción de los usuarios de la propuesta de solución se aplicó una encuesta (ver Anexo 3) a un grupo de 15 Especialistas del Centro de Software Libre. Las preguntas 1, 2 y 3 de la encuesta se relacionan a través de lo que se denomina el “Cuadro Lógico de ladov” que se muestra en la siguiente tabla.

Tabla 10 Cuadro lógico de ladov

Fuente: elaboración propia

1. ¿Podría calificar su satisfacción con respecto a su entorno de escritorio?	2. ¿Considera usted que se debe modificar la configuración de fondos de pantalla?						
	No			No sé			
	3. ¿Le gustaría agregar varias imágenes como fondo de pantalla?						
	Sí	No sé	No	Sí	No sé	No	S
Me gusta más de	1	2	6	2	6	6	
Me da	2	2	3	2	3	3	
Me gusta más	3	3	3	3	3	3	
Me da	6	3	6	3	4	4	
No me gusta más	6	6	6	6	4	4	
Me da	2	3	6	3	3	3	

**La forma de utilizar la tabla es la siguiente:**

Cada encuestado recibe una evaluación individual en dependencia de las respuestas que dé a las preguntas cerradas. Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 (máxima satisfacción) y -1 (máxima insatisfacción). El número resultante de la interrelación de las tres preguntas indica la posición de cada encuestado en la siguiente escala de satisfacción: clara satisfacción +1, más satisfecho que insatisfecho 0.5, no definido y contradictorio 0, más



insatisfecho que satisfecho -0.5 y clara insatisfacción -1. El ISG se calcula mediante la siguiente fórmula:

$$ISG=A(+1)+B(+0.5)+C(0)+D(-0.5)+E(-1)/N$$

En esta fórmula A, B, C, D, E, F representan la cantidad de encuestados colocados respectivamente en las posiciones de satisfacción 1; 2; 3 o 6; 4; 5 y donde N representa la cantidad total de encuestados.

### Resultados obtenidos

1. Los resultados obtenidos de la aplicación de la encuesta se presentan en la Tabla 8.

Tabla 11 Resultados obtenidos de los encuestados

Fuente: Elaboración propia

Categorías	N=15	Escala
Clara satisfacción	10	A
Más satisfecho	4	B
No definido	0	C
Más insatisfecho	0	D
Clara de	0	E
Contradictorio	1	F

### 2. Cálculo del Índice de Satisfacción Grupal

$$ISG=A (+1) +B (+0.5) + F (0) /N$$

$$ISG= (10(+1) +4(+0.5)) +1(0) /15=0.80$$

### 3. Interpretación del resultado del ISG

El valor obtenido del ISG fue 0.80 lo que indica máxima satisfacción de los usuarios con respecto al Componente para la configuración de los fondos de pantalla de la Distribución

Cubana GNU/Linux Nova 8.0. Se puede afirmar que se cumplió el objetivo general de la investigación. Las respuestas a las preguntas abiertas brindadas por los encuestados reafirman los beneficios que traerá la utilización del sistema informático propuesto.

### **Conclusiones del capítulo**

Se logró informatizar el proceso de configuración de fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0 a través del componente desarrollado. La aplicación de las pruebas unitarias permitió corroborar que el flujo de trabajo de las funcionalidades implementadas es el correcto, pues cada sentencia se ejecuta al menos una vez. La ejecución de las pruebas funcionales posibilitó identificar las deficiencias de la propuesta de solución y garantizar el cumplimiento de los requisitos definidos. La realización de la prueba de integración demostró que el software desarrollado es compatible con la Distribución Cubana GNU/Linux Nova en su versión 8.0. Mediante las pruebas de aceptación y la técnica de ladov se confirmó la alta satisfacción del cliente con un ISG de 0.80 hacia el componente para la configuración de fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0.

