



Universidad de las Ciencias Informáticas.
Facultad 10.

Título: Tocatoro – Desktop: Un gestor de configuración de Interfaces.



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

Autoras: Valia Cantillo Pileta.

Yaneby Vázquez Rosas.

Tutora: Lic. Yenisleydis Cariaga Cristo.

Co-tutor: Ing. Abel Meneses Abad.

Junio, 2008.

Ciudad de la Habana, Cuba.

*La libertad no es poder elegir entre unas pocas
opciones impuestas, sino tener el control de tu
propia vida. La libertad no es elegir quién será tu
amo, es no tener amo.*

Richard Stallman.



Declaración de Autoría.

DECLARACIÓN DE AUTORÍA.

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

Para que así conste firmo la presente a los ____ días del mes de junio del año 2008

Valia Cantillo Pileta

Firma del Autor

Yaneby Vázquez Rosas

Firma del Autor

Yenisleydis Cariaga Cristo

Firma del Tutor

DATOS DE CONTACTO.

*Nombre y Apellidos del **Tutor**: Yenisleydi Cariaga Cristo.*

Email: yeni@uci.cu

Graduada de Licenciatura en Sociología en la Universidad de la Habana en el año 2004.

Profesora de la Universidad de las Ciencias Informáticas desde su egreso.

Ha sido tutora, consultora y co-tutora de tesis durante los últimos 2 cursos de la UCI en temas de software libres.

Ha participado en eventos como Fórum de Ciencia y Técnica, UCIENCIA, Informática 2007 y Universidad 2008.

Opta actualmente por la categoría docente de asistente y es maestrante en Estudios sociales de Ciencia y tecnología.

Se ha vinculado al proyecto productivo Unicornio, soporte e inmigración, de la facultad 10, en el rol de Jefe de Capacitación

Ha impartido cursos de postgrado de Metodología de Investigación Científica a profesores de la universidad y conferencias relacionados con el software libre en institutos fuera de la universidad.

Fue Sello Forjadores del Futuro a Jóvenes Investigadores de las BTJ en el año 2007.

Es Vicedecana de Extensión Universitaria de la Facultad10.

*Nombre y Apellidos del **Co-Tutor**: Abel Meneses Abad.*

Email: abelma@uci.cu.

Profesor universitario en la Universidad de Ciencias Informáticas.

Graduado de Ing. en Telecomunicaciones y Electrónica en el año 2004 en el ISPJAE.

Ha impartido asignaturas como Sistemas Operativos, Teleinformática II, Práctica Profesional y asignaturas del 2do perfil: GNU/Linux Básico, Propiedad Intelectual, Programación en Perl; e imparte postgrados de GNU/Linux nivel Básico, Programación Web.

Opta por la categoría docente de Instructor

Ha cursado postgrados como: Ciencia, Tecnología y Sociedad, GnuX/Linux, Ideología y Política de la Revolución Cubana.



Datos de Contacto.

Ha presentado ponencias en eventos y forma parte del grupo de investigaciones de Software Libre de la UCI y del Grupo Técnico Nacional.

Es líder del Proyecto Unicornios (Servicios y Soporte para la Migración a SWL de la UCI).

AGRADECIMIENTOS.

Aunque es posible que los agradecimientos de una tesis sean lo primero que se lea, también es cierto que son lo último que se escribe, o por lo menos eso ha sido así en mi caso. Ha llegado ya el momento de recostarse para atrás en la silla y comenzar a recordar todos los buenos momentos de estos años y a la gente que ha logrado que fuesen así, y que también han estado ahí apoyándome cuando no eran tan buenos. A todos me gustaría darles las gracias y entre ellos, que son muchos, me gustaría nombrar a algunos.

En primer lugar a la admirable fuerza que rige el universo

Al amor, la humildad, la belleza.

A Fidel y a la Revolución.

A mis padres por su amor, preocupación, y por todo el esfuerzo que han realizado en todos estos años, además de por estar a mi lado en los buenos y malos momentos.

A Karina -que es como mi hermana mayor-, por darme demasiado, por estar -aunque en la distancia- muy pendiente de mí.

A mi tía, tíos y primos por estar siempre presente.

A Yeni, mi tutora y entrañable amiga por brindarme todo su apoyo y colaboración.

A Abel, co-tutor, quien además de darme el tema me ayudó con sus exigencias y experiencias.

A Maylú, por servirme de ejemplo y serme tan cercana.

A Yaneby por estar a mi lado en la realización de este trabajo, porque nunca me abandonó y estuvo ahí en todo momento, por ser además mi amiga y soportarme todo este tiempo.

A mis compañeros en especial a Aliosha, por socorrerme tanto, a Yuliet, Yoe, Noli, Pedro, Yumir y Nidia por ayudarme cuando lo necesité, por todos los momentos en que me hicieron reír y por haberme soportado a lo largo de estos años.

Al Rector, demás miembros de la directiva, profesores y al resto de los trabajadores de esta Universidad de las Ciencias Informáticas.

Valia Cantillo Pileta.

AGRADECIMIENTOS.

Sabiendo que jamás existirá una forma de agradecer una vida de lucha, sacrificio y esfuerzo constante, quiero que sientan que el objetivo logrado también es de ustedes y que la fuerza que me ayudo a conseguirlo fue su apoyo.

Quiero Agradecer en primer lugar a la Revolución por crear una Universidad de excelencia como esta y permitir que pudiese estudiar en ella para formarme como profesional.

A la admirable fuerza que rige el universo.

A mis papás porque gracias a su cariño, guía y apoyo he logrado terminar mis estudios.

A mis hermanas por estar siempre presente.

A Yumir (mi novio), por ayudarme, apoyarme y soportarme en mis buenos y malos momentos.

Y a toda mi familia que durante estos años han estado siempre al tanto de mis estudios.

A Yeni tutora del trabajo que con mucha profesionalidad, ética, optimismo y solidaridad me apoyó y guió en la realización de esta tesis, llegando a ser más que mi tutora, mi amiga.

Al Ing. Abel Meneses Abad por darme la oportunidad de realizar este trabajo y ayudarme con el mismo.

A Maylú por ser mi profesora y guía en estos dos últimos años.

A Guzmán por dedicarme tiempo y saldar mis dudas.

A mis amigos que me brindaron su ayuda sin interés.

A mis compañeros(as) de estudio que de ellos he aprendido muchas cosas en la vida.

En fin a todos los que por razones de espacio no puedo mencionar aquí, pero que no se me olvidan pues de una forma u otra contribuyeron tanto a mi formación profesional como al desarrollo de este trabajo.

Yaneby Vázquez Rosas.



DEDICATORIA.

Dedico la presente tesis a los seres que más amo en este mundo:

A mi abuela Inés por estar siempre presente, y por regalarme todo el amor del mundo.

A mis padres que han batallado conmigo desde que nací y que aun siguen cuidando de mí.

A mi prima Karina por ser ante todo mi hermana, mi amiga y mi apoyo incondicional.

A todos ellos quiero retribuir con mi trabajo, háganlo suyo también.

Valia Cantillo Pileta.



DEDICATORIA.

En la vida siempre hay personas que están presentes en cada acto que se realiza para dar su apoyo incondicional, esas personas no se olvidan, se llevan en nuestros corazones.

*Quisiera dedicar este trabajo especialmente a **mi bisabuela** por regalarme todo su amor y cariño.*

*A **mi mamá**, por darme la vida y ser protagonista en cada momento trascendental de mi vida.*

*A **mi papa**, que ha sido que ha sido luz, ejemplo y guía.*

*A **mi hermanita menor** por ser mi alma gemela.*

*A **mi hermana mayor y a mis sobrinitos bellos**.*

*A **Yumir (mi novio)** por su apoyo incondicional y por quererme tanto.*

Yaneby Vázquez Rosas.

RESUMEN.

Ante la tarea de enfrentar un proceso de migración de Software Privativo a Software Libre, se hace necesario la configuración de las aplicaciones que en el nuevo sistema se usarán, debido mayormente a que las personas tienden a temer al cambio. El objetivo de este trabajo de diploma es analizar y diseñar un gestor de configuración de interfaces del entorno de escritorio Gnome de los Sistemas GNU/Linux para la enseñanza primaria cubana: Tocororo – Desktop, que permita una interacción agradable y amena entre los niños y la tecnología.

Con esta investigación se logra una sistematización del estudio de algunos de los entornos de escritorio de los sistemas operativos GNU/Linux. Se analizan las ventajas y desventajas de los mismos y los elementos que hacen de cada uno de ellos un candidato para su uso en las escuelas primarias. Llegando a la conclusión de la factibilidad de la utilización de Gnome como el más configurable y más factible para el país, por ser el que se utiliza en Nova/Linux, que se perfila como la distribución cubana de Software Libre y para su implementación por el **Ministerio de Educación (MINED)**. En las diferentes historias de usuario se aborda todo el proceso planificación y definición de la propuesta.

Finalmente se obtiene la propuesta de una aplicación que permitirá a los usuarios configurar su escritorio, de modo que se puedan entender mejor con él y les sea familiar, logrando así un mejor aprendizaje de las tecnologías utilizadas.

Palabras claves: Software Privativo, Software Libre, Gnome, GNU/Linux, Tocororo – Desktop, MINED.

ÍNDICE.

DECLARACIÓN DE AUTORÍA.....	I
DATOS DE CONTACTO.....	II
AGRADECIMIENTOS.....	IV
DEDICATORIA.....	VI
RESUMEN.....	VIII
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 ¿POR QUÉ SOFTWARE LIBRE?	6
1.2 DEFINICIÓN DE ENTORNOS DE ESCRITORIOS.....	10
1.3 ENTORNOS ESCRITORIOS EN LA ACTUALIDAD.....	10
1.3.1 CDE.....	11
1.3.2 Xfce.....	14
1.3.3 KDE.....	16
1.3.4 Gnome.....	19
1.4 ANÁLISIS COMPARATIVO.....	22
1.5 TOCORORO – DESKTOP.....	23
1.5.1 <i>Análisis internacional de Migración a SWL dedicada a niños</i>	23
1.5.2 <i>Análisis nacional de Tocororo – Desktop</i>	24
1.6 METODOLOGÍAS ÁGILES.....	25
1.6.1 Scrum.....	26
1.6.2 XP.....	27
1.7 TECNOLOGÍAS UTILIZADAS.....	28
1.7.1 Lenguajes.....	28
1.7.2 IDE.....	29
1.7.3 Herramienta Case.....	29
1.7.4 Herramientas de Diseño.....	30
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	33
2.1 ELEMENTOS GESTIONADOS POR TOCORORO – DESKTOP.....	33
2.1.1 GrubSplash.....	34
2.1.2 BootSplash.....	36

2.1.3	<i>GDM</i>	38
2.1.4	<i>ScreenSplash</i>	41
2.1.5	<i>Escritorio</i>	46
2.2	HERRAMIENTAS DE CONFIGURACIÓN EXISTENTES	55
2.3	PROPUESTA DEL SISTEMA: TOCORORO – DESKTOP	57
3.1	PLANIFICACIÓN Y DEFINICIÓN	62
3.1.1	<i>Lista de Reserva del Producto</i>	62
3.1.2	<i>Modelo de historias de usuarios del dominio</i>	67
3.1.3	<i>Definición de las principales Historias de Usuarios</i>	68
3.1.4	<i>Tareas de la ingeniería</i>	73
3.1.5	<i>Plan de Lanzamiento</i>	78
3.1.6	<i>Tarjetas CRC</i>	79
3.1.7	<i>Prototipos de interfaz</i>	80
	CONCLUSIONES	84
	RECOMENDACIONES	85
	REFERENCIAS BIBLIOGRÁFICAS	86
	BIBLIOGRAFÍA	87
	GLOSARIO DE TÉRMINOS	90
	ANEXOS	93

INTRODUCCIÓN.

El ser humano ha requerido desde tiempos inmemoriales de medios que le permitan realizar cálculos con los más disímiles propósitos, entre ellas el procesamiento de la información. Con el paso de los años estos medios se han ido evolucionando hasta las potentes computadoras, protagonistas de la era del conocimiento.

En la actualidad, el uso de estas computadoras es uno de los componentes fundamentales del proceso de modernización de la sociedad a nivel mundial y del cual no escapa Cuba, insertada ya, con un protagonismo creciente y sólido, en dicho proceso.

Uno de los sectores fundamentales en el que se trabaja para la enseñanza de la computación es en las escuelas primarias de todo el país, que constituyen la base de la educación y tiene carácter obligatorio y universal, con beneficio para todos los niños y niñas, partiendo del principio de que a la más joven generación hay que enseñarla, atenderla y educarla [*Santiesteban, 2006*]¹. Su fin es contribuir a la formación integral del escolar, fomentando desde los primeros grados, la interiorización de los conocimientos y orientaciones valorativas, que se reflejan gradualmente en los sentimientos, formas de pensar y comportamiento, que se corresponden con los valores e ideales de la Revolución Socialista.

La utilización de la computación como medio de aprendizaje en este nivel de enseñanza abre nuevos caminos para la dirección del proceso docente educativo. Garantiza además el vínculo del mismo con el entorno tecnológico en que se desarrolla la sociedad.

Actualmente en las escuelas primarias del país hay alrededor de 24 000 computadoras con softwares de aprendizaje para facilitar la educación. En Cuba se lleva a cabo el proceso de enseñanza - aprendizaje basado en software privativo. Los softwares de aprendizaje se encuentran soportados sobre Windows.

La utilización de sistemas operativos privativos tiene notables desventajas, como son la exigencia de pago por concepto de licencia para su uso, a lo cual se unen restricciones adicionales. Ante estas y otras problemáticas se tiene la perspectiva del **Software Libre (SWL)** como alternativa.

Cuba es un país bloqueado por una de las más grandes potencias, que además es uno de los países que lideran el desarrollo informático en el mundo. En esta isla, “el bloqueo impuesto limita, entorpece, y encarece la adquisición del software legal necesario para el desarrollo de la sociedad cubana”, y es por esta razón que la perspectiva del software libre se abre como solución a esta problemática para el desarrollo de la informática cubana [Olivé, 2005]².

El software libre alude a la libertad que poseen los usuarios al utilizar determinado software, constituye una alternativa a los problemas propietarios y un paso superior a una sociedad con un flujo libre de la información, para la mayoría de los ámbitos, tanto públicos como privados. Dentro de este conjunto de soluciones informáticas aparecen los entornos de escritorios como vía para ofrecer una interacción amigable y cómoda.

Dentro de los entornos de escritorios libres **Common Desktop Environment (CDE)**, **K Desktop Environment (KDE)**, **XForms Common Environment (Xfce)** y **GNU Network Object Model Environment (Gnome)** son los más sencillos, y este último constituye la línea de producción de Nova, distribución que se perfila como candidata para su implementación por el MINED, y que produce la **Universidad de las Ciencias Informáticas (UCI)**. Siendo además la distribución cubana más completa hasta el momento, que cuenta con un equipo de soporte técnico comprometido con su desarrollo.

Lo que se propone con esta investigación es la creación de un gestor de configuración de interfaces del entorno de escritorio Gnome, amenizando así su interacción con los niños, motivando el proceso de aprendizaje desde la raíz.

Tocororo-Desktop es el nombre de este gestor de configuración que integra creatividad y sueños para los niños y adolescentes cubanos con el fin de que la informática se parezca más a ellos, se involucre más en sus conceptos y su manera singular de ver la vida. Es exactamente un proyecto que tiene como objetivo configurar las interfaces de usuario de los sistemas GNU/Linux para la enseñanza primaria cubana, insertándose su desarrollo en el proceso de migración tecnológica que vive nuestro país.

En la actualidad no existen en el país referencias del desarrollo de una aplicación similar a Tocororo-Desktop que posea estas características. Debido, fundamentalmente, al hecho de

que todavía estamos en un estadio temprano del proceso de migración a software libre, en el que el gestor de configuración de interfaces pretende insertarse.

Con este trabajo se pretende demostrar la fiabilidad del desarrollo de Tocororo-Desktop, cuya importancia para el proceso educativo que se gesta en las primarias del país se resume en la idea de llevar la informática a los niños, no los niños a la informática, es decir, lograr que esta ciencia se convierta en un medio de aprendizaje que se adecúe a los sueños y fantasías propios de la infancia.

Existen herramientas para gestionar la configuración de los entornos de escritorios, por ejemplo en Gnome existe el *gdmsetup* que se usa para las preferencias de ventanas de entrada como por ejemplo la pantalla de inicio, acceso remoto, entre otros. Además otras herramientas como *gconf-editor* que es un editor de configuración que permite combinar la configuración de los distintos programas. De esta forma se puede realizar una configuración compatible para todas las aplicaciones.

En general la configuración-personalización de algunos componentes de un sistema operativo es un proceso algo engorroso. Todas estas alternativas son difíciles de utilizar, por su complejidad visual, por lo que se necesita de cierto conocimiento en el campo de la informática, además de que no permiten crear nuevos temas, es por ello que surge el **problema científico** de ¿Cómo simplificar el proceso de configuración del entorno de escritorio Gnome, del sistema operativo GNU/Linux?, de ahí que el **objeto de estudio** sean los Entornos de escritorios de los sistemas operativos GNU/Linux, y tenga como **campo de acción** el estudio del Entorno de escritorio Gnome del sistema operativo GNU/Linux, ambos encaminados a cumplir el **objetivo general** de Analizar y diseñar un gestor de configuración de interfaces del entorno de escritorio Gnome de los Sistemas GNU/Linux para la enseñanza primaria cubana: Tocororo – Desktop. Para cumplir con el objetivo general, se han definido los siguientes **objetivos específicos**:

1. Estudiar los entornos de escritorios CDE, KDE, Xfce y Gnome de los sistemas operativos GNU/Linux.
2. Demostrar la factibilidad del uso de Gnome para la implementación de Tocororo - Desktop.