## CONCLUSIONES GENERALES

La presente investigación documentó el proceso de desarrollo del componente para la configuración de fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0. Una vez finalizado el mismo se pueden arribar a las siguientes conclusiones:

1. El análisis del marco teórico sobre el proceso configuración de fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova, así como el estudio de aplicaciones informáticas para la configuración de los fondos de pantalla de GNU/Linux, demostraron la necesidad de desarrollar un componente que contribuya con la configuración de fondos de pantalla.
2. Los 4 requisitos funcionales y los 3 no funcionales definidos permitieron cumplir con las necesidades del cliente. El análisis y diseño propició la elaboración de un componente para la configuración de fondos de pantalla en los Entornos de Escritorio de la Distribución Cubana GNU/Linux Nova 8.0. El uso de los patrones de diseño GRASP y el patrón arquitectónico 3-capas brindó una mayor calidad al software desarrollado.
3. Cumpliendo con las exigencias del cliente y las restricciones del diseño se desarrolló un componente que personaliza los fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0 permitiendo mejorar la configuración de los fondos de pantalla.
4. La aplicación de las pruebas de software permitió evaluar la solución desarrollada y garantizar el correcto funcionamiento del componente implementado. La utilización de la técnica de ladov permitió conocer la satisfacción de los usuarios hacia el componente para la configuración de fondos de pantalla de la Distribución Cubana GNU/Linux Nova 8.0 con un ISG de 0.80.

## **RECOMENDACIONES**

Se recomienda el uso de otros entornos de escritorios para posteriores versiones de GNU/Linux Nova, como XFCE, KDE Plasma o Cinnamon, debido a las funcionalidades de personalización y configuración que estos poseen.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] MMMM: *Qué es un fondo de pantalla | MMMM Blog*. URL <https://blog.uclm.es/mariamarmolina/que-es-un-fondo-de-pantalla/>. - abgerufen am 2022-06-16
- [2] SILVA, YARITZA YANET GONZÁLEZ ; ZUBIETA, ALEXIS LÓPEZ ; FUENTES, ALLAN PIERRA ; RIVAS, ING DAIRELYS GARCÍA: Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas., S. 80
- [3] *Top 10+ Best Linux Wallpaper Changer Software - redessy.com*. URL <https://redessy.com/top-10-best-linux-wallpaper-changer-software/>. - abgerufen am 2022-06-30
- [4] GONZÁLEZ SILVA, YARITZA YANET ; LÓPEZ ZUBIETA, ALEXIS: *Núcleo del entorno de escritorio de Nova Ligerio 2015, 2014*
- [5] TOLEDO RUIZ, SUEYAILE: *Herramienta informática para la administración remota de tareas programadas en Nova 7.0., 2020*
- [6] 1386, NOVA 3 0 PARTE DE LA FAMILIA GNU/LINUX DETALLE DEL ESCRITORIO DE NOVA GNU/LINUX DESARROLLADOR UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS WEB OFICIAL [HTTP://WWW UCI CU](http://www.uci.cu) [HTTP://WWW NOVA CU](http://www.nova.cu) PLATAFORMAS SOPORTADAS ; AMD64 ; ESPAÑOL, MIPSEL INFORMACIÓN GENERAL LANZAMIENTO OFICIAL 15 DE FEBRERO DE 2011 NÚCLEO MONOLÍTICO LICENCIA PRINCIPALMENTE GPL MODELO DE DESARROLLO SOFTWARE LIBRE INTERFAZ GRÁFICA POR DEFECTO GNOME IDIOMAS: *Nova GNU/Linux - EcuRed*. URL [https://www.ecured.cu/Nova\\_GNU/Linux](https://www.ecured.cu/Nova_GNU/Linux). - abgerufen am 2022-05-25
- [7] *Nova ESCRITORIO | Universidad de las Ciencias Informáticas*. URL <https://www.uci.cu/investigacion-y-desarrollo/productos/nova/nova-escritorio>. - abgerufen am 2022-06-23
- [8] OMG Unified Modeling Language™ (OMG UML), Superstructure.
- [9] TANNHAUSSER: *Así será el nuevo GNOME Control Center*. URL <https://lamiradadelreplicante.com/2016/05/29/asi-sera-el-nuevo-gnome-control-center/>. - abgerufen am 2022-11-28. — La mirada del replicante
- [10] *Definición de fondo de pantalla (o de escritorio)*. URL [https://www.alegsa.com.ar/Dic/fondo\\_de\\_pantalla.php](https://www.alegsa.com.ar/Dic/fondo_de_pantalla.php). - abgerufen am 2022-11-28
- [11] JENIFA, ASHLIN: *Los 9 mejores entornos de escritorio Linux para usar en 2022*. URL <https://geekflare.com/es/linux-desktop-environment/>. - abgerufen am 2022-11-10. — Geekflare
- [12] *Entornos de escritorio*. URL <https://docs.freebsd.org/doc/6.0-RELEASE/usr/share/doc/es/books/handbook/x11-wm.html>. - abgerufen am 2022-06-23
- [13] *GNOME – Simple, beautiful, elegant*. URL <https://www.gnome.org/>. - abgerufen am 2022-11-10
- [14] *Create a Custom Transitioning Background for Your Gnome 2.28 Desktop | Linux Journal*. URL <https://www.linuxjournal.com/content/create-custom-transitioning-background-your-gnome-228-desktop>. - abgerufen am 2022-07-05
- [15] *Inicio - Comunidad KDE*. URL <https://kde.org/es/>. - abgerufen am 2022-11-10

- [16] *Escritorio Cinnamon*. URL <https://spins.fedoraproject.org/es/cinnamon/>. - abgerufen am 2022-07-05
- [17] JAMES, JOSHUA: *Entorno de escritorio Cinnamon en Debian 11 Bullseye*. URL <https://www.linuxcapable.com/es/c%C3%B3mo-utilizar-el-entorno-de-escritorio-de-canela-en-debian-11-bullseye/>. - abgerufen am 2022-07-05. — LinuxCapable
- [18] *Entorno de escritorio Xfce*. URL <https://www.xfce.org/?lang=es>. - abgerufen am 2022-07-05
- [19] *¿Qué es un área de trabajo y cómo me ayudará?* URL <https://help.gnome.org/users/gnome-help/stable/shell-workspaces.html.es>. - abgerufen am 2022-10-30
- [20] STAFF, MOYENS: *5 de los mejores cambiadores de fondos de pantalla para Linux*. URL <https://es.moyens.net/linux/5-de-los-mejores-cambiadores-de-fondos-de-pantalla-para-linux/>. - abgerufen am 2022-11-09. — Moyens I/O
- [21] *Variety*. URL <https://peterlevi.com/variety/>. - abgerufen am 2022-05-20. — Variety
- [22] *Wallch, una alternativa a Variety Wallpaper Changer | Ubunlog*. URL <https://ubunlog.com/wallch-una-alternativa-a-variety-wallpaper-changer/>. - abgerufen am 2022-10-30
- [23] NOEL: *El visor y organizador de imágenes Shotwell acaba de pública su versión 0.30.8 | LiGNUx.com*. URL <https://linux.com/el-visor-y-organizador-de-imagenes-shotwell-acaba-de-publica-su-version-0-30-8/>. - abgerufen am 2022-07-05. — LiGNUx
- [24] SHOTWEL, SHOTWELL SHOTWELL ES UN VISOR Y ORGANIZADOR DE IMÁGENES PARA EL ESCRITORIO GNOME DESARROLLADOR YORBA FOUNDATION YORBA/SHOTWELL LANZAMIENTO INICIAL 2011-03-22 ÚLTIMA VERSIÓN ESTABLE 0 9 0 GÉNERO ORGANIZADOR DE FOTOS DIGITALES SISTEMAS OPERATIVOS COMPATIBLES LINUX IDIOMA PLURILINGÜE LICENCIA GNU LESSER GENERAL PUBLIC LICENSE 2 1 SITIO WEB: *Shotwell - EcuRed*. URL <https://www.ecured.cu/Shotwell>. - abgerufen am 2022-07-05
- [25] MUÑOZ GONZÁLEZ, DAREYNA DE LA CARIDAD: *API REST asociada al repositorio de Nova*, Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas, 2018
- [26] TOLEDO RODRÍGUEZ, MIOSOTIS: *Módulo para la gestión de copias de seguridad en Nova 360*, Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas, 2019
- [27] TAMARA, R. S.: Metodología de desarrollo para la Actividad productiva de la UCI.
- [28] *¿Qué es UML | Lenguaje de modelado unificado*. URL <https://www.uml.org/what-is-uml.htm>. - abgerufen am 2022-05-25
- [29] *Visual Paradigm - EcuRed*. URL [https://www.ecured.cu/Visual\\_Paradigm](https://www.ecured.cu/Visual_Paradigm). - abgerufen am 2022-07-05
- [30] *Glade - A User Interface Designer*. URL <https://glade.gnome.org/>. - abgerufen am 2022-05-20
- [31] TEAM, THE GTK: *The GTK Project - A free and open-source cross-platform widget toolkit*. URL <https://www.gtk.org/docs/language-bindings/python>. - abgerufen am 2022-05-20. — The GTK Team
- [32] NIVEL, LENGUAJE DE PROGRAMACIÓN C. LENGUAJE DE PROGRAMACIÓN DEL ALTO ; LINUX, PARA EL SISTEMA OPERATIVO UNIX CREADOR DENNIS RITCHIE DESARROLLADOR DENNIS RITCHIE DISEÑADO POR DENNIS RITCHIE LANZAMIENTO INICIAL 1972 GÉNERO LENGUAJE DE PROGRAMACIÓN PARA SISTEMAS SISTEMAS OPERATIVOS COMPATIBLES ; WINDOWS ; UNIX.: *C (lenguaje de*