3. Identificar todos los componentes configurables en el sistema operativo que afectan la imagen visual.
4. Diseñar el gestor de configuración de interfaces: Tocororo – Desktop para la personalización de los sistemas GNU/Linux en las primarias cubanas.

Para cumplir con los objetivos se define el siguiente conjunto de **tareas**:

1. Realizar un levantamiento bibliográfico.
2. Realizar un análisis comparativo de los entornos de escritorio CDE, KDE, Xfce, Gnome.
3. Profundizar en el estudio de los componentes configurables de los entornos de escritorios.
4. Utilizar las metodologías ágiles **SCRUM** y eXtreme Programming (**XP**).

La **estructuración del contenido** es:

Capítulo 1: Fundamentación teórica. Este capítulo se enmarcará en la justificación del por qué software libre, además de una actualización a nivel internacional y nacional de los entornos de escritorios en sistemas operativos GNU/Linux, haciendo un estudio de las principales características que poseen para llegar a una conclusión sobre cual utilizar para la aplicación de nuestro sistema. Incluye una explicación de las metodologías y tecnologías utilizadas en el desarrollo de la investigación y el estado del arte del sistema que se propone.

Capítulo 2: Características del sistema. En este capítulo se hará referencia a las características del entorno de escritorio Gnome, se analizarán los elementos configurables de este entorno de escritorio y que gestiona Tocororo - Desktop. Se realiza un estudio de las herramientas de configuración existentes, además de presentar una descripción general de la propuesta del sistema que se estudia, cuáles son sus objetivos así como una explicación de en qué consiste el sistema.

Capítulo 3: Planificación y Definición del sistema. En este capítulo se plasmarán cuestiones relacionadas con el análisis y el diseño del sistema, se abarcará la definición de las historias de usuarios que propone la metodología ágil utilizada. Se muestran los



Introducción.

resultados de esta etapa que lleva consigo las tarjetas **Clase Responsabilidades Colaboradores (CRC)** y los prototipos de interfaces generados.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.1 ¿Por qué Software Libre?

Una sociedad libre necesita **Software Libre (SWL)**. Necesita la libertad para inspeccionar el software, aprender de él, y modificarlo de acuerdo a sus necesidades. Las computadoras se usan para compartir ideas, cultura e información. Sin estas libertades sobre el software, se corre el riesgo de perder el control sobre lo que se comparte. Esto ya está sucediendo hoy; desde tecnologías evidentemente complicadas como la **Gestión Digital de Restricciones (DRM)** hasta la **Computación Confiable (TC)**. El derecho que tiene cualquier persona de formar parte de la cultura se está viendo amenazado. Si hay que renunciar a la libertad para usar un software, quizá entonces no se esté contento con él.

El SWL como tecnología trae asociada una corriente de pensamiento que plantea entre otras muchas cosas que el conocimiento no debe tener amos ni fronteras, que cada persona debe tener la libertad de compartirlo con el resto del mundo sin ataduras. Gracias a esa ideología, se ha desarrollado en el mundo un movimiento muy fuerte cuyo objetivo es escribir programas informáticos que cumplan las siguientes condiciones [*Stallman, 2000*]³:

Libertad 0 Ejecutar el programa con cualquier propósito (privado, educativo, público, comercial, etc.).

Libertad 1 Estudiar y modificar el programa (para lo cual es necesario poder acceder al código fuente).

Libertad 2 Copiar el programa de manera que se pueda ayudar al vecino o a cualquiera.

Libertad 3 Mejorar el programa y hacer públicas las mejoras, de forma que se beneficie la comunidad de usuarios.

Estos derechos típicamente no están disponibles con el software privativo. Usualmente en el software propietario hay que pagar una “licencia de uso” al creador (como el pago de derechos por el uso de una patente) y se está sujeto a las condiciones del fabricante.

Normalmente estas condiciones no otorgan ningún derecho al usuario final. Los derechos mencionados anteriormente tienen una serie de efectos colaterales sobre el Software:

- Tiende a ser muy eficiente (porque mucha gente lo optimizan, mejoran).
- Tiende a ser muy robusto (mucha gente *puede* arreglarlo, no solamente el creador o la compañía que lo produce). Mucha gente tiende a contribuir, porque es en el interés de todos mejorar esta base común.
- Tiende a ser muy diverso: la gente que contribuye tiene muchas necesidades diferentes y esto hace que el software esté adaptado a una cantidad más grande de problemas.

Dichas condiciones no implican que el software sea gratis, solo cambian la importancia de lo que se vende, el modelo de negocios está orientado al servicio que se brinda, no al software como mercancía. Debido a la distribución del código y la documentación, no se establecen relaciones de extrema dependencia entre el proveedor y el cliente, transfiriendo así el conocimiento generado para su posterior desarrollo independiente.

Por otra parte el software privativo no ayuda de esta manera como el SWL pues este requiere de:

- **Cursos de aprendizaje costosos.** Es difícil aprender a utilizar eficientemente el software propietario sin haber asistido a costosos cursos de capacitación.
- **Secreto del código fuente.** El funcionamiento del software propietario es un secreto que guarda celosamente la compañía que lo produce. En muchos casos resulta riesgosa la utilización de un componente que es como una caja negra, cuyo funcionamiento se desconoce y cuyos resultados son impredecibles, o es imposible encontrar la causa de un resultado erróneo.
- **Ilegalidad de copias sin licencia para el efecto.** Es ilegal hacer copias del software propietario sin antes haber contratado las licencias necesarias.
- **Dependencia a proveedores.** En la mayoría de los casos el gobierno se hace dependiente de un solo proveedor, de modo que si cambia la metodología de este, se puede ver afectado el gobierno con dichos cambios.

El SWL tiene independencia de código, lo que posibilita la continuidad del desarrollo del software, si en él se permite dar soporte a determinada empresa, otra puede retomar el proyecto fácilmente. Sin embargo, el software privativo promueve la incompatibilidad entre programas, archivos y versiones del mismo programa, generando mayor dependencia y costes. En ocasiones se hace preciso incluso cambiar de ordenador para poder usar nuevas versiones de programas, porque las restricciones de hardware de ese nuevo programa han aumentado, esto provoca incompatibilidades entre ordenadores con sistemas antiguos y otros modernos. El SWL nos libra de estas incomodidades. En Linux se da gran dependencia al principio y decrece según se aprende.

El SWL reduce el costo, sobre todo si no se piratea, aunque no hay que descartar tener que comprar alguna licencia, porque no todo el SWL es gratuito, pero aun así, se tiene ciertas libertades de conocimiento del software que subsanan todos esos posibles gastos. Por otra parte, en el software privativo los costes van creciendo con el tiempo, con el número de programa, el número de versiones y el número de ordenadores. Aunque el SWL tiene alto coste de migración, este es independiente del número de ordenadores y/o programas.

El SWL es más seguro, estable, y no hay virus o aplicaciones no deseadas que se instalan por sí mismas. No hay espías porque es software libre. Además de la transparencia de código, que mientras el software privativo es testeado por millones de personas, el SWL es un código consensuado desde la comunidad, y que no responde a los intereses de una única empresa.

Cuba se ha caracterizado siempre por ser un país consumidor de tecnologías informáticas, cuyos productores son, en su mayoría, transnacionales pertenecientes a países del primer mundo. Ésta relación de consumo ha generado una dependencia tecnológica que ahoga el desarrollo de la naciente Industria de Software Cubana y aumenta la amplia brecha digital existente entre Latinoamérica y Europa o la América Anglosajona. Es por ello que en la actualidad, el país se encuentra en la necesidad objetiva de independizarse de los monopolios de software debido a la inexistencia de recursos para pagar las licencias de uso de sus productos. De hecho, aun en caso de tener el poder adquisitivo, no sería posible por las normativas "antiterroristas" del gobierno norteamericano que impiden el acceso directo a las mismas.

Un ejemplo de estas medidas es la restricción que tienen las direcciones de Internet Protocol (IP) pertenecientes a la isla. Es imposible descargar actualizaciones o software de los sitios de Microsoft y Sun Microsystems. Esto trae aparejada la necesidad de comprar el software mediante terceros países a un costo mayor, sin contar con la posibilidad de desarrollar una Industria Nacional de Software con una dependencia tan estrecha de compañías norteamericanas.

Para garantizar la soberanía tecnológica de la nación cubana ya se hace indispensable una migración parcial o total a SWL. Dicha situación crea una demanda fácilmente cubierta por Nova con muchas ventajas adicionales:

- Creado y mantenido en la UCI por un proyecto exclusivamente dedicado a su desarrollo y mantenimiento que además se hace responsable por su buen funcionamiento, servicio de actualizaciones, capacitación y Soporte Técnico.
- Compuesto en su mayoría por aplicaciones de SWL, por lo que está exento de pagos de licencias por su uso.
- Puede constituir la base para la creación de software a la medida sin incurrir en costos adicionales. Incluso se pueden crear sistemas derivados para múltiples entornos que responderían a una misma plataforma.
- En caso de su implantación a nivel nacional, se puede crear una o más instituciones que provean servicios especializados sobre la distribución.
- Rompe la relación de dependencia existente entre Cuba y Microsoft, logrando que la infraestructura nacional no sea afectada directamente por las decisiones de dicha empresa, y que en el caso de una situación de crisis se puedan mantener los servicios sin necesitar actualizaciones provenientes de compañías norteamericanas.
- Estimula la vinculación de estudiantes al desarrollo de software y fomenta la innovación pues Nova acepta la colaboración de cualquier persona dispuesta a contribuir con el proyecto.

1.2 Definición de Entornos de Escritorios.

Un entorno de escritorio (en inglés, **Desktop Environment**) es un conjunto de software para ofrecer al usuario de un ordenador una interacción amigable y cómoda. Es una solución completa de interfaz gráfica de usuario **Graphical User Interface (GUI)**, que lo provee de íconos, barras de herramientas, aplicaciones e integración entre aplicaciones con habilidades como drag&drop (arrastrar y soltar) que hacen que se obtenga un manejo más familiar del sistema operativo.

Un “entorno de escritorio” puede ser cualquier elemento que vaya desde un simple gestor de ventanas hasta una completa “suite” de aplicaciones de escritorio tales como KDE o Gnome. En general cada entorno de escritorio se distingue por su aspecto y comportamiento particular, aunque algunos tienden a imitar características de escritorios ya existentes, pero no es más que un conjunto de software ideado para ofrecer al usuario de un ordenador un ambiente cómodo.

El primer entorno de escritorio moderno que se comercializó fue desarrollado por Xerox (Precisamente el nombre "Xerox", determinado por su inventor Chester Carlson, diferenciaba en el mercado su tecnología de copiado en "seco", ya que "xer" es la raíz griega para significar seco. Por entonces el copiado "*mojado*" era el mayormente utilizado en la industria.) En los años 1980. Actualmente el entorno más conocido es el ofrecido por la familia Windows aunque existen otros como los de **Maquintosh** (Classic y Cocoa) y de SWL como Gnome, KDE o Xfce.

1.3 Entornos Escritorios en la actualidad.

El desarrollo de la informática a nivel mundial trae aparejado un conjunto de cambios en muchos aspectos de software, cada desarrollador o usuario, aprovechando las ventajas que proporciona el SWL, cambia, modifica y adapta su sistema operativo a sus necesidades y según sus posibilidades. Años atrás, los sistemas operativos libres no tenían un gestor de escritorio que fuese común para todos, por lo que se basaban en la ejecución por consola del sistema operativo.

Aunque con probabilidad de no ser la primera solución en cuanto a entornos de escritorios "amigables" para el usuario, la difusión a mediados de 1995 del sistema operativo Windows95™ supuso un cambio radical en la interacción de los usuarios de a pie con los ordenadores. De los sistemas unidimensionales de línea de instrucciones (los terminales), se pasó a la metáfora de entorno del escritorio bidimensional, donde el ratón ganó terreno al teclado. Windows95™, más que una innovación tecnológica, debe ser acreditado como el sistema que consiguió adentrarse en todos los entornos personales y de oficina, marcando las pautas a seguir (normas que, a principios del siglo XXI, todavía se siguen padeciendo).

Los seguidores del SWL, rápidamente se hicieron eco de este notable éxito y, a la vista de que los entornos UNIX carecían de sistemas tan intuitivos a la vez que libres, decidieron ponerse manos a la obra; es entonces cuando comienzan a surgir proyectos como KDE, Gnome y otros, que se encargaron de crear entornos de escritorios compatibles para sistemas operativos libres y en especial para Linux.

En la actualidad existen un sinnúmero de entornos de escritorios, cada uno con sus ventajas y desventajas, según lo que se quiera lograr. En este capítulo, atendiendo a las necesidades del objetivo trazado, se hará referencia a 4 de ellos: CDE, Xfce, KDE y Gnome.

1.3.1 CDE.

Common Desktop Environment (CDE) es un entorno de escritorio gráfico para UNIX, basado en la biblioteca Motif que es una librería para la creación de entornos gráficos bajo X Window System, es también un estándar de la industria bajo el código Institute of **E**lectrical and **E**lectronics **E**ngineers (**IEEE**) 1295. Muchos desarrolladores argumentan que se ha vuelto obsoleto en comparación con otras librerías como GTK o Qt, pero continúa siendo utilizado por varios sistemas antiguos. CDE fue desarrollado en conjunto por Hewlett-Packard, **I**nternational **B**usiness **M**achines (**IBM**), Novell y Sun Microsystems.

CDE es una interfaz fácil de utilizar que proporciona una apariencia homogénea entre los entornos UNIX^(R). SunSoft^(TM), Inc., Hewlett-Packard Company, IBM Corporation y Novell, Inc. aportaron sus tecnologías "mejor desarrolladas" para establecer un nuevo estándar para las interfaces de usuario y las aplicaciones basado en el Sistema de ventanas X^(TM) y en Motif^(R).

La contribución de SunSoft a la implementación del CDE incluye su protocolo de mensajes ToolTalk^(TM), la herramienta de desarrollo DevGuide, herramientas de correo y agenda de sus utilidades DeskSet^(TM) de OpenWindows^(TM) y las capacidades de arrastrar y soltar y de clasificación de datos.

El Solaris CDE ofrece beneficios adicionales a sus usuarios y programadores y a su vez mantiene la conformidad con los estándares CDE. Por ejemplo, puede utilizar un visualizador de imágenes para visualizar, girar, ampliar y convertir imágenes y archivos PostScript^(TM).

Si se está acostumbrado al entorno OpenWindows, tiene acceso a los fondos, las paletas de color y el menú emergente del Espacio de trabajo que ya conoce. Puede también ejecutar las aplicaciones OpenWindows sin necesidad de modificaciones, con la interacción de arrastrar y soltar entre las aplicaciones OpenWindows y CDE.

Los programadores pueden auxiliarse de las utilidades de conversión de archivos para simplificar la transición de sus aplicaciones OpenWindows y Motif a CDE. Las características más importantes de las que dispone bajo CDE son:

1. Entorno CDE.
2. Herramientas de escritorio.
3. Entorno de desarrollo CDE.

El entorno CDE está formado por:

- Gestor de inicio de sesión - Ventana para entrar en el escritorio, disponible automáticamente cuando se reinicia su estación de trabajo.
- Gestor de sesiones - Servicio que inicia sus aplicaciones en el escritorio al inicio de la sesión y "recuerda" su estado cuando se finaliza la sesión.
- Gestor de ventanas - Servicio que controla las ventanas de las aplicaciones CDE y proporciona el Panel frontal para poder lanzar las aplicaciones CDE. [Ver Anexo 1].

Desde el Panel frontal puede iniciar las siguientes herramientas de escritorio:

- Gestor de archivos - Aplicación de escritorio en la que puede crear, buscar y utilizar objetos del espacio de trabajo: archivos, carpetas y aplicaciones.
- Agenda - Aplicación que se utiliza para planificar citas y tareas, definir recordatorios, realizar e imprimir listas de citas y tareas, examinar otras agendas y planificar citas en grupo.
- Aplicación de correo - Herramienta para enviar, recibir y gestionar los mensajes de correo electrónico (email).
- Editor de textos - Editor para crear y revisar documentos ASCII tales como memorandos, mensajes de correo o archivos de recursos.
- Gestor de aplicaciones - Contenedor para las aplicaciones y otras herramientas disponibles en su sistema.
- Gestor de estilos - Herramienta para personalizar la apariencia del escritorio, sus colores, fondo y tamaño de font
- Terminal - Ventana en la que se introducen comandos del sistema operativo, se ejecutan comandos UNIX y se copian y pegan textos.
- Editor de iconos - Herramienta para crear archivos de imágenes de mapa de pixels y de mapa de bits para utilizarlas como iconos o fondos.
- Visualizador de imágenes - Herramienta que se utiliza para visualizar, editar, imprimir y convertir los formatos de archivo de las imágenes monocromas y en color, así como los documentos de varias páginas tales como los archivos PostScript.
- Menú del espacio de trabajo - Menú emergente que contiene elementos para gestionar el espacio de trabajo del escritorio
- Ayuda en línea - Sistema de información de ayuda que se utiliza para examinar la información disponible sobre las diversas aplicaciones de escritorio. [Ver Anexo 2].

Hasta el año 2000, CDE era considerado el escritorio estándar para los sistemas UNIX, pero al mismo tiempo, escritorios de SWL como KDE y Gnome fueron ganando madurez y convirtiéndose en partes esenciales de la plataforma GNU/Linux. Hacia el 2001, Hewlett-Packard y Sun Microsystems anunciaron que reemplazarían CDE por Gnome para sus sistemas UNIX.

Aún así, las versiones de Sun Solaris 10 (al menos hasta la 3/05) aún lo incluyen y representa una alternativa estable para el usuario que mantiene un uso específico de su estación. La última versión de CDE incluye características nuevas en cuanto a productividad, interoperatividad y gestión de equipos de escritorio.

1.3.2 Xfce.

Por otra parte, Xfce (éxfeis) es un entorno de escritorio ligero para sistemas tipo UNIX como Linux. Se configura íntegramente con el ratón (mouse) y que exige poca memoria para ejecutarse, por lo que es recomendado para computadoras con pocos recursos. Provee además, un marco de trabajo para el desarrollo de aplicaciones.

El nombre Xfce originalmente provenía de **XForms Common Environment (XFce)**, pero debido a los grandes cambios en el código, ya no usa el kit de herramientas de XForms, como originalmente lo hacía. El nombre sobrevivió, pero ya no se indica como XFce sino Xfce. Los desarrolladores están de acuerdo en que el nombre carece de significado actualmente, aunque se le suele desglosar como **X Free Cholesterol Environment (Xfce** entorno X libre de colesterol) en referencia al poco consumo de memoria que realiza y a la velocidad con que se ejecuta al no tener elementos superfluos a diferencia de otros entornos de escritorio más grandes.

Está basado en la librería The **GIMP ToolKit** (conjunto de rutinas para GIMP **GTK**), la cual contiene los objetos y funciones para crear la interfaz gráfica de usuario. Maneja widgets (pequeña aplicación o programa, enfocado en dar fácil acceso a funciones frecuentemente usadas y proveer de información visual) como ventanas, botones, menús, etiquetas, deslizadores, pestañas, etc, y además es compatible con los temas de Gnome. Es fácil de configurar, muy estable, incluye manejador de sesiones, administrador de archivos, un panel, calendario y un panel de control. Es sencillo pero a la vez muy útil.

A diferencia de Gnome y KDE, Xfce no incluye aplicaciones por defecto, lo que lo hace extremadamente ligero. Incluye muchas applets que te permiten agregarle funcionalidad, como controles para Winamp, para el sonido, indicadores de red y de uso del sistema, entre otros.

Algunas de las características de Xfce son:

- Un escritorio sencillo y fácil de utilizar
- Completamente configurable vía ratón con arrastrar y soltar, etc.
- Panel principal similar al de CDE, con menos, “applets” y accesos directos a las aplicaciones.
- Gestor de ventanas integrado, gestor de ficheros, gestor de sonido, módulo de compatibilidad con Gnome, y otras mucho más.
- Aspecto configurable (desde que utiliza GTK).
- Rápido, ligero y eficiente: ideal para máquinas viejas/lentas o máquinas con limitaciones de memoria.

Xfce utiliza como predeterminado el gestor de ventanas Xfwm que provee su propio gestor de configuración y soporta atajos de teclado y es completamente modificable gráficamente. Xfce se parecía en sus inicios al entorno de escritorio CDE, pero fue alejándose notablemente debido a que fue reprogramado nuevamente desde cero, y a diferencia de sus anteriores versiones, ahora cuenta con un sistema modular pudiendo gestionar un sistema de tipo multihead de manera bastante sencilla.

Dentro de los componentes de Xfce se encuentran:

- Mousepad, editor de texto simple.
- Terminal, emulador de terminal.
- Thunar, nuevo gestor de archivos, reemplazo del antiguo Xffm.
- Xarchiver, frontispicio para programas de compresión de datos.
- Xfburn, programa para grabar CD/DVD.
- Xfmedia, reproductor multimedia basado en xine.
- Xfprint, gestor de impresoras.
- Xfwm, gestor de ventanas.

Xfce no es tan fácil de encontrar en las distribuciones de software libre comparado con otros entornos de escritorios como KDE y Gnome. Pocas distribuciones lo incluyen como el entorno predefinido, aunque muchas como Debian y Fedora Core lo incluyen como componente opcional.

Otra de las desventajas de Xfce, y que ha provocado que no sea el seleccionado para su utilización en el sistema, es que no gestiona completamente el escritorio, debido a que es un entorno ligero. Ejemplo de esto lo demuestra la creación de accesos directos y la opción de (copy&page) copiar y pegar; que son acciones sencillas y fundamentales para facilitar el trabajo de los niños en los ordenadores; demostrando que no es el óptimo para su uso debido a que los usuarios finales de Tocororo – Desktop son ellos.

1.3.3 KDE.

En 1996 nace el proyecto **K Desktop Environment (KDE)** de las manos de Matthias Ettrich (creador de LyX) y otros hackers, quienes buscaban crear una interfaz gráfica unificada para sistemas UNIX. KDE es el entorno de escritorio más popular. Provee muchas aplicaciones básicas, una suite de oficina y un framework para el desarrollo de aplicaciones; es otro de los modernos entornos de escritorio gráfico e infraestructura de desarrollo para sistemas UNIX, en particular para LINUX.

La inspiración inicial para crear KDE provino de CDE, diseñado en cooperación por los proveedores de sistemas UNIX tales como IBM y Sun Microsystems, y en aquellos tiempos sólo funcionaba en los ordenadores que ellos mismos vendían. El diseño de CDE se hizo utilizando las bibliotecas Motif del sistema X Window suministradas por el Open Group (el que administraba inicialmente las X Window), lo que implicaba que cualquier fabricante que quisiera entregar su producto con CDE debía usar una licencia de Motif. Éste hecho era inaceptable en el entorno de SWL, así que escribieron KDE (nótese el cambio de C por K) para disponer de una alternativa a CDE. El proyecto KDE fue patrocinado por varios proveedores y aficionados admiradores de SWL, como Caldera, Delix, O'Reilly Associates y SuSe.

En 1998 se publicó KDE 1.0. Esta versión contenía un panel (barra de tareas y lanzador de aplicaciones), un escritorio sobre el cual dejar iconos, un administrador de archivos (Kfm) que permite inspeccionar directorios y realizar operaciones tales como editar, copiar, eliminar y renombrar archivos, lanzando automáticamente la aplicación correspondiente y un gran número de utilidades. KDE 2.0, lanzado en el año 2000, fue reescrito casi por completo. Esta versión incluía Konqueror, además de muchas nuevas tecnologías con el

objetivo de mejorar la integración entre aplicaciones. KDE 3.0 fue publicado en el año 2002, y es la evolución de KDE 2.

Una de las principales características de KDE es que todos sus componentes pueden ser configurados en mayor o menor medida por el usuario, los más avanzados, pueden incluso editar archivos de configuración manualmente, y obtener así un mayor control sobre el comportamiento, logrando que no sea del modo predefinido del sistema. De hecho, esa es la idea esencial del proyecto KDE, “crear un entorno gráfico que permita a todos los usuarios adecuar el sistema a sus gustos y comodidades”. Esto no impide que KDE resulte fácil de usar para nuevos usuarios, detalle al que no se resta importancia.

La interfaz gráfica de KDE recuerda a la Macintosh por sus iconos cristalizados y utiliza mucho el color azul, además de iconos animados al ejecutar cada aplicación, posee solo una barra inferior que al igual que Windows tiene tanto los programas como el alojamiento de ventanas. También al igual que Windows que utiliza el Explorer como navegador para acceder a los archivos, KDE utiliza Konqueror, su propio navegador para acceder a los archivos. También al igual que Gnome existen múltiples herramientas como: calculadora; editor de texto sencillo; calendario, procesador de palabra, hoja de cálculo, un excelente reproductor multimedia (Amarok) y muchas otras.

KDE es un entorno de escritorio contemporáneo y fácil de usar. Algunas de las cosas que brinda al usuario son:

- Un bello escritorio contemporáneo.
- Un escritorio completamente transparente en red.
- Un sistema de ayuda integrado que facilita un adecuado y consistente acceso a la ayuda durante el uso del escritorio KDE y sus aplicaciones.
- Un aspecto consistente de todas las aplicaciones KDE.
- Menús y barras de herramientas estándar, teclas programables, esquemas de color, etc.
- Internacionalización: KDE está disponible en más de 40 idiomas.
- Un diálogo de configuración del escritorio centralizado y consistente.
- Un gran número de aplicaciones KDE útiles.

Este entorno de escritorio es muy amigable, tiene una apariencia muy profesional, funcional y configurable. Las aplicaciones básicas incluyen el manejador de ventanas (kwm), un administrador de archivos, una terminal, un sistema de ayuda y un panel de control. Además incluye aplicaciones de Gráficos, Red, Multimedia, Juegos y Desarrollo. Tiene una Suite de oficina que contiene un Procesador de Palabras, Hoja de Cálculo, Software para hacer presentaciones, un organizador y todo lo demás que se pueda convertir en básico.

KDE se basa en el principio de la personalización. Las opciones más comunes son accesibles en su mayoría desde menús y diálogos de configuración. La apariencia de KDE es configurable en varios niveles. Tanto el gestor de ventanas (llamado Kwin) como los controles (botones, menús y otros) utilizan "estilos" intercambiables, que definen cada aspecto de su apariencia. Es por este motivo que KDE no mantiene una única apariencia entre versiones, sino que se opta por aquella más ampliamente aceptada en el momento de cada nuevo lanzamiento.

KDE permite el uso de temas tanto de iconos como de *look and feel*. Existe un amplio número de aplicaciones para KDE desde pequeñas applets hasta programas muy completos y complejos; está escrito casi exclusivamente en C++, un lenguaje derivado del lenguaje de programación C con algunas funcionalidades añadidas, en especial en cuanto a la programación orientada a objetos. A pesar de las críticas en contra de este lenguaje (inicialmente menos maduro), su adopción por el proyecto KDE ha resultado en un desarrollo más dinámico y ciclos de lanzamiento más breves, a la vez que ha permitido producir programas eficientes en menor cantidad de líneas de código que las requeridas para iguales tareas empleando lenguajes de programación estructurados (por ejemplo: C).

El gran problema fue que los creadores de KDE decidieron elegir como soporte gráfico básico a un paquete de ventanas y artilugios gráficos (widgets) llamado Qt; que es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario y fue creada por la compañía noruega TrollTech™, que tenía algunos inconvenientes en su licencia pues no estaba amparada bajo ninguna de SWL.

Se daba, por tanto, la circunstancia de que a pesar de que las aplicaciones de KDE estaban licenciadas bajo la **General Public License (GPL)** que fue creada por la Free Software Fundación a mediados de los 80, y está orientada principalmente a proteger la libre

distribución, modificación y uso de software, su propósito es declarar que el software cubierto por esta licencia es SWL y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios. Enlazaban con esta biblioteca de manera que se hacía imposible su redistribución.

Consecuentemente, se estaba violando una de las cuatro libertades del SWL enunciadas por Richard Stallman en su Manifiesto del SWL y mencionadas anteriormente. Algo que actualmente ha sido corregido, pero que en aquellos tiempos tuvo su efecto colateral en la formación del proyecto Gnome, al que muchas distribuciones de SWL (Ubuntu, Red Hat, Yellow Dog, Mandrake) se han adherido prefiriéndolo antes que KDE como su entorno de escritorio predeterminado.

1.3.4 Gnome.

GNU Network Object Model Environment (Gnome) es un entorno de escritorio de SWL para sistemas operativos de tipo Unix bajo tecnología X Window que tiene muchas aplicaciones que utilizan su biblioteca GTK+, también es líder como escritorio y como plataforma de desarrollo tanto para UNIX como para GNU/Linux, es uno de los entornos de escritorios más usado, fue creado con la idea de proporcionar un conjunto de aplicaciones amigables y un escritorio potente pero fácil de utilizar, dirigido a todo tipo de usuarios de ordenadores. Gnome es el manejador de escritorios oficial del proyecto GNU y lo fundaron dos mexicanos, Miguel de Icaza y Federico Mena para crear un entorno de escritorio completamente libre para sistemas operativos libres, en especial para GNU/Linux.

Se han propuesto desde los inicios de Gnome varias formas de traducirlo al español, pero no se ha encontrado ninguna que haya satisfecho a todos, aunque Gnome-es colabora traduciendo el escritorio, aplicaciones y documentación al castellano, para eliminar la barrera del inglés. Sin embargo, de su nombre podemos ver que Gnome es parte del proyecto GNU y, por tanto, SWL (algunas veces conocido como Open Source). En la actualidad, todo el código contenido en Gnome debe de estar bajo licencia GNU GPL o GNU LGPL.

Desde sus principios Gnome ha sido 100% SWL y amigable, ha ido creciendo paulatinamente con sus reiteradas publicaciones, comenzando en 1997 como una alternativa completamente libre a KDE hasta la actualidad. En noviembre de 1998 ya se

lanzó la versión 0.99, pero la primera realmente popular distribuida prácticamente por cualquier distribución de GNU/Linux sería Gnome 1.0, en marzo de 1999. Cabe destacar que la experiencia de esta primera versión estable de Gnome no fue muy satisfactoria, ya que muchos la consideraron como llena de erratas críticas. Por eso, Gnome October (Gnome 1.0.55) es tratada como la primera versión del entorno de escritorio GNOME realmente estable.

Como se puede observar, con Gnome October se intentó evitar versiones de publicación numeradas para no entrar en una "carrera" de versiones con KDE. Y actualmente se encuentra disponible en más de 48 idiomas y su última versión es la 2.22. El equipo de desarrolladores de Ximian han desarrollado Ximian Gnome que además de hacer el entorno más amigable, viene con un paquete con muchas utilidades y entre todas estas hay 2 que lo hacen interesante: Evolution y Red Carpet. El primero es un organizador personal, tiene la posibilidad de mandar y administrar emails, tiene una agenda para organizarse la vida y hasta es una agenda telefónica. Red Carpet es un programa que permite navegar por directorios de SWL, donde se puede tanto como bajar nuevo software, como actualizar el ya existente.

Su interfaz gráfica recuerda un poco a Windows 98 o XP, en el sentido de que esta más orientado a lo funcional que a un diseño bonito, Al entrar a su escritorio veremos dos barras una inferior y otra superior en donde están todas las aplicaciones y programas, y en la inferior se ubican las ventanas de las aplicaciones que utilizamos, sus iconos con sus descripciones hace muy intuitivo y rápido para acceder a las aplicaciones, también permite usar más de un escritorio independiente que puede pasar aplicaciones de un espacio de trabajo a otro, pues basta arrastrarlas con el ratón.

El proyecto de Gnome es de alta calidad y es soportado por grandes compañías como HP, Novell, MandrakeSoft y RedHat. Gnome tiene una de las comunidades más grandes que existe. Dos factores llevaron a la creación de dicho proyecto en 1997:

- La elección de la biblioteca Qt (por parte del proyecto KDE), que por aquel entonces poseía una licencia incompatible con la GPL de GNU, aunque libre: la QPL.
- La importancia del lenguaje C++ para el desarrollo de KDE.

Aunque ya se ha normalizado la incompatibilidad de las bibliotecas, siguen existiendo rivalidades entre ambos proyectos, que en la actualidad se consideran beneficiosas generalmente y existe, de hecho, una constante cooperación e inspiración mutua. Para su funcionamiento Gnome utiliza las librerías GTK+ que son la base de todo programa para Gnome y necesita ejecutarse con un Window Manager (por default sawfish).

Desde sus inicios se ha utilizado la biblioteca de controles gráficos GTK, originalmente desarrollada para el programa The GIMP. Al igual que KDE, tiene paquetes de software ya incluidos para las tareas básicas y la interacción con el sistema. Existe un gran número de aplicaciones para Gnome y también una amplia cantidad de temas e iconos para el mismo.

Los componentes principales de Gnome son:

- Un entorno de usuarios basado en ventanas.
- Una rica colección de herramientas, bibliotecas, y componentes para desarrollar potentes aplicaciones en Unix.
- Un conjunto de aplicaciones de productividad ofimática.

Los objetivos del proyecto Gnome son:

- La libertad para crear un entorno de escritorio que siempre tendrá el código fuente disponible para reutilizarse bajo una licencia de SWL.
- El aseguramiento de la accesibilidad, de modo que pueda ser utilizado por cualquiera, sin importar sus conocimientos técnicos y discapacidad física.
- Hacer que esté disponible en muchos idiomas.
- Un ciclo regular de liberaciones y una estructura de comunidad disciplinada.

Gnome está pensado tanto para usuarios finales como para desarrolladores. Este escritorio cuenta con iconos para acceder rápidamente a aplicaciones y con menús que le permiten iniciar programas. El escritorio Gnome es configurable: puede configurar los menús, los iconos, las tipografías, el fondo, el protector de pantalla, el tema, el administrador de ventanas, sonido, la interacción con las ventanas y muchos otros detalles de acuerdo a su gusto.

Entre los menús existen múltiples herramientas: calculadora, editor de texto sencillo, gnotepad; calendario gnomecal, procesador de palabra, hoja de cálculo, reproductor multimedia y un excelente navegador (Firefox), un excelente suite de ofimática (Open Office) y un completo cliente de correo (Evolution) un cliente de mensajería instantánea (Gaim) entre muchas otras.

Gnome es uno de los proyectos más populares que existen de SWL, es gratis, usable, con herramientas de accesibilidad e internacionalizables. Te permite, mediante el uso de GTK crear aplicaciones con el *Look and Feel* de Gnome y puedes desarrollar en C, C++, Python, Perl, Java e incluso C#.

1.4 Análisis Comparativo.

Elegir el mejor escritorio es como tener una opinión: todo el mundo tiene una. Incluso aquellos que llevan muchos años y han usado muchos gestores de ventanas y escritorios diferentes, pocas veces se concuerdan únicamente con uno solo, pues cada cual tiene sus características que lo identifican y dan al usuario una decisión de gustos personales, ya sea por funcionalidad, rendimiento, diseño u otro, por lo que hay que ser razonable en este sentido. Por tanto hay que tener en cuenta las propias preferencias de cada uno. Ninguna elección es mala, ni buena, solo es la mejor opción para lo que se busca.

En este sistema se decidió en particular el uso de Gnome pues es el ideal porque además de ser más funcional, más discreto, elegante, muy intuitivo y de muy buen porcentaje de usabilidad, [Ver Anexo 3], hace recordar a Windows que se está más acostumbrado a usar y lo identifica una característica que lo hace especial y que los demás no cumplen. Es el caso de ser producido por la UCI, debido a que constituye la línea de producción de Nova, distribución que se perfila como candidata para su implementación por el MINED. Siendo además la distribución cubana más completa hasta el momento, que cuenta con un equipo de soporte técnico comprometido con su desarrollo. Se puede apreciar entonces que Gnome cumple todas las restricciones que exige el entorno de escritorio que integrará un gestor de configuración de interfaces para sustituir a los sistemas privativos que existen actualmente en el país y en especial en las escuelas primarias.

1.5 Tocororo – Desktop.

Varios son los esfuerzos que se han hecho en los Sistemas GNU/Linux relacionados con la personalización, pero su enfoque es general, está destinado a una mayoría de usuarios y no a un tipo de usuario en particular; los esquemas que se proponen de diseño de las interfaces pueden no ser los más óptimos por razones lógicas, sobre todo en personas no acostumbradas al mundo de la informática o que simplemente se inician en este sector. Así existen experiencias de diseño de escritorios en el que los elementos se parecen más a la vida real, las personas tienen por tanto una referencia natural para el trabajo con estos, haciéndoles más fácil su entendimiento y aprendizaje, pero son esfuerzos aislados y ninguno se enfoca específicamente a los niños, no tienen en cuenta por tanto sus criterios, carecen de su inocencia o la simplicidad con la que ven la complejidad de nuestro mundo. Y es que como diría una especialista del diseño de gráficos por computadora, las herramientas deben estar hechas para asemejarse a las personas, no para las personas parecerse a ellas.

Esta es la principal idea de Tocororo -Desktop, “llevar la informática a los niños, no los niños a la informática”, es decir, lograr que esta ciencia se convierta en un medio de aprendizaje que se adecuó a los sueños y fantasías propios de la infancia.

1.5.1 Análisis internacional de Migración a SWL dedicada a niños .

A nivel internacional se han dado los primeros pasos, como en España, donde la Junta de Extremadura ha decidido una migración total de todos los sectores de la vida de esa región. No obstante las experiencias se enfocan más a los ambientes de multimedia que a la estética de los entornos de escritorio, deficiencia que aprovecha sin embargo el proyecto **One Laptop Per Child (OLPC)**, en el español **Una Laptop Por Chico (ULPC)**, del magnate de la informática Nicholas Negroponte. En términos coloquiales también es conocido como El portátil de 100 dólares, y consiste en otorgar una computadora portátil de 100 dólares, con el propósito de proporcionar a cualquier niño en el mundo acceso y conocimiento a las tecnologías de la información como formas modernas de educación. El proyecto cuenta con apoyo de Google, AMD, Red Hat, News Corp, Brighstar y la colaboración de otras empresas. [Negroponte, 2006]⁴.



El sistema está basado en GNU con núcleo Linux y un sistema de escritorio ultra-simple en el que las ventanas siempre se encuentran maximizadas. Hay controles alrededor de la ventana, en forma de marco, que pueden mostrarse u ocultarse mediante la presión de una tecla. La otra pieza clave consiste en el empleo del famoso entorno educativo Squeak que es un mundo de objetos interactivos con vida propia gracias al lenguaje SmallTalk (el propio Squeak está escrito en este lenguaje), mediante el cual niños de cualquier edad aprenden conceptos gracias a la experimentación directa con gráficos de tortuga y multimedia.

Estos computadores portátiles serán vendidos inicialmente a los gobiernos y entregados a los niños en las escuelas bajo el principio "una computadora para cada niño". Con algunas ideas como la concepción de un panel triangular.

Como Tocaroro-Desktop es un proyecto que concibe adecuar las interfaces gráficas de los sistemas GNU/Linux a un tipo de usuario en particular, se habla por tanto de la configuración de todas ellas como un todo reunidas en una aplicación. Existen en el mundo aplicaciones como el Administrador de Configuración de Gnome (gconf), que ofrecen una idea general de todos los elementos del sistema al que normalmente tiene acceso cualquier usuario, y no interfaces solamente, a lo que se une el hecho de que el estándar de diseño del escritorio de este Entorno no sea el más ideal para los fines que se pretende. Otros como el Beryl y CompizFusion complementan las funcionalidades gráficas, la alternan con dinamismo, pero siguen sin poder cambiar el estándar que ya se ha creado y por el cual se rigen todos, no es por tanto el modelo que se debe de usar para Tocaroro - Desktop, si bien puede servir como hasta ahora de complemento.

1.5.2 Análisis nacional de Tocaroro – Desktop.

La presencia en nuestro país de los sistemas libres es algo totalmente nuevo, no existe por tanto una experiencia al respecto fuera del ámbito de la UCI que pueda tomarse como referencia para el desarrollo de Tocaroro-Desktop, menos aún en la educación, sector en el que todos sus planes de estudio se enfocan al aprendizaje desde edades tempranas basado en Windows, que es un sistema privativo al que estamos limitados por el bloqueo que tiene sobre nuestro país una de las más grandes potencias, que además lidera el desarrollo de la informática en el mundo.

1.6 Metodologías Ágiles.

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte se tienen las más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán. Estas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos.

Las metodologías ágiles forman parte del movimiento de desarrollo ágil de software, que se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un proyecto. Se le denomina ágil como la habilidad de responder de forma versátil al cambio para maximizar los beneficios. Las metodologías ágiles varían en su forma de “responder al cambio”, pero en general comparten características como las siguientes:

- Usar procesos de construcción iterativos
- Entregar software funcional lo más pronto posible.
- Privilegiar el valor de la gente sobre el valor del proceso.
- Fortalecer la comunicación y la colaboración.

Las metodologías ágiles, dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. De forma que una metodología ágil es la que tiene como principios que:

- Los individuos y sus interacciones son más importantes que los procesos y las herramientas.
- El software que funciona es más importante que la documentación exhaustiva.
- La colaboración con el cliente en lugar de la negociación de contratos.
- La respuesta delante del cambio en lugar de seguir un plan cerrado.

Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software, y a

la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales.

Teniendo en cuenta las características propias del sistema en cuestión, se propone el uso de dos de las muchas metodologías ágiles existentes, Scrum para la parte de planificación, y XP para la parte de desarrollo. La elección de estas metodologías para el desarrollo del sistema se debe fundamentalmente a que el sistema que se propone exige muchos cambios rápidos de requisitos, además de que los que posee son imprecisos, por lo que existe un alto riesgo técnico en sus implementaciones.

1.6.1 Scrum.

Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos, y es además una forma de gestionar proyectos de software, no es una metodología de análisis, ni de diseño, es una metodología de gestión del trabajo. Sus principales características se pueden resumir en. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días, el resultado de cada sprint es un incremento ejecutable que se muestra al cliente. Y las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

Scrum no está definida como metodología independiente, sino como un complemento a otras metodologías como lo son XP y **Microsoft Solution Framework (MSF)**. Como método, enfatiza valores y prácticas de gestión, sin pronunciarse sobre requerimientos, implementación y demás cuestiones técnicas; de allí su deliberada insuficiencia y su complementariedad.

En Scrum se fomentan valores como:

- Equipos auto-dirigidos y auto-organizados. No hay Director que decida; la excepción es el Scrum Master que debe ser 50% programador y que resuelve problemas, pero no manda.
- Una vez elegida una tarea, no se agrega trabajo extra. En caso que se agregue algo,

se recomienda quitar alguna otra cosa.

- Encuentros diarios con las tres preguntas siguientes:
 - ✓ ¿Qué es lo que se hizo el día anterior?
 - ✓ ¿Qué es lo que se va a hacer hoy?
 - ✓ ¿Qué impedimentos tengo para realizar mi trabajo?

- Iteraciones de treinta días; se admite que sean más frecuentes.
- Demostración a participantes externos al fin de cada iteración.
- Al principio de cada iteración, planeamiento adaptativo guiado por el cliente.

1.6.2 XP.

XP es la más destacada de las metodologías ágiles de desarrollo de software. Al igual que los demás, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsión.

Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Dentro de las características de XP están:

1. Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
2. Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
3. Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera el código es revisado y discutido mientras se escribe es más importante que la posible pérdida de productividad inmediata.

4. Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
5. Corrección de todos los errores antes de añadir nueva funcionalidad.
6. Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.

1.7 Tecnologías Utilizadas.

Tocororo –Desktop será implementado utilizando los lenguajes de XML y Python-C++/GTK; los que serán programados sobre el IDE Anjuta. Además de que se modelará sobre la Herramienta Case Visual Paradigm y se utilizarán como herramientas de diseño Gimp, Blender e Inkscape.

1.7.1 Lenguajes.

XML.

Extensible Markup Language (XML) es el lenguaje que se emplea para la configuración gráfica de todos los entornos de escritorio, temas de **Gnome Display Manager (GDM)**. XML es el lenguaje natural en el que se expresa la mayor parte de las configuraciones de un entorno de escritorio programado sobre Gtk. Interfaces como el GDM, el Screensplash o el gnome-panel lo emplean para mostrar sus contenidos.

Python-C++/GTK.

En principio **C/C++** constituyen los lenguajes fundamentales en la programación para el escritorio. Sobre ellos se desarrolló el **Application Programming Interface (API Gtk)** que emplearemos para la aplicación.

Las razones para incluir Python se enfocan en direcciones aparentemente distintas. La primera de ellas se debe a su empleo cada vez más difundido en el desarrollo efectivo de aplicaciones, ofreciendo comodidad y sencillez en el trabajo. La segunda tiene que ver con

las herramientas a emplear, las cuales tiene en gran por ciento su aporte indiscutible o han migrado a esta línea de desarrollo en los últimos tiempos.

1.7.2 IDE.

Anjuta.

Anjuta es hasta el momento el único entorno de desarrollo que genera una estructura compatible con las normas de un proyecto GNU. Es, por tanto, la alternativa, además de que los lenguajes antes mencionados, especialmente Python, se integran a este perfectamente.

Aún cuando Glade resulta para muchos bastante trabajoso por el hecho de no ofrecer integración o forme parte de un **I**ntegrated **D**rive **E**lectronics (**IDE**) en particular, no divisamos hasta el momento un diseñador o IDE que pueda integrarse a Gtk mejor.

1.7.3 Herramienta Case.

Visual Paradigm.

Esta herramienta **C**omputer **A**ided **S**oftware **E**ngineering (**CASE**) es seleccionada para el modelado de Tocororo – Desktop, y esto se debe en gran medida a que posee un paquete para el trabajo con metodologías ágiles, además de utilizar **L**enguaje **U**nificado **d**e **M**odelado (**UML**). Visual Paradigm, es un producto que facilita la organización para el diseño visual de diagramas.

Visual Paradigm ofrece:

- Entorno de creación de diagramas para UML 2.0.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas

Esta herramienta ayuda al equipo de desarrollo de software, a documentar y cubrir el proceso de modelado, construcción y desarrollo de software, acelerando la contribución individual y a nivel de proyecto de cada equipo de desarrollo, que además soporta lenguajes como: Java, C++, CORBA Interface Definition Language (IDL), Hypertext Pre-processor (PHP), Schema y XML. Además, una de las nuevas ventajas de este software es que permite la importación y exportación de ficheros XML, que es uno de los lenguajes utilizados en el desarrollo de Tocororo – Desktop.

1.7.4 Herramientas de Diseño.

Gimp.

GNU Image Manipulation Program (GIMP) es un programa de edición de imágenes, que se utilizó en el desarrollo de Tocororo – Desktop para la creación de prototipos y de imágenes que conformarán los temas insertados en el sistema. Fue elegido principalmente por ser un programa libre y gratuito, englobado en el proyecto GNU y disponible bajo la licencia GNU/GPL. Además existen versiones de este software totalmente funcionales para Windows, para Mac OS X y para otros sistemas operativos, convirtiéndolo en el programa de manipulación de gráficos disponible en más sistemas operativos. Gimp existe en español además de en la lengua de desarrollo original, el inglés, y también, en francés, italiano, alemán, ruso, sueco, noruego, coreano, neerlandés y en algunas otras lenguas adicionales más.

Gimp fue desarrollado como una herramienta libre para trabajar con imágenes y se ha convertido en una alternativa libre al Photoshop, el dominador de las aplicaciones industriales de impresión y gráficas, aunque posee una interfaz muy diferente. GIMP lee y escribe la mayoría de los formatos de ficheros gráficos, entre ellos jpg, bitmap, gif, png, pcx, tiff, y los de Photoshop, además de poseer su propio formato de almacenamiento de ficheros, xcf. También es capaz de importar ficheros en pdf, e imágenes vectoriales en formato svg creadas, con Inkscape, que es otra de las herramientas que se utilizan en el desarrollo del sistema.

Además de un uso interactivo, Gimp permite la automatización de muchos procesos mediante macros. Para ello incluye un lenguaje llamado Scheme para este propósito.

También permite el uso para estas tareas de otros lenguajes como Perl, Tool Command Language (Tcl) y Python que es uno de los lenguajes sobre los que se programa Tocororo - Desktop.

Inicialmente, la biblioteca de controles gráficos GTK, fue desarrollada para Gimp, y dio origen al entorno de ventanas de Gnome, además de que esta biblioteca se utiliza para el trabajo en el sistema de gráficos programados en Python.

Blender.

Aunque en sus inicios Blender fue desarrollado como una aplicación propia del estudio de animación holandés NeoGeo, el principal autor, Ton Roosendaal, fundó la empresa "Not a Number Technologies" (NaN) en junio de 1998 para desarrollar y distribuir el programa. Luego la compañía quedó en bancarota en 2002 y los acreedores acordaron ofrecer Blender como un producto de SWL y gratuito bajo los términos de la GNU GPL a cambio de 100.000 €. A partir del 2003 se hizo público el código fuente.

Blender es un programa multiplataforma, dedicado especialmente al modelado y creación de gráficos tridimensionales. Actualmente es compatible con todas las versiones de Windows, Mac OS X, Linux, Solaris, FreeBSD e IRIX. Es libre, gratuito y con un tamaño de origen realmente pequeño comparado con otros paquetes de 3D, dependiendo del sistema operativo en el que se ejecuta. Tiene capacidad para una gran variedad de primitivas geométricas, incluyendo curvas, mallas poligonales, vacíos, NURBS, metaballs.

Además, Blender utiliza el lenguaje Python, uno de los principales programas en el desarrollo del sistema, para automatizar o controlar varias tareas. Blender acepta formatos gráficos como TGA, JPG, Iris, SGI, o TIFF. También puede leer ficheros Inventor.

Inkscape.

Inkscape se inició en 2003 como una bifurcación (fork) de código del proyecto Sodipodi. Este último, desarrollado desde 1999, estaba a su vez basado en Gill, el trabajo de Raph Levien. Desde la bifurcación, Inkscape, entre otras cosas, ha cambiado de usar el lenguaje de programación C al C++; cambió al conjunto de herramientas GTK+ implementadas con C++ (gtkmm); rediseñó la interfaz de usuario y añadió gran cantidad de nuevas

funcionalidades. Esta implementación del estándar **Scalable Vector Graphics SVG** ha mostrado mejoras progresivas, pero aún está incompleta. Inkscape es actualmente un proyecto del Google Summer of code.

Inkscape es una herramienta de dibujo libre y multiplataforma para gráficos vectoriales SVG. La principal motivación de Inkscape es proporcionar a la comunidad una herramienta de SWL de dibujo SVG que cumpla completamente con los estándares SVG, **Cascading Style Sheets (CSS2)** y XML, que es el lenguaje utilizado para el trabajo con gráficos en la creación de temas de GDM, BootSplash y otros pertenecientes a Tocororo - Desktop.

Inkscape se encuentra desarrollado principalmente para el sistema operativo GNU/Linux, pero es una herramienta multiplataforma que funciona en Windows, Mac OS X, y otros sistemas derivados de Unix. Es una aplicación que tiene buena compatibilidad multilingüe, particularmente para scripts complejos.

Una de las prioridades del proyecto de Inkscape es la consistencia de su interfaz y utilidad. Esto incluye los esfuerzos a seguir para desarrollar interfaces humanas de Gnome, la accesibilidad del teclado universal. Inkscape ha logrado un progreso significativo en la utilidad desde que el proyecto empezó. Además de que la interfaz antecesora de Inkscape fue la de Sodipodi, que era en parte basado en CoreIDRAW y Gimp, otra de las herramientas utilizadas en el trabajo con imágenes para el sistema Tocororo – Desktop.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.

2.1 Elementos gestionados por Tocororo – Desktop.

Cada entorno de escritorio se distingue por su aspecto y el tipo de usabilidad que disponen, aunque algunos tienden a imitar características de escritorios ya existentes. Desde el principio, el objetivo principal de Gnome ha sido proporcionar un conjunto de aplicaciones amigables y un escritorio fácil de utilizar, logrando que todo usuario de un computador pueda trabajar de forma cómoda y familiar en cualquier aplicación; desde escribir un texto o navegar por Internet, hasta diseños de diferentes tipos o realizar actividades complejas.

Las características más importantes de un entorno de escritorio son su alto grado de configurabilidad y las múltiples maneras que puede ofrecer para ejecutar tareas y poder proporcionar interoperabilidad entre sus elementos. Por lo general, una misma acción se puede ejecutar de más de una manera, detalles que no le faltan a Gnome.

Los usuarios de las diferentes computadoras siempre buscan la vía para configurarla a sus gustos y necesidades, cambiando las imágenes de fondo, los estilos de las ventanas y la forma de los paneles. En cada sistema existen herramientas que gestionan la configuración de cada una de estas partes que conforman el entorno de escritorio, pero dichos sistemas no son fáciles de utilizar, lo que provoca que personas con bajo conocimiento de informática no puedan personalizar su computador.

Tocororo – Desktop es un gestor de configuración de interfaces que facilita esta actividad del sistema, incluyendo la personalización de aspectos que pertenecen al entorno de escritorio y además de algunos elementos de la distribución que se utilicen, pero que es necesario incluir en el sistema para darle la posibilidad al usuario de configurarlos también. Es una aplicación de escritorio diseñada para que los niños configuren el sistema a su gusto, ya sea desde la imagen del Arranque Dual, hasta el fondo de pantalla que se tenga activado. En este capítulo se explicará en qué consisten cada uno de los elementos que incluye Tocororo – Desktop, como son el GrubSplash, el BootSplash, el GDM, el ScreenSplash y el Escritorio.

2.1.1 GrubSplash.

El Dual Boot o Arranque Dual son distintas formas de llamar a la capacidad de una computadora de poder tener más de un **Sistema Operativo (SO)** y poder iniciar con cualquiera de ellos. El Dual Boot se ha convertido en algo esencial, sobre todo para aquellos usuarios que comienzan a usar GNU/Linux o comparten la **Personal Computer (PC)** con otras personas que necesitan inevitablemente el uso de programas hechos para la plataforma del SO Windows.

Al encender una computadora que tiene instalado más de un sistema operativo, el asistente de arranque posibilitará al usuario seleccionar cuál de los sistemas instalados quiere utilizar para comenzar su carga. La capacidad de seleccionar el sistema a cargar es responsabilidad del *Cargador o Gestor de arranque (Boot Loader)*.

El **GRand Unified Bootloader (GRUB)** es un Gestor de arranque que permite elegir entre los diferentes sistemas operativos y/o las diferentes versiones de ellos, en el mismo disco duro. Fue desarrollado por el Proyecto GNU y en los momentos actuales viene preinstalado en la mayoría de las distribuciones de GNU/Linux modernas, es por ello que se toma de referencia para la explicación de las diferentes partes de los Dual Boot.

En la prisa y el afán de configurar GNU/Linux al gusto y necesidades de cada cual, se echa a rodar la imaginación buscando una mejor imagen de la PC. Atendiendo a que los usuarios del sistema son niños, es obligatorio prestar atención a la interfaz que se les mostrará y de nada vale configurar el BootSplash y la interfaz de los gestores y entornos de escritorio, si la vista que presenta el sistema, no es agradable a la vista de los infantes. Esta es la razón para incluir dentro del gestor de configuración de interfaces, esta parte del sistema.

¿Cómo crear un tema de GrubSplash?

EL GRUB puede cargar una imagen como fondo sin afectar las funcionalidades de este, pero dicha imagen debe cumplir ciertas condiciones. Se puede diseñar una imagen, escogerla del “almacén de fotos” o simplemente bajarla de Internet, sin embargo, para cuidar la calidad de esta, no se debe utilizar ninguna que contenga muchas gradaciones (sombras) de colores.

Lo primero que se necesita para realizar esta edición, es tener una imagen para poder poner como fondo de GRUB, pero la misma debe cumplir con cuatro requisitos esenciales:

1. Formato: ***.xpm**
2. Resolución: **14 colores**
3. Dimensión: **640x480 pixels**
4. Compresión: ***.gz [Ver Anexo 4].**

Para realizar la edición de un archivo de imagen se pueden emplear programas como GIMP, Adobe Photoshop, Corel Photo Paint o el programa de edición de imágenes que se prefiera. Es importante saber que no es una tarea sencilla el obtener un resultado satisfactorio, ya que la imagen debe tener una resolución de 14 colores, lo que es complicado lograr.

Luego de haber elegido la imagen, solo resta configurar el GrubSplash con la misma. Lo ideal es guardarla en alguna carpeta dentro del directorio `/boot/grub/` con un nombre que haga referencia a esta (Ej. `image`). Luego de guardada, se comprime y se copia para `/boot/grub/image` (si se siguió el ejemplo), para entonces comenzar a configurar el Grub, y se hace necesario modificar el archivo `menu.lst`, que será el encargado de cargar las modificaciones. [Ver Anexo 5]. Luego solo nos queda reiniciar el ordenador y disfrutar de la nueva vista de nuestra primera imagen de elección de SO.

Además de esta modificación de agregar imágenes al GRUB, también se puede modificar el tiempo de espera para la elección del SO. Esto se refiere a que el GRUB tiene configurado como tiempo de espera 5 segundos para la carga del SO como tiempo predeterminado; un plazo que en algunas ocasiones, resulta extremadamente corto. 30 segundos es un tiempo

razonable para poder elegir el SO con el que queremos trabajar, pero cada usuario puede poner el plazo de tiempo que guste.

Para esta modificación lo que se debe hacer es abrir una consola o terminal y autenticándose como administrador del sistema o usuario root en la mayoría de los casos, entrar en el archivo central del GRUB (menu.lst) para cambiar la opción de *timeout* por la cantidad de segundos que se desee para que espere el GRUB y, como todo cambio en este archivo, para quedar confirmado solo deberá guardar los cambios y reiniciar el ordenador.

Otra de las oportunidades de modificar nuestro entorno que nos da el GRUB es la de cambiar el título del menú del GRUB. Esta edición es útil si cambiamos el SO de default por otro, o si se quieren personalizar los textos que refieren a los SO del menú GRUB. Para ello solo es necesario autenticarse como root o administradores del sistema y en el archivo menu.lst modificar la opción de title por el título se visualizará. Los mismos pasos serían si lo que se quiere es cambiar los colores del menú GRUB, solamente se tendría que modificar los colores de la opción # Pretty colours y escribirle los colores deseados.

¿Cómo activar el tema diseñado?

Para activar el tema de GrubSplash que se ha diseñado solo se necesita guardar los cambios hecho en el documento menu.lst y reiniciar el ordenador así, cuando comience a cargar, se disfrutará de los cambios hechos.

2.1.2 BootSplash.

En el GNU/Linux en modo texto muestra en la consola líneas de texto que asiste al usuario en el proceso de arranque del sistema operativo. Bootsplash oculta esas líneas de texto y se apoya en componentes gráficos como una barra de progreso que familiariza a usuarios asiduos a otros sistemas operativos con interfaz gráfica con SWL sin perder sus experiencias anteriores. Igualmente puede configurarse para trabajar en modo detallado o "verbose" pulsando la tecla F2 y presentará los mensajes propios del proceso de arranque. Además de estos componentes, el BootSplash contiene animaciones realizables en aplicaciones de diseño gráfico como Gimp.

Un tema de boot splash es un directorio localizado en `/etc/boot splash/themes/debian`. Contiene tres directorios hijos, nombrados comúnmente animaciones, config, e images [Ver Anexo 6] donde se encuentran distribuidos sus componentes, que análogamente a como ocurría en la elaboración de un tema de GDM a través de la edición de un fichero con extensión xml, su gestión corre a cargo de `boot splash-1024x768.cfg` situado en config. ¿Por qué `boot splash-1024x768.cfg` y no cualquiera de las imágenes? Porque ese es el archivo de configuración que nos muestra de dónde añadir las dos imágenes.

¿Cómo crear un tema de Boot Splash?

Para hacer un tema de boot splash solo hay que tener presente que lo más importante de un splash son las imágenes de background y la barra de progreso de acceso. Lo que se explica a continuación es donde se ubican estas imágenes en el fichero xml, y luego solo habrá que cambiar en estas líneas de código sus nombres por los de las imágenes que se cambiarán.

Las imágenes boot splash y silent (imágenes background), son las q se visualizan en el inicio del Sistema y la de background del modo consola de este, son asignadas a las variables `jpeg` y `silentjpeg`, respectivamente, que permiten reconocerlas. [Ver Anexo 7]. Estas imágenes se sitúan en el subdirectorio images debido a que otra localización no es permitida.

Sin dudas lo más interesante del proceso de creación de un tema de boot splash es la barra de progreso, en la que más que su forma (la cual es rectangular) interesan, a los efectos de la configuración del fichero `.cfg`, sus colores y localización en la pantalla.

En un análisis del código del archivo xml que sigue las especificaciones de la `gui/greeter/greeter.dtd` [Ver Anexo 7], la línea precedida por la sentencia `box silent noover` identifica el fondo base de la barra, en tanto la sentencia `box silent inter` y `box silent` intervienen en el diseño de la animación: la primera define los colores de la parte superior de la barra animada, mientras que la segunda la inferior. Los valores, tal y como se nos presentan representan lo siguiente:

```
box silent "algo" <inicio en x ><inicio en y><final en x ><final en y>
```

El caso anterior es extensible también a los bordes de la barra de progreso, señalado su código en el recuadro azul [Ver Anexo 7]. Como puede verse en este caso, se pueden definir tantas líneas como se quiera con el color deseado, como la barra de progreso de la Distribución Ubuntu, de moda por estos tiempos.

¿Cómo activar el tema diseñado?

Una vez concluida la etapa de diseño del boot splash, activarlo es una tarea sencilla, basta con crear en el directorio `/etc/bootsplash/themes/debian` un enlace simbólico denominado `current` a la carpeta donde se encuentra el tema. Finalmente se hace en la consola `updateinitramfs -u` y listo.

Nada hay más tedioso para un usuario que tener siempre la misma configuración gráfica de sus sistemas. A tales efectos GNU/Linux ofrece la ventaja de hacerlo uno mismo, sin la necesidad de que otros decidan lo que debemos o no ver, algo que se aplica sobre todo a los sistemas operativos privativos como Windows, cuyas interfaces son esquemáticas y obedecen a los patrones definidos por su creador, no a los gustos y deseos del usuario final, siempre para ellos relegados a un segundo plano.

2.1.3 GDM.

El GDM es la imagen que da la posibilidad de loguearse y poder iniciar la sección por cierto usuario elegido. Una de las posibilidades con las que se encuentra al utilizar GDM es la de personalizarlo mediante templates o themes. Dentro de la siempre útil Gnome-Look, existe un apartado específico para ello que cuenta con casi un centenar de diferentes templates listos para ser usados en el sistema. Pero lo que se propone mostrar cómo se crea un tema de GDM para obtener una idea de lo que se implementa en este módulo del sistema.

Un GDM Theme está compuesto Básicamente por:

- Una imagen de fondo
- Una captura de pantalla que se muestran en la configuración del GDM
- Iconos (imágenes para la barra de iconos)

- Un Archivo xml, donde se realiza el tema definido. Este archivo permite la declaración de los marcos, las imágenes, el cuadro de inicio de sesión, marco de alerta, fondo, iconos en varios subtítulos.
- El GdmGreeterTheme.desktop archivo, que no es verdadero, archivo del escritorio, pero contiene en nombre del XML, captura de pantalla y otro tipo de información que se mostrará en la pantalla de configuración del GDM.

Los temas de GDM se crean en un fichero XML, que sigue las especificaciones de `gui/greeter/greeter.dtd`. Los ficheros de temas se encuentran en el directorio `/usr/share/gdm/themes`.

Dentro de este archivo el campo `Screenshot` apunta a un fichero de 200x150 de tamaño con el tema en acción (Si no existe no pasa nada, pero le hace la vida más agradable al usuario). El campo `Greeter` apunta a un fichero XML que contendría la descripción del tema.

Una vez que se tiene el tema listo e instalado, puede probarse con el script `gdmthemetester` (este guión asume que se tiene instalado `Xnest`). Toma dos argumentos, el primero el entorno que va a usarse. Será uno de `console`, `console-timed`, `flexy`, `remote-flexi` o `xdmcp`. Donde `console` es el login estándar de consola, `console-timed` es una consola con tiempo máximo definido, `flexi` es para algún servidor local flexible, `remote-flexi` es para un servidor flexi que no es local (como un servidor `flexiserver` de `Xnext` ejecutándose en un display remoto) y `xdmcp` es para conexiones remotas a través de `xdmcp`. El segundo argumento es el nombre del tema. Así que para comprobar el aspecto del tema en modo `xdmcp` con el tema `circles`, tendría que ejecutar:

```
gdmthemetester xdmcp circles<br />
```

Hay que asegurarse de ejecutar todos los entornos con su tema, y comprobar el aspecto del aviso de mayúsculas pulsadas presionando la tecla `Bloq Mayúscula`. Esta es una buena manera de hacer una captura de pantalla, sólo hay que hacerla de la ventana `Xnext`. Esto puede hacerse desde `Gnome` pasando el foco a esa ventana y pulsando `Alt-Impr Pant`.

Una vez que se ha hecho todo esto, se hace un tarball que contenga el nombre del directorio (de manera que se pueda desempaquetar en el directorio `/usr/share/gdm/themes`).

Este será el tarball que distribuirá y que la gente puede instalarse con el programa de configuración gráfico (gdmconfig). Puede hacerse esto con los comandos:

```
cd /usr/share/gdm/themes<br />tar czvf <theme_name>.tar.gz <theme_name><br />
```

¿Cómo hacer un tema de GDM?

Siempre es más fácil comenzar de un tema existente, en lugar de hacerlo de cero. En general, el primer tema que se va a cambiar es la imagen de fondo. La imagen de fondo debe ser al menos tan grande como la resolución de la pantalla. A fin de que el tema sea adecuado a otros usuarios, es una buena idea usar al menos 1600x1200 píxeles de imágenes, éstas se ven como fuertes incluso en las pantallas de alta resolución. Una pequeña imagen también se puede usar, pero se pierde calidad cuando es ampliado. La imagen debe ser colocada en el tema del directorio y declarada en el fichero xml [Ver Anexo 8].

A continuación, se tendrán que reposicionar los menús, iconos y las etiquetas con el fin de que coincida con su imagen de fondo. Aunque se puede hacer todo transparente, un cuadro de inicio de sesión centrada no siempre es la mejor opción. Se pueden definir imágenes, rectángulos y etiquetas con etiquetas. [Ver anexo 9].

Usando la etiqueta **Item**, también se puede especificar la ubicación de los botones, mensajes del sistema, la pantalla de login, etc añadiendo ID=tema de la etiqueta. Todo el sistema de etiquetas GDM. [Ver Anexo 10].

Se usa la etiqueta **pos** dentro de cada uno de los **items** para definir su sección X, y la posición y asegurarse de que todo parece en armonía con su imagen de fondo. Por último, el xml debe ser declarado en la GdmGreeterTheme.desktop del archivo de la siguiente manera:

```
Greeter=theme.xml
```

¿Cómo se activa el tema diseñado?

Una vez que se realiza la obra maestra, necesitará una captura de pantalla. Si se tiene **xnest** instalado, puede utilizarse `gdmthemetester` y se puede probar con los GDM que vienen por defecto en `cd /usr/share/gdm/themes`. Luego la siguiente línea de comando:

```
$ gdmthemetester xdmcp tema_a_alegir
```

Ahora solo se captura la ventana y se hace una referencia en `GdmGreeterTheme.desktop` añadiendo también el autor, el nombre y la descripción [*Ver Anexo 11*].

Por último, solo queda hacer un archivo de compresión con la extensión `tar`, esto se puede hacer con la siguiente línea de código:

```
$ gzip tar -zcf mi_tema.tar.gz
```

Puede ser fácilmente instalado en cualquier computador desde la pantalla de configuración de GDM.

2.1.4 ScreenSplash.

ScreenSplash o pantalla de inicio viene a ser la pequeña ventana que se abre al iniciar el sistema y que va indicando la carga de diversas aplicaciones, (también se llama quitamiedos, para que no de la sensación de que se ha roto el equipo por si tarda, y mostrar el avance de la carga). O sea es la pantalla de transición que aparece después de haber ingresado correctamente el usuario y contraseña, y mientras se carga el escritorio. Generalmente son imágenes y no provee contenido, es como una presentación. Esta ventana o esta presentación se pueden cambiar con el fin de adaptar el sistema al gusto y no lo contrario.

¿Cómo hacer un screensplash?

Lo primero que se debe hacer es crear el fondo, junto con ese texto que es el nombre escogido y la barrita horizontal de abajo, que es donde van apareciendo los iconos mientras se carga el escritorio.

Estos son los pasos a seguir:

1. Se crea una nueva imagen de las dimensiones que se quiere puede ser 800x686px). Se selecciona un color de fondo, y uno más claro como color de frente. Se busca ahora la herramienta de degradado y se cambia la forma a radial. Ya se puede lanzar el degradado haciendo click en el centro de la imagen y llevando el otro extremo hasta uno de los lados.
2. Se crea una nueva capa que se le llama *rejilla*. Ahora se va a realizar un patrón nuevo para rellenar esta capa con la rejilla. Se crea una nueva imagen de 25x25px con la capa de fondo transparente. Con la herramienta de seleccionar rectángulos se selecciona una línea vertical justo en el centro y se arrastra el color negro encima. Ahora se hace una línea horizontal en el medio y también se rellena de negro.
3. Se hace click en *Seleccionar-> Todos* y se va a *Script-Fu -> Selección -> A Patrón*. Se pide un nombre para el patrón, se le llama *Rejilla*. Ahora ya se puede seleccionar de *Diálogos -> Patrones* y arrastrarlo sobre la capa *rejilla*.
4. Se aleja un poco la imagen con el *zoom out* (tecla [-]) y se selecciona la herramienta de cambiar la perspectiva. Se hace click sobre la capa *rejilla* y se mueve el vértice inferior derecho más hacia la derecha, y el inferior izquierdo más hacia la izquierda. Así se consigue el efecto de profundidad de la rejilla. Se puede jugar un poco con los vértices para conseguir el efecto que se quiera.
5. Una vez que se tiene la rejilla con efecto de perspectiva, toca ir a *Capa -> Máscara -> Añadir máscara de capa* y crear una máscara iniciada a Blanco (Opacidad Total). Se selecciona el color negro y el blanco como colores de frente y fondo, respectivamente. En la herramienta de degradado, se pone la forma a bilineal y se marca la pestaña de invertido, para que vaya de blanco a negro. Sólo se quiere que se vea una parte de la rejilla, por el centro-arriba de la imagen, así que se lanza el degradado haciendo click en el medio de la zona en la que se quiere que permanezca, y se le lleva hacia abajo con la tecla [Ctrl] pulsada para asegurar que queda completamente vertical. Si no gusta cómo queda, *Editar -> Deshacer* y se vuelve a intentar.
6. Cuando ya se tenga visible la parte de la rejilla que se quiere y el resto transparente, se va a *Capa -> Máscara -> Aplicar máscara de capa*. Se desenfoca un poco la capa (*Filtros -> Desenfocar -> Desenfocar Gaussiano (Blur)* de 4px) y se ajusta la

opacidad de la capa para que no se vea demasiado. Tiene que verse lo justo, sólo se quiere que la imagen tenga una perspectiva de profundidad y para eso sirve la rejilla.

Ya se tiene el *fondo* y la *rejilla*. Ahora se necesita el texto que fue escogido, para ello cumpla los siguientes pasos:

1. Seleccione la herramienta de textos, con color gris oscuro y la fuente que se desea. Se escribe el texto deseado y se mueve a donde se quiera. Se selecciona la capa de *fondo* y se va a *Filtros* -> *Mapa* -> *Mapa de relieve (Bump Map)*. En la ventana que aparece, en la opción de *Mapa de relieve* se selecciona la capa con el texto. Con los parámetros de *profundidad* y *elevación* se juega para obtener un relieve más o menos vistoso del texto en la capa de *fondo*. Una vez que se acepta, ya se puede borrar la capa con el texto si se quiere.

Sólo queda la banda rectangular de abajo, para lo que es necesario:

1. Crear una nueva capa que se le llama *banda* y se selecciona el rectángulo que se quiere que sea la banda con la herramienta de selecciones rectangulares. Una vez que se tenga, se coge la herramienta de degradados, con forma bilineal y de blanco a transparente (pero con la pestaña de invertido activada). Se lanza un degradado horizontal desde el centro hacia un lado, con la tecla [Ctrl] pulsada.
2. Con la capa *banda* seleccionada, se va a *Seleccionar* -> *Ninguno* y *Filtros* -> *Desenfoque* -> *Desenfoque gaussiano (blur)* de 2px.

Se tiene el fondo del splash con el nombre escogido. Es importante guardar la imagen en formato .xcf para poder abrirla en otro momento con el Gimp y conservar todas las capas.

Ahora al resto, el **logo**, al que se va a aplicar un efecto *glossy* con reflejos para darle volumen (efecto gominola) y una estrella-chapa típica con el texto *New*. Se desglosa por pasos:

1. Se carga la imagen que se tiene guardada (en formato XCF) después de crear el fondo de la imagen en la explicación anterior. Se tiene las capas *fondo* (con el texto como mapa de relieve), la capa *rejilla* y la capa *bandas*. Se crea una nueva capa a la que se llama *reflejo_fondo*. Se va a crear un efecto *glossy* al fondo. Se va a

- Seleccionar* -> *Todos* y después a *Seleccionar* -> *Encoger* y se encoge la selección en 5 pixels. Se quiere que la selección sea una media luna, y no un rectángulo. Cuando se tenga una selección hecha, si se quiere quitar algo de dicha selección se tiene que seleccionar con la tecla [Ctrl] pulsada. Con la tecla [Shift] se suma lo que se seleccionó a la selección hecha, y con [Ctrl] se le quita. Así que se le quita un fragmento circular, con la herramienta de selección de elipses.
- Ya se tiene la elipse quitada, sólo se tiene que lanzar un degradado lineal de blanco a transparente, inclinado hacia la derecha y de arriba a abajo. Después, se va a *Filtros* -> *Desenfocar* -> *Desenfocar Gaussiano (Blur)* y se le aplica con un radio de 8 pixels. Se ajusta la transparencia de la capa. Si hace falta se puede borrar partes de la capa con la herramienta goma de borrar y una brocha grande y muy blanda.
 - Se crea una capa a la que se llama *logo_nombre deseado*. Se descarga el logo en formato PNG, se abre con el Gimp, se selecciona la palabra del logo con la herramienta de selecciones rectangulares, se copia y se pega en la capa de la imagen. Se coloca la palabra en el sitio correcto y se va a *Capa* -> *Anclar la capa*. De nuevo en la imagen descargada, ahora se selecciona el logo, se copia y se pega en la imagen, anclando finalmente la capa.
 - Ahora se aplica el efecto glossy a la palabra elegida. Se duplica la capa para trabajar en una copia. Con la herramienta de recoger colores de la imagen, se pincha en una de las zonas rojas del logo, con la herramienta de seleccionar zonas contiguas se selecciona, se pincha en la primera letra de la palabra y se arrastra el color rojo que se ha recogido. Así, se rellena el resto de las letras de la palabra con los mismos colores que el logo.
 - Con la herramienta seleccionar regiones contiguas, se pincha en medio de la primera letra de la palabra. Luego se va a *Seleccionar* -> *Encoger* y se encoge la selección en 4px. Se crea una nueva capa llamada *glossy_primerletradelapalabra*. Con la herramienta de degradados, se lanza uno de blanco a transparente de arriba a abajo sobre la letra. Después de este paso, en los pantallazos el logo aparece con el degradado de derecha a izquierda, en vez de izquierda a derecha, pero es lo mismo.
 - Teniendo activa la capa *glossy_primerletradelapalabra*, se desenfoca con *Filtros* -> *Desenfocar* -> *Desenfocar gaussiano (blur)* de 5px y después se ajusta la transparencia de la capa.

7. Se repite este paso con todas las letras del logo, de modo que se tendrá la capa con el logo y la palabra y encima de ella 6 capas con los degradados desenfocados correspondientes a cada letra. Recordar ajustar la transparencia de cada una de esas seis capas para mejorar el efecto.
8. Ya se tiene la palabra en modo glossy. Se le pone una sombra blanca, duplicando la capa en la que está la palabra que se ha pintado con los colores del logo. Esta capa se renombra a *sombra* y se le pone debajo de la otra. Se va a *Capa -> Colores -> Niveles* y se mueve la pestañita de la derecha (blancos) totalmente a la izquierda. Así se consigue que toda la palabra esté en blanco. Ahora se le aplica un *Filtro -> Desenfoque -> Desenfoque gaussiano (blur)* de 12px y se ajusta la transparencia de la capa. Ya se tiene la sombra.
9. Con el mismo procedimiento, ahora en vez de letras se tiene que jugar con el propio logo circular. En cada una de las 3 zonas y de los 3 círculos habrá que repetir el paso número 5, quedando 6 capas más con los degradados desenfocados de los elementos del logo circular. Luego se le crea una sombra blanca como se ha descrito en el paso 8.
10. Se abre el programa de diseño vectorial Inkscape (SWL y gratuito). Se crea un documento nuevo y con la herramienta de crear estrellas y polígonos se dibuja una estrella.
11. Arriba hay una barra donde se puede modificar el número de esquinas, el radio... y algunos parámetros más de la estrella. Se pone número de esquinas a 22, longitud del radio a 0,800 y los otros valores se le dejan en 0.
12. Haciendo click derecho sobre la estrella y en relleno y borde, podemos elegir con la rueda un color sólido como relleno. El naranja es muy común para estas estrellas. Una vez que la tengamos pintada, vamos a *Archivo -> Exportar mapa de bits* y le ponemos un nombre con extensión PNG. Antes de exportarla, en esa misma ventana le decimos a qué tamaño la queremos exportar, cambiamos el ancho a unos 200px. Le damos a exportar y ya podemos cerrar el Inkscape.
13. Con el Gimp, se abre esa imagen que se acaba de exportar. Como se ha guardado en formato PNG, se guarda la estrella sobre una capa transparente. Se crea una nueva capa llamada *estrella* en la imagen splash. Se va a la imagen de la estrella, se selecciona todo, se copia y se pega en la imagen del logo. Se ancla la capa.

14. Se coloca la estrella en la esquina derecha superior. Si hace falta se va a *Capa* -> *Escalar la capa* para reducirle un poco el tamaño de la estrella. Se *duplica la capa*, se pone debajo de la anterior, se va a *Capa* -> *Colores* -> *Niveles* y se lleva la pestaña de la izquierda (negros) hasta la derecha del todo. Así se tiene la estrella en negro. Se le aplica un *Filtros* -> *Desenfoque* -> *Desenfoque gaussiano blur* de 10px para conseguir la sombra de la estrella. Con la herramienta de textos, con el color y una tipografía adecuados, se escribe "New". Se pincha en la herramienta de rotar la capa o la selección, se hace click en la capa de texto y así se rota un poquito.

De esta manera se habrá creado el tema de ScreenSplash y solo nos queda activarlo.

¿Cómo activar el tema de ScreenSplash diseñado?

Para activar el tema de splash del inicio de gnome es necesario cambiar la variable "splash_image" del gconf-editor de la siguiente manera:

```
$ gconf-editor  
/ -> Apps -> gnome-session -> options  
/home/TU_USUARIO/.splash/mi_imagen_splash.png
```

Otra vía para cambiar la grafica de bienvenida de Gnome es la siguiente:

1. Activa los repositorios universe y multiverse.
2. Ir al menú de Ubuntu, Sistema, Administración y abrir Gestor de paquetes Synaptic, busca por gnome-splashscreen-manager.
3. Una vez instalado vuelva al menú principal de Ubuntu, Sistema, Preferencias y tendrá una nueva opción Splash Screen.
4. Ahora desde esa nueva opción, al hacer sobre ella doble click se le abrirá una nueva ventana. Para agregar una imagen que desee usar click en el botón Install, para eliminarla Delete y para que sea la nueva imagen splash click en Active.

2.1.5 Escritorio.

El escritorio, es la pantalla que se ve después de iniciar la sesión, es una de las características más importantes del equipo, puede contener accesos directos a los

programas, documentos, carpetas e incluso impresoras o dispositivos externos que se utilicen con más frecuencia, también puede haber contenido *activo*, como un canal de información bursátil o de noticias.

El escritorio de Microsoft Windows es una entidad indivisible. Es el núcleo del SO, aunque la consola **Disk Operating System (DOS)** continua existiendo en él, pero es una aplicación que se ejecuta sobre el escritorio.

El SWL ofrece el modelo opuesto, el núcleo del sistema es la shell. El escritorio es una aplicación grafica que se ejecuta sobre la shell. La aplicación escritorio en SWL se conoce como el Gestor de Ventanas o **Windows Manager (WM)**.

Windows Manager.

El WM es un programa que controla la ubicación y apariencia de las aplicaciones bajo el sistema X Windows. Los ordenadores modernos suelen ofrecer una interfaz gráfica de usuario que facilita la interacción con el sistema operativo.

Las plataformas Windows y Macintosh ofrecen métodos de visualización y control de las ventanas e interacción con las aplicaciones, estandarizados por sus vendedores. En cambio el sistema gráfico X Window, popular en el ámbito de sistemas Unix y similares, como GNU/Linux, permite al usuario escoger entre varios gestores según sus gustos o necesidades. Los gestores de ventanas difieren entre sí de muchas maneras, incluyendo apariencia, consumo de memoria, opciones de personalización, escritorios múltiples o virtuales y similitud con ciertos entornos de escritorio ya existentes.

Los gestores de ventanas para SWL son:

- AfterStep
- Blackbox
- FVWM
- FVWM95
- [Fluxbox
- IceWM

- JWM
- KWin
- Metacity
- Openbox
- Window Maker
- XPde
- Xfwm

Dentro de estos gestores de ventanas, Metacity es el gestor de ventanas por defecto para el entorno de escritorio Gnome, es por ello que se utilizará en esta investigación. El desarrollo de Metacity empezó por parte de Havoc Pennington bajo la GNU GPL. Antes de la introducción de Metacity en Gnome 2.2, este usaba Enlightenment como gestor de ventanas y luego Sawfish. A pesar de que Metacity es parte del proyecto Gnome y está diseñado para integrarse en el escritorio, no requiere Gnome para ejecutarse, y Gnome puede ser usado con distintos gestores de ventanas disponibles que soportan la parte de la especificación **Manual de Convenciones para la Comunicación entre Clientes (ICCCM)** que Gnome necesita.

Metacity está implementado en GTK+, lo cual le permite usar *themes* (temas) y mezclarse con otras aplicaciones GTK+. Se centra en la simplicidad y la usabilidad, más que en las novedades o trucos publicitarios.

Metacity, a diferencia de los anteriores gestores de ventanas de Gnome, tiene pocas opciones de configuración y su introducción como la elección por defecto fue controvertida. Aquellos a favor de Metacity dicen que Gnome está dirigida a nuevos usuarios de ordenador que no necesitan la configurabilidad extrema de Sawfish o Enlightenment. Havoc Pennington escribió una redacción explicando como escribió Metacity y como simplificó el escritorio Gnome. Las críticas dicen que Metacity ha sacrificado la flexibilidad y el control a menudo asociados con aplicaciones Unix. Esta perceptible carencia de flexibilidad ha llevado al desarrollo de add-ons como Brightside. A pesar del estado incompleto de la documentación del desarrollo del theme de Metacity, muchos themes han sido escritos para Metacity.

¿Cómo crear un tema de Metacity?

Diseñar un tema de Metacity es una tarea relativamente simple, es basado en una estructura XML definida. Para empezar es necesario crear un directorio en ambos de los lugares siguientes:

```
$PREFIX/share/themes/[theme_name]/metacity-1 /
```

```
$CASA /. el themes/[theme_name]/metacity-1 /
```

Según el nombre del tema. Aunque Metacity descubrirá los temas en cualquier ubicación, es quizás sabio guardar el tema dentro del directorio home mientras se escribe y se pone a punto.

El próximo paso para crear un tema de Metacity, es revisar el archivo que se encuentra dentro de ese directorio llamado 'metacity-tema-1.xml'. Este es el archivo que contiene la descripción de XML para el tema que tiene la descripción de **Document Type Definition (DTD)** siguiente. Al crear un nuevo tema de Metacity, es útil tomar un tema existente y copiar la descripción del xml para entonces comenzar a modificarlo, en lugar de crear un nuevo tema completamente desde el principio.

Antes de entrar en una explicación detallada de cada parte del documento, es necesario conocer de manera global los pasos y las variables que se soportan en él:

Tipos de la ventana.

Hay 6 ventanas específicas que Metacity reconoce, estas son:

<i>normal</i>	Una ventana sobre-novelada normal
<i>dialogo</i>	Una ventana de diálogo
<i>modal_dialog</i>	Una ventana del diálogo que tiene el modal estatal, intervención requerida por el usuario antes de que ellos puedan actuar recíprocamente con él, es la ventana del padre.
<i>menu</i>	Un menú de ventana habilitado que posee la opción de rasgue fuera.
<i>utility</i>	Una utilidad de pequeñas ventana, paletas o caja de herramientas
<i>border</i>	Una ventana que no debe mostrar el típicamente ninguna decoración. la ventana de la pantalla llena.

Styles.

Para cada tipo de la ventana, es necesario trazar un estilo (o decoración de la ventana) hacia él. Cada uno de esos estilos es una composición de estilos para diferentes estados del marco. Es preciso especificar cómo cada uno de estos estados del marco debe aparecer en su estilo. Los estados del marco son determinados por lo siguiente:

- Si la ventana se enfoca o desenfoca.
- Si la ventana se aumenta al máximo u obscurece (o ambos).
- Si la ventana puede ser verticalmente, horizontalmente, verticalmente y horizontalmente el cambio de tamaño.

Cada uno de los estados del marco se traza a un estilo del marco. Un estilo del marco es dividido en dos partes diferentes – en el marco los pedazos y en la ventana los botones. Esta separación hace más fácil el construir un tema de Metacity.

Los pedazos de marco, son imprescindible. Si se omite cualquiera de los estos, entonces no se dibujará nada para él. Metacity reconoce los pedazos del marco siguientes:

<i>entire_background</i>	El marco entero de la ventana que es primero arrastrado.
<i>titlebar</i>	El área sobre la ventana de la aplicación.
<i>titlebar_middle</i>	El área de la barra de títulos sin considerar 'el borde'.
<i>left_titlebar_edge</i>	El área al lado izquierdo de la barra de título.
<i>right_titlebar_edge</i>	El área al lado derecho de la barra de título.
<i>top_titlebar_edge</i>	El área a la encima de la barra de título.
<i>bottom_titlebar_edge</i>	El área al fondo del de la barra de título.
<i>title</i>	Área que contiene el título.
<i>left_edge</i>	El borde izquierdo del marco.
<i>right_edge</i>	El borde derecho del marco.
<i>bottom_edge</i>	El borde del fondo del marco.
<i>overlay</i>	Igual que el <i>entire_background</i> , sólo que es arrastrando el último.

Además se reconocen los botones de ventanas:

<i>close</i>	Botón de cerrar ventana.
<i>maximize</i>	Botón de maximizar ventana.
<i>minimize</i>	Botón de minimizar ventana.
<i>menu</i>	Botón de menú.

Y las siguientes posiciones de los botones:

<i>left_left_background</i>	Especifica el background del primer botón a la izquierda.
<i>left_middle_background</i>	Especifica el background del segundo botón a la izquierda.
<i>left_right_background</i>	Especifica el background del tercer botón a la izquierda.
<i>right_left_background</i>	Especifica el background del primer botón a la derecha.
<i>right_middle_background</i>	Especifica el background del segundo botón a la derecha.
<i>right_right_background</i>	Especifica el background del tercer botón a la derecha.

Para cada botón que aparece en nuestra ventana, Metacity reconoce los siguientes estados:

<i>normal</i>	Cuando el botón aparece normal, sin otros estados.
<i>pressed</i>	Cuando el botón aparece clikeado por el ratón.
<i>prelight</i>	Cuando el botón aparece debajo del ratón sin ser clikeado.

Luego de conocer cuáles son los valores que pueden tomar las variables dentro del documento XML que contiene los temas de metacity, solo resta tomar un tema y comenzar a modificar dichos valores hasta lograr el efecto deseado.

¿Cómo activar el tema creado?

Para activar el tema de metacity es necesario que se pruebe previamente. Así se podrán eliminar los errores que puedan haber surgido en su creación. Esto se puede hacer con la siguiente línea de código:

```
gconftool -2 - -type = string - -set / apps/ metacity/ general/ theme[nombre del tema]
```

Además existen aplicaciones para comprobar dichos temas, ejemplo: el software *metacity-theme-viewer*. El uso de esta aplicación es simple, y evalúa cuantos temas sean necesarios evaluar. Si se produce un error en el tema creado, la aplicación muestra la línea donde se encuentra el error, y solamente es necesario corregirlo y volver a comprobar con dicho software.

Paneles.

Además de configurar las formas de las ventanas, se hace necesario gestionar la configuración de los paneles. Los paneles son las aplicaciones más importantes del escritorio, pues ellas proveen las funcionalidades que lo hacen ser lo que es desde el punto de vista conceptual al ofrecer un entorno para la ejecución de programas de manera interactiva y estéticamente diferente, que permite entre otras cosas:

1. Mostrar las aplicaciones disponibles para su uso por los usuarios.
2. Mostrar una lista de las aplicaciones en ejecución.
3. Que las aplicaciones activas puedan ser visualizadas en varios espacios de trabajo (workspace).

Para lograr este objetivo, los mismos están compuestos a su vez por items o elementos divididos en las siguientes categorías:

- Menú de aplicaciones.
- Lanzadores de aplicaciones (aplicaciones externas y applets: aplicaciones incrustadas).
- Lista de aplicaciones en ejecución.

Es importante saber la función de cada una de estas partes:

Menú de aplicaciones.

El menú de aplicaciones es una lista que contiene programas de aplicación instalados en nuestra PC desglosados en “categorías” según su función. Esta lista puede no abarcar todos los programas disponibles, esto depende de los gustos particulares de los usuarios.

Lanzadores de aplicaciones.

Un lanzador de aplicación es un elemento que permite la ejecución de un programa o contiene una lista de sub-items, que proveen esta funcionalidad. Es por tanto en cierta medida similar a un menú de aplicaciones, aún cuando, como este, no puede subdividirse en categoría alguna, solo permitir ejecutar aplicaciones.

A tales efectos existen dos clasificaciones fundamentales de lanzadores de aplicaciones, estas son:

1. Lanzador de aplicación independiente del panel.
2. Lanzador de aplicación dependiente del panel o applet.

Los *lanzadores de aplicaciones independientes del panel* son aquellos que permiten ejecutar aplicaciones que no dependen de este, y están contenidas por lo general en el menú de aplicaciones, siendo su función fundamental la de posibilitar el acceso rápido a los programas que el usuario necesita y que al hacerlo a través de dicho menú les tomaría más tiempo. Son por tanto lanzadores configurables, es decir que el usuario decide qué programas ejecutar a través de ellos. Ejemplos:

- Blender.
- OpenOffice.
- Console.
- Manejador de archivos.
- NetBeans.

Los *lanzadores de aplicaciones dependientes del panel* por su parte son lanzadores predefinidos, que por lo general posibilitan la ejecución de un programa no presente en el menú de aplicaciones, cuya función es general, es decir que sirve para cualquier tipo de usuario. Ejemplos de ellos se tienen los siguientes:

1. Selector de idioma.
2. Visor de tiempo.
3. Redactor de notas.
4. Menú de acciones.
5. Visor del escritorio personal.
6. Selector de espacio de trabajo (workspace).

Lista de aplicaciones en ejecución.

Como su nombre lo indica, la lista de aplicaciones en ejecución es un elemento que muestra las aplicaciones que un usuario ejecuta en todo momento, no las que puede ejecutar, sino las que se están ejecutando por él en un instante dado. A tales efectos existen varias maneras de que esta lista sea visible, la más común de todas es la referida a cuando se clickea Alt+Tab y se selecciona con right/left el programa que se quiere visualizar en primer plano. La otra se muestra en el panel, pero su selección se hace a través del mouse.

2.2 Herramientas de configuración existentes.

Como Tocororo-Desktop es un proyecto que concibe adecuar las interfaces gráficas de los sistemas GNU/Linux a un tipo de usuario en particular, se habla por tanto de la configuración de todas ellas como un todo, reunidas en una aplicación, además, las experiencias de algo similar como se ha relatado no son muchas, aunque en el mundo del SWL existen aplicaciones que gestionen la configuración de algunos de los elementos del sistema a los que el usuario tiene acceso normalmente.

Ejemplo de estas aplicaciones está el Administrador de Configuración de Gnome (gconf) que ofrece una idea general de todos los elementos del sistema al que normalmente tiene acceso cualquier usuario, y no a interfaces solamente, a lo que se une el hecho de que el estándar de diseño del escritorio de este entorno no sea el más ideal para los fines que se

pretenden, ya que es una herramienta complicada visualmente, por lo que los niños no podrán trabajar fácilmente con ella.

Otras aplicaciones como el Beryl y CompizFusion complementan las funcionalidades gráficas, las alternan además, con dinamismo, pero siguen sin poder cambiar el estándar que ya se ha creado y por el cual se rigen todos los desarrolladores, no es por tanto un modelo a seguir, aunque bien puede servir, como hasta ahora, de complemento.

Existe también, la herramienta Gnome Art Manager, la cual es un programa para la personalización de los escritorios de Gnome. Posee grandes funcionalidades que se pueden acceder desde el menú de esta aplicación, el que está a su vez estructurado por secciones como por ejemplo:

1. **Backgrounds:** (*Fondos de pantalla*): fondos de pantalla. Seleccionando All (*Todo*) se pueden obtener todos los fondos disponibles.
2. **Desktops Themes:** (*Temas de Escritorios*): el cual incluye Application, Windows Border e Icon.
3. **Applications:** (*Aplicaciones*): el cual configura los temas para las barras laterales, color de los paneles y color de las fuentes.
4. En **Windows Border** (*Borde de Ventanas*): encargado de los temas de las ventanas.
5. **Icons** (*Iconos*): temas de iconos.
6. **Other Themes** (*Otros Temas*) incluye Login Manager, Splash Screen, GTK+ Engines.
7. **Login Manager** (*Administrador de Inicio de Sesión*): imágenes para la pantalla de inicio de sesión
8. **Splash Screen** (*Pantalla de Splash*): imágenes para la pantalla de Splash (la que aparece después de iniciar sesión)
9. **GTK+ Engines** (*Motores GTK+*): temas para el motor GTK.

Pero esta herramienta posee la dificultad de que las imágenes Splash sólo se pueden descargar y luego instalarlas a mano, lo cual no es factible para la aplicación, ya que se necesitaría de conexión Internet a la hora de configurar, y además de que es una aplicación destinada a los niños y que por ende no tienen mucho conocimiento de informática.

2.3 Propuesta del sistema: Tocororo – Desktop

Estudios del tema demuestran que aunque varios son los esfuerzos que se han hecho en los Sistemas GNU/Linux relacionados con la personalización, su enfoque es general, destinado a una mayoría de usuarios y no a un tipo de usuario en particular; los esquemas que se proponen de diseño de las interfaces, pueden no ser los más óptimos por razones lógicas, sobre todo en personas no acostumbradas al mundo de la informática o que simplemente se inician en este sector. Así existen experiencias de diseño de escritorios en el que los elementos se parecen más a la vida real, las personas tienen por tanto una referencia natural para el trabajo con estos, haciéndoles más fácil su entendimiento y aprendizaje. Sin embargo son esfuerzos aislados y ninguno se enfoca específicamente a los niños, no tienen en cuenta por tanto sus criterios, carecen de su inocencia o la simplicidad con la que ven la complejidad de nuestro mundo. Y el objetivo es que las herramientas deben estar hechas para asemejarse a las personas, no para que las personas se parezcan a ellas.

Tocororo-Desktop es un sistema, que tiene como objetivos el desarrollo de un gestor para la configuración, de cara al usuario final, de cualquiera que fuere la distribución GNU/Linux que se emplee, para la educación primaria cubana. La idea del sistema, sería que el gestor de configuración de interfaces proveyera la configuración de manera automática, instalando además el software necesario para este tipo de enseñanza.

Los aportes tecnológicos concretos de este sistema serían varios como a continuación se relatan:

1. Gestor de configuración de interfaces independiente de los gestores de configuración conocidos.
2. Configuración distinta de las interfaces de usuario que incluye por el momento:
 - Diseño diferente del Ambiente de Trabajo:
 - ✓ Panel de aplicaciones metamorfoseable con funcionalidades nuevas.
 - ✓ Concepción distinta y más natural del escritorio.
 - ✓ Otras funcionalidades estéticas.
 - Nuevo Administrador de Sesiones (Display Manager) con tendencia a la multimedia.

3. Aplicaciones visuales para el diseño de las interfaces. En la actualidad no existen aplicaciones visuales que nos permitan diseñar cómodamente las interfaces de usuario conocidas.

Con este sistema se gestionan procesos básicos como son la configuración del GrubSplash, el BootSplash, el GDM, el ScreenSplash y el escritorio, el cual incluye los temas de metacity, los paneles y el background, todos estos elementos explicados anteriormente.

Para gestionar la configuración del GrubSplash y el ScreenSplash, no se hace necesario el cambio de estructura del greeter.dtd, porque los conceptos que se manejarán por este sistema no difieren de los que se tuvieron en cuenta en la gui/greeter/greeter.dtd. Es por ello que el sistema creará solo temas básicos sin modificar la estructura de los archivos xml, y la aplicación cargará dichos temas para que el usuario pueda elegir entre ellos y aplicarlo en el ordenador.

Sin embargo, en aspectos como el GDM, si se hace necesario modificar la estructura del greeter porque la idea de este sistema es crear una imagen de forma más interactiva, con nuevos conceptos en cuanto a la estructura normal. Estos conceptos son los que dan origen al **Tocororo Display Manager (TDM)**, el cual incluye en su implementación conceptos como la no aparición en sus inicios del prompt de logueo, o la interacción de las imágenes, para lograr explicaciones de los objetos que en ella se encuentran. En otras palabras, se incluyen tags o etiquetas que permitirán mostrar a los usuarios mensajes de cada uno de los objetos que conforman la imagen del GDM.

Además de este nuevo concepto, surge para la configuración de los paneles de Tocororo – Desktop el **Tocororo Panel (TocoPan)**. TocoPan es un estilo de panel un tanto diferente al panel clásico, pues a diferencia de este es circular; esta característica única podría limitarlo físicamente, al abarcar un poco más de espacio en la pantalla del monitor, no obstante a ello tiene la ventaja indiscutible de ser estéticamente más vistoso que los paneles lineales y sus posibilidades de efectos son más amplia.

Como todo panel, TocoPan ofrece las funcionalidades anteriormente, solo que su estructura interna difiere un tanto de la idea ofrecida por los paneles lineales. A tales efectos, TocoPan

cuenta con anillos para contener las diferentes categorías de lanzadores arriba descritas y un conjunto de otros elementos para realizar el resto de las funciones.

Menú de aplicaciones.

En un TocoPan existen dos maneras de visualizar el menú de aplicaciones, las más conocida de ellas se refiere a los menús flotantes, estilo similar al clásico cuyo comportamiento no difiere mucho de este, en tanto la segunda aparece incrustada dentro del panel con algunas ventajas que la hacen más efectiva que la primera.

A diferencia de los paneles lineales en los que pueden coexistir sobre estos varios (en realidad el mismo) menús de aplicaciones, en un TocoPan existe uno y solo un menú de aplicaciones para acceder a los programas, localizado por lo general en la parte inferior o superior del panel en ejecución por defecto. No obstante esta realidad puede ser cambiada en el TocoPan Editor donde se puede transformar o crear a gusto propio, no sin antes tener en cuenta ciertas restricciones de diseño a fin de que se cumplan las funcionalidades básicas de un panel.

Lanzadores de aplicación.

Los lanzadores de aplicaciones se corresponden en un TocoPan con dos anillos para describir funciones distintas según su localización:

- Anillo exterior: En él se localizan los lanzadores de aplicaciones independientes del panel.
- Anillo interior: Concentra en su estructura a los lanzadores de aplicaciones dependientes del panel.

Lista de aplicaciones en ejecución.

La lista de las aplicaciones en ejecución se muestran en el centro del panel de tres formas, o bien independiente de este en un panel lineal situado en la parte superior o inferior de la pantalla del monitor, de la manera clásica o en su estilo Compiz Fusion. Las tres primeras formas referidas al panel y las dos últimas son accesibles a través de la combinación de

teclas Alt+Tab, en tanto que la tercera una vez activa siempre es visible en los lugares citados (superior o inferior de la pantalla).

TocoPan integra estilos básicos y algunos que pertenecen a otras herramientas. Los estilos aplicables se muestran a continuación:

1. **Estilo Clásico:** El estilo clásico siempre está activo por defecto.
2. **Estilo CompizFusion:** Este estilo no aparece por defecto, hay que activarlo, y cuando se hace, el estilo clásico deja de funcionar. Sus efectos se relatan en sitios especializados en el tema del Compiz Fusion, de manera que no los trataremos en esta ocasión, aunque si resaltaremos la gran variedad de figuras 3D que incluye, como por ejemplo el efecto del cuadrado y del cilindro incluido en la última versión del producto.
3. **Estilo TocoPan:** De la misma manera que ocurría con el Compiz Fusion esta forma no aparece por defecto, hay igualmente que activarla y al hacerlo las dos anteriores dejan de funcionar. En este nuevo estilo, el procedimiento para enfocar otra aplicación en ejecución se realiza de la forma tradicional, es decir presionando simultáneamente Alt+Tab+Down Left/Right.

Además TocoPan incluye varias formas de mostrar las aplicaciones activas, estas formas serán explicadas a continuación:

1. Lineal.

La forma lineal se asemeja a las formas anteriormente citadas, la diferencia está en el lugar donde ocurre, pues si las dos primeras lo hacían sobre la pantalla ocupando presumiblemente mucho espacio de la misma, esta lo hace dentro del TocoPan, y solo dentro de este.

Características:

- Las aplicaciones aparecen una al lado de la otra.
- Solo se observa dentro del panel al presionar Alt+Tab la próxima aplicación.

- El movimiento para la selección de la aplicación a activar ocurre en la dirección derecha-izquierda.

2. Cascada.

Las cascadas recuerdan un tanto al viejo y obsoleto SO Windows, pero solo eso, nada más.

Características:

- Las aplicaciones aparecen en cascada
- Todas las aplicaciones son observables dentro del panel
- El movimiento para acceder a la próxima aplicación ocurre en la dirección fondo-frente

3. Rotacional

Esta es quizás la forma más llamativa de todas, pues al presionar Alt+Tab el panel rota sobre su eje mostrándonos la próxima aplicación.

Además se integrará la aplicación Gnus que es una aplicación visual desarrollada por el proyecto Tocororo-Desktop la cual permite el diseño de paneles teniendo como base cada una de las formas descritas anteriormente. Cuenta además con un conjunto de funciones para el modelo creado en dependencia de las características de este, al tiempo que propone un esquema para su diseño.

Capítulo 3: Planificación y Definición del Sistema.

3.1 Planificación y Definición.

En la etapa de la planificación del sistema lo que se pretende es establecer una visión del sistema, además de fijar expectativas y asegurar el financiamiento del software. En esta etapa, todo se centra sobre el qué, intentando definir que información ha de ser procesada, que función y rendimiento se desea, que comportamiento del sistema, que interfaces van a ser establecidas, que duración se necesita para definir un sistema correcto, en fin, han de identificarse las historias de usuarios claves del sistemas y del software. Cuenta con tres tareas principales, dentro de las que se encuentra el análisis de las historias de usuarios.

3.1.1 Lista de Reserva del Producto.

Luego de identificados los conceptos que implica el objeto de estudio, se hace necesario analizar qué es preciso hacer para que se cumplan todos los objetivos trazados en la investigación. Desde el inicio del desarrollo de sistemas existe un gran problema, la identificación de las historias de usuarios del sistema. La **Lista de Reserva del Producto** o **Product Backlog List (LRP)** define posibles elementos que pueden ser requerimientos técnicos y del negocio, funciones, errores a reparar, defectos, mejoras y actualizaciones tecnológicas requeridas. Esta es una lista priorizada que define el trabajo que se va a realizar en el proyecto. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto. En principios una LRP del sistema sería:

Prioridad	Descripción	Estimación	Estimado por:
Alta	Gestionar la configuración del GDM.	1	Semana
	Gestionar la configuración del BootSplash.	2	Semana
	Gestionar la configuración del GrubSplash	2	Semana
	Gestionar la configuración del ScreenSplash	2	Semana
	Gestionar la configuración del Escritorio.	2	Semana
	Gestionar la configuración del panel del escritorio.	1	Semana
	Gestionar la configuración de los temas de Metacity.	2	Semana
	Gestionar la configuración del fondo de escritorio.	1	Semana
	Cómo crear temas de cada una de las interfaces anteriores.	2	Semana
	Creación de los prototipos de interfaz para cada uno de los módulos del sistema.	2	Semana
Media	Programación de los prototipos de interfaz para cada uno de los módulos.	3	Semana
	Creación de temas para cargar en la aplicación.	8	Días

De un estudio de las actividades reflejadas en la LRP, se deducen las siguientes historias de usuarios:

Historias de Usuarios:

Las historias de usuarios son servicios que el sistema debe ser capaz de brindar. En él se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que el sistema debe realizar, y las condiciones extremas a determinar por el sistema. En el sistema que se propone se tienen los siguientes:

➤ Gestionar la configuración del Sistema:

Incluye la gestión de la configuración de todos los elementos que se integrarán en el sistema, además de la compatibilidad entre ellos.

➤ Configurar la interfaz del escritorio

Permite al usuario configurar la interfaz del escritorio de manera general, e incluso diferenciar entre las diferentes partes que lo componen, como por ejemplo:

✓ Configuración del panel.

Este permite la configuración del panel, incluyendo el cambio de formas de este y de los elementos y menús que lo componen, para lo que se crea un nuevo concepto que se llamará TocoPan, y una nueva interfaz con el nombre de Gnus.

✓ Configuración del tema de iconos.

Se refiere a la configuración de los temas de iconos, el cambio de sus formas y colores.

✓ Configuración del fondo de escritorio.

Se enmarca en la configuración del background, es decir, la imagen de fondo del escritorio.

➤ Configurar la interfaz del GDM.

Posibilita la configuración de los temas de GDM, en este caso con imágenes interactivas que aunque no difíciles de implementar, crean un nuevo concepto del archivo XML, a este nuevo concepto se le conoce como TDM (Tocororo – Desktop Display Manager). Para configurar esta interfaz, se hace necesario que el usuario tenga permisos de administrador del sistema.

➤ Configurar la interfaz del Bootsplash.

Brinda la facilidad para cambiar la interfaz del boot splash, por cualquiera que el usuario elija, para lo que se hace necesario, al igual que en el TDM, los permisos de administrador del sistema.

➤ Configurar la interfaz del GrubSplash.

Facilita la configuración del Grub permitiéndole al usuario la inserción de imágenes si así lo desea o al cambio de muchos elementos de esta interfaz.

➤ Configurar la interfaz del ScreenSplash.

Establece la gestión de la configuración del ScreenSplash.

➤ Apariencia e interfaz:

1. El sistema debe presentarse en una interfaz donde el usuario mantenga alto control de la información que necesita y de su forma de presentación.
2. La interfaz debe ser simple e intuitiva. En su diseño debe tenerse en cuenta el orden visual y reflejarse la jerarquía de la información.
3. Debe mantenerse la compatibilidad entre los diferentes módulos que integran el software, además de las resoluciones de pantalla.

➤ Usabilidad:

1. La interfaz del sistema deberá ser diseñada para lograr un mayor y mejor uso e interacción con los usuarios, deben diseñarse basados en el enfoque de aproximación al usuario, tratando de satisfacer sus necesidades y sirviéndoles de herramienta para realizar sus funciones.
2. Estos servicios deben tributar a que los usuarios consigan sus objetivos específicos con efectividad, eficiencia y satisfacción en el contexto donde estos se definan.

➤ Rendimiento:

1. El sistema debe ser capaz de procesar ágilmente los nuevos cambios que solicite el usuario, para no provocar lentitudes a la hora de la instalación.

➤ Seguridad:

1. En algunos de los casos el usuario necesita permisos de root para realizar algunos cambios, es necesario que el sistema provea seguridad ante esta información e identifique al usuario correctamente.
2. Garantizar que los cambios sean gestionados únicamente por quien tiene derecho.
3. Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que este activo.

➤ Soporte:

1. El sistema será instalado sobre SWL.
2. El sistema debe funcionar bajo el entorno de escritorio Gnome.
3. El sistema funcionará sobre cualquier distribución que se instale.

➤ Confiabilidad:

1. En la parte de la configuración del GRUB y el BOOT solo lo podrán hacer los usuarios con permisos de admón.
2. La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.

- Legalidad:
 1. La plataforma escogida para el desarrollo de la aplicación, está basada en la licencia GNU/GPL.
- Hardware:
 1. Mínimo Requerido: RAM 128 Mb, Hard Disk 512 Mb.
 2. Mínimo Recomendado: RAM 256 Mb, Hard Disk 1 Gb.
- Diseño e implementación:
 1. El sistema funcionará sobre una aplicación desarrollada utilizando SWL.
 2. Se desarrollará sobre un sistema GNU/Linux, utilizándose herramientas como el Gimp, Blender, Inskape y Visual Paradigm.
- Políticos – culturales:
 1. Como es un producto destinado para Cuba, el sistema debe ajustarse a las condiciones político-culturales que este país requiere.
- Funcionalidad:
 1. Reducir al mínimo el tiempo de configuración.
 2. Que la configuración sea lo mas eficiente posible.

3.1.2 Modelo de historias de usuarios del dominio.

Un modelo de historias de usuario de dominio es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales de dominio o del problema.

Representa conceptos del mundo real, no de los componentes de software. Representa la estructura y dinámica de la organización, en este caso se representarán los principales conceptos y los elementos más importantes del contexto del sistema. Este modelo se representa a través de un diagrama UML.

Algunas veces en dominios de negocio pequeños, no es necesario realizar un modelo de objetos para el dominio, puede ser suficiente con un glosario de términos. Por la relativa simplicidad del entorno donde se desarrolla el sistema, no es necesario profundizar a través de un modelo de negocio, basta con el modelo de dominio para capturar los principales conceptos alrededor del problema que la aplicación resuelve.

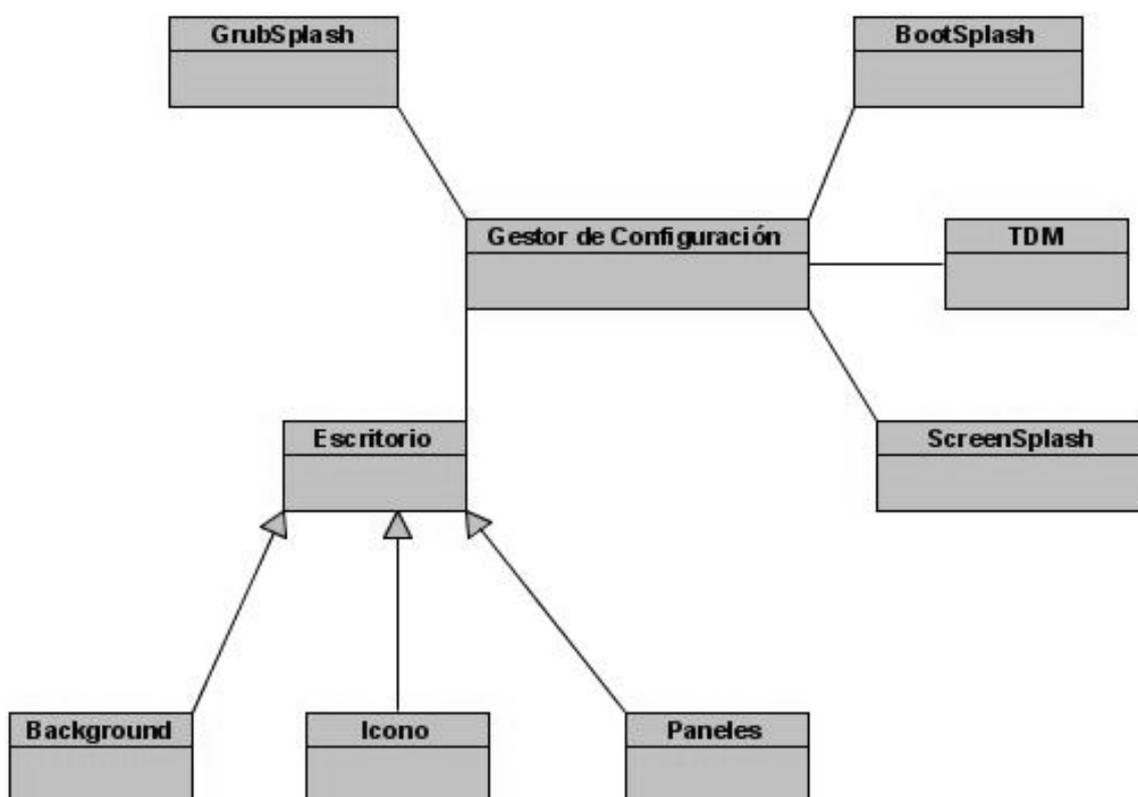


Imagen 1 Modelo de historias de usuario del dominio.

3.1.3 Definición de las principales Historias de Usuarios.

Utilizando las facilidades que brindan las metodologías ágiles, se representarán los requisitos del sistema mediante historias de usuarios, que recogen datos importantes como la descripción de la acción que se ejecutará por el actor del sistema. A continuación se

presentan las historias de usuarios, y los actores que interactúan con ellas, determinados para satisfacer los requerimientos funcionales del sistema:

Historia de Usuario	
Número: U-TD-01	Nombre Historia de Usuario: Gestionar la configuración del GrubSplash.
Modificación de Historia de Usuario Número:	
Usuario: Usuario del Sistema	Iteración Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Alto	Puntos Reales:
Descripción: Comienza cuando el usuario del Sistema solicita la configuración del GrubSplash, el sistema muestra una interfaz para elegir un tema de los predeterminados, el usuario selecciona uno de los temas, y entonces el sistema aplica la configuración elegida por el usuario al GrubSplash.	
Observaciones: No se necesita permiso de administración del sistema.	

Historia de Usuario	
Número: U-TD-02	Nombre Historia de Usuario: Gestionar la configuración del BootSplash.
Modificación de Historia de Usuario Número:	
Usuario: Usuario del Sistema	Iteración Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales:
Descripción: Comienza cuando el usuario del Sistema solicita la configuración del BootSplash, el sistema muestra una interfaz para elegir un tema de los predeterminados, el usuario selecciona uno de los temas, y entonces el sistema aplica la configuración elegida por el usuario al BootSplash.	
Observaciones: Se necesita permiso de administración del sistema.	

Historia de Usuario	
Número: U-TD-03	Nombre Historia de Usuario: Gestionar la configuración del TDM.
Modificación de Historia de Usuario Número:	
Usuario: Usuario del Sistema.	Iteración Asignada: 4
Prioridad en Negocio: Alta.	Puntos Estimados: 1
Riesgo en Desarrollo: Alto.	Puntos Reales:
Descripción: Comienza cuando el usuario del Sistema solicita la configuración del TDM, el sistema muestra una interfaz para elegir un tema de los predeterminados, el usuario selecciona uno de los temas, y entonces el sistema aplica la configuración elegida por el usuario al TDM.	
Observaciones: Se necesita permiso de administración del sistema.	

Historia de Usuario	
Número: U-TD-04	Nombre Historia de Usuario: Gestionar la configuración del ScreenSplash.
Modificación de Historia de Usuario Número:	
Usuario: Usuario del Sistema.	Iteración Asignada: 3
Prioridad en Negocio: Alta.	Puntos Estimados: 2
Riesgo en Desarrollo: Alto.	Puntos Reales:
Descripción: Comienza cuando el usuario del Sistema solicita la configuración del ScreenSplash, el sistema muestra una interfaz para elegir un tema de los predeterminados, el usuario selecciona uno de los temas, y entonces el sistema aplica la configuración elegida por el usuario al ScreenSplash.	
Observaciones: No se necesita permiso de administración del sistema.	

Historia de Usuario	
Número: U-TD-05	Nombre Historia de Usuario: Gestionar la configuración del Escritorio.
Modificación de Historia de Usuario Número:	
Usuario: Usuario del Sistema.	Iteración Asignada: 4
Prioridad en Negocio: Alta.	Puntos Estimados: 1
Riesgo en Desarrollo: Alto.	Puntos Reales:
Descripción: Comienza cuando el usuario del Sistema solicita la configuración del escritorio, el sistema muestra una interfaz para elegir un tema de los predeterminados, el usuario selecciona uno de los temas, y entonces el sistema aplica la configuración elegida por el usuario al escritorio.	
Observaciones: No se necesita permiso de administración del sistema.	

Historia de Usuario	
Número: U-TD-06	Nombre Historia de Usuario: Gestionar la configuración del panel.
Modificación de Historia de Usuario Número: U-TD-05	
Usuario: Usuario del Sistema.	Iteración Asignada: 4
Prioridad en Negocio: Alta.	Puntos Estimados: 1
Riesgo en Desarrollo: Alto.	Puntos Reales:
Descripción: Comienza cuando el usuario del Sistema solicita la configuración del escritorio, y dentro de este del panel, entonces el sistema muestra una interfaz para elegir varios aspectos como la forma del mismo, además del estilo de las listas de aplicaciones activas y otras, el usuario modifica las opciones deseadas y el sistema aplica la configuración elegida por el usuario al panel de escritorio.	
Observaciones: No se necesita permiso de administración del sistema.	

Historia de Usuario	
Número: U-TD-07	Nombre Historia de Usuario: Gestionar la configuración de los temas de ícono.
Modificación de Historia de Usuario Número: U-TD-05	
Usuario: Usuario del Sistema.	Iteración Asignada: 4
Prioridad en Negocio: Alta.	Puntos Estimados: 2
Riesgo en Desarrollo: Alto.	Puntos Reales:
Descripción: Comienza cuando el usuario del Sistema solicita la configuración del escritorio, y dentro de este de los temas de íconos, entonces el sistema muestra una interfaz para elegir varios aspectos como la forma del mismo, además del estilo de los íconos y otros, el usuario modifica las opciones deseadas y el sistema aplica la configuración elegida por el usuario al tema de ícono del escritorio.	
Observaciones: No se necesita permiso de administración del sistema.	

Historia de Usuario	
Número: U-TD-08	Nombre Historia de Usuario: Gestionar la configuración del fondo de escritorio.
Modificación de Historia de Usuario Número: U-TD-05	
Usuario: Usuario del Sistema.	Iteración Asignada: 4
Prioridad en Negocio: Alta.	Puntos Estimados: 1
Riesgo en Desarrollo: Alto.	Puntos Reales:
Descripción: Comienza cuando el usuario del Sistema solicita la configuración del escritorio, y dentro de este del fondo de escritorio, entonces el sistema muestra una interfaz para elegir varios aspectos como las imágenes, además de la posición y otros, el usuario modifica las opciones deseadas y el sistema aplica la configuración elegida por el usuario al fondo del escritorio.	
Observaciones: No se necesita permiso de administración del sistema.	

3.1.4 Tareas de la ingeniería.

En este epígrafe se muestran las principales tareas que dan solución a cada una de las Historias de Usuarios definidas anteriormente.

Tarea	
Número tarea: 1	Número historia: U-TD-01
Nombre tarea: Estudio del fichero xml que define la estructura del GrubSplash.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 10 octubre 2007	Fecha fin: 24 octubre 2007
Programador responsable: Equipo XP	
Descripción: Estudio de los diferentes howto para crear o personalizar el GrubSplash.	

Tarea	
Número tarea: 2	Número historia: U-TD-01
Nombre tarea: Creación del prototipo de interfaz para gestionar la configuración del GrubSplash.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 10 abril 2008	Fecha fin: 24 abril 2008
Programador responsable: Equipo XP	
Descripción: Creación del prototipo de interfaz que permitirá la gestión de configuración del GrubSplash, para comprobar que su diseño es acorde con las normas de gui/greeter.dtd a la hora de su implementación.	

Tarea	
Número tarea: 3	Número historia: U-TD-02
Nombre tarea: Estudio del fichero xml que define la estructura del BootSplash.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 24 octubre 2007	Fecha fin: 31 octubre 2007
Programador responsable: Equipo XP	
Descripción: Estudio de los diferentes howto para crear o personalizar el BootSplash.	

Capítulo 3: Planificación y Definición del Sistema.

Tarea	
Número tarea: 4	Número historia: U-TD-02
Nombre tarea: Creación del prototipo de interfaz para gestionar la configuración del BootSplash.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 24 abril 2008	Fecha fin: 3 mayo 2008
Programador responsable: Equipo XP	
Descripción: Creación del prototipo de interfaz que permitirá la gestión de configuración del BootSplash, para comprobar que su diseño es acorde con las normas de gui/greeter.dtd a la hora de su implementación.	

Tarea	
Número tarea: 5	Número historia: U-TD-03
Nombre tarea: Estudio del fichero xml que define la estructura del TDM.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 31 octubre 2007	Fecha fin: 7 noviembre 2007
Programador responsable: Equipo XP	
Descripción: Estudio de los diferentes howto para crear o personalizar el GDM, el cual servirá de base para la creación de un howto para la configuración del TDM.	

Tarea	
Número tarea: 6	Número historia: U-TD-03
Nombre tarea: Creación del prototipo de interfaz para gestionar la configuración del TDM.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 5 mayo 2008	Fecha fin: 12 mayo 2008
Programador