programación) - EcuRed. URL

[https://www.ecured.cu/C\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://www.ecured.cu/C_(lenguaje_de_programaci%C3%B3n)). - abgerufen am 2022-07-02

[33] *Editores de código*. URL <https://desarrolloweb.com/coleccion/es/editores-codigo>. - abgerufen am 2022-07-02

[34] *Documentation for Visual Studio Code*. URL <https://code.visualstudio.com/docs>. - abgerufen am 2022-07-02

[35] *Git*. URL <https://git-scm.com/>. - abgerufen am 2022-07-05

[36] VERSIONES, GIT GIT ES UN SOFTWARE DE CONTROL DE ; WINDOWS, PENSANDO EN LA EFICIENCIA Y LA CONFIABILIDAD DEL MANTENIMIENTO DE VERSIONES DE APLICACIONES CUANDO ÉSTAS TIENEN UN GRAN NÚMERO DE ARCHIVOS DE CÓDIGO FUENTE CREADOR MATTHEW AHRENS Y JEFF BONWICK DESARROLLADOR ORACLE EQUIPO DE DESARROLLO DE OPENZFS COMO FORK DISEÑADO POR LINUS TORVALDS VERSIONES 2 7 0 ÚLTIMA VERSIÓN ESTABLE PLANTILLA:FECHA DE LANZAMIENTO PLATAFORMAS SOPORTADAS MICROSOFT ; WINDOWS, LINUX SISTEMAS OPERATIVOS COMPATIBLES MICROSOFT ; [HTTP://GIT-SCM.COM](http://git-scm.com), LINUX IDIOMA INGLES LICENCIA GNU GPL v2 SITIO WEB: *Git - EcuRed*. URL <https://www.ecured.cu/Git>. - abgerufen am 2022-07-05

[37] *Especificación de Requisitos del Software*. URL <https://spuzi.github.io/Spuzipedia/ingenieriaDelSoftware/analisis/ERS.EspecificacionRequisitosSoftware.html>. - abgerufen am 2022-07-06

[38] *Arquitectura de n capas*. URL <http://iutll-abdd.blogspot.com/2012/05/arquitectura-de-n-capas.html>. - abgerufen am 2022-09-22

[39] PRESSMAN, ROGER: *Ingeniería de software. Un enfoque práctico.*, 2010

[40] HAMILTON, DANIEL: *Estándares de codificación de software y pautas de programación*. URL <https://es.parasoft.com/blog/an-ounce-of-prevention-software-safety-security-through-coding-standards/>. - abgerufen am 2022-10-06. — Parasoft

[41] *Diagrama de componentes: modelado eficiente de sistemas con módulos de software*. URL <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-componentes/>. - abgerufen am 2022-10-08. — IONOS Digital Guide

[42] TURRADO, JORGE: *Qué son las pruebas de software*. URL

<https://www.campusmvp.es/recursos/post/que-son-las-pruebas-de-software.aspx>. - abgerufen am 2022-10-08. — campusMVP.es

[43] SOMMERVILLE, IAN: *Software Engineering*

## ANEXOS

### **Anexo 1: Entrevista realizada a los especialistas del equipo de desarrollo del Centro de Software Libre (CESOL) de la Universidad de las Ciencias Informáticas (UCI)**

**Objetivo:** Evaluar la propuesta de solución desarrollada y conocer el índice de satisfacción de sus usuarios potenciales.

- ¿En qué lenguaje de programación está escrito el Centro de Control de Gnome que utiliza Nova?
- ¿Qué funcionalidades son imprescindibles para la implementación de la propuesta de solución?
- ¿Cuáles características son necesarias para la interfaz de la propuesta de solución?
- ¿Con cuáles sistemas o versiones debe ser compatible la propuesta de solución?



**Anexo 2:** Guía de observación

**Observador:** Fidel Alejandro Sánchez Fernández

**Lugar:** Laboratorio de práctica profesional del Centro de Software Libre (CESOL)

**Objetivo:** Identificar los elementos fundamentales sobre los Entornos de Escritorios de GNU/Linux existentes, las características de la Distribución Cubana GNU/Linux Nova y conocer sobre las aplicaciones existentes para configurar los fondos de pantalla.

1. Entornos de Escritorios

- ¿Cuáles EE se utilizan en los sistemas de GNU/Linux?
- Características de los EE en los sistemas de GNU/Linux.

2. Distribución Cubana GNU/Linux Nova

- ¿Cuál es el EE que utiliza la Distribución Cubana GNU/Linux Nova?
- ¿Cómo se realiza la configuración de los fondos pantalla Distribución Cubana GNU/Linux Nova?
- ¿Qué opciones permite esta configuración?

3. Aplicaciones para configurar los fondos de pantallas

- ¿Cuáles son las aplicaciones más populares disponibles para sistemas de GNU/Linux, para configurar los fondos de pantalla?
- ¿Qué funcionalidades poseen estas aplicaciones?
- ¿Cuáles son las ventajas y desventajas de utilizarlas?

### **Anexo 3: Encuesta a un grupo de usuarios del Centro de Software Libre (CESOL) de la Universidad de las Ciencias Informáticas (UCI)**

**Objetivo:** Medir el índice de satisfacción de los usuarios respecto al grado de personalización y configuración de la Distribución Cubana GNU/Linux Nova.

1. ¿Podría calificar su satisfacción con respecto a su entorno de escritorio?

1\_\_ 2\_\_ 3\_\_ 4\_\_ 5\_\_

2. ¿Considera usted que se debe modificar la configuración de fondos de pantalla?

Si\_\_ No\_\_ Tal vez\_\_

3. ¿Le gustaría agregar varias imágenes como fondo de pantalla?

Si\_\_ No\_\_ Tal vez\_\_

4. ¿Le gustaría agregar imágenes a su pantalla de bloqueo?

Si\_\_ No\_\_ Tal vez\_\_

5. ¿Le gustaría configurar su fondo de pantalla desde el Centro de Control o a través de un programa externo instalable en su sistema?

Centro de Control\_\_ Programa externo\_\_ Otro\_\_

6. ¿Cómo le gustaría la configuración de sus fondos de pantalla?

## Anexo 4: Acta de aceptación de los productos de trabajo





### Acta de aceptación de productos de trabajo

#### ACTA DE ACEPTACIÓN DE PRODUCTOS DE TRABAJO

En cumplimiento del **Convenio de colaboración** establecido entre el Centro de Software Libre (CESOL) y el estudiante **Fidel Alejandro Sánchez Fernández** de la Facultad 1 de la Universidad de las Ciencias Informáticas y en función de la ejecución del proyecto: **Componente para la configuración de fondos de pantalla de la distribución cubana GNU/Linux Nova 8.0**, se hace entrega del producto que se relaciona a continuación:

- Componente para la configuración de fondos de pantalla de la distribución cubana GNU/Linux Nova 8.0

La parte Cliente, luego de haber revisado el producto de trabajo relacionado anteriormente procede a firmar la aceptación de los mismos en total conformidad:

Entrega	Recibe
<b>Nombre y apellidos:</b> Fidel Alejandro Sánchez Fernández	<b>Nombre y apellidos:</b> Enmanuel Cristian de la Cruz Mora
<b>Cargo:</b> Estudiante: Facultad 1	<b>Cargo:</b> Especialista B
<b>Firma:</b> 	<b>Firma:</b> 

Fecha: 27/10/2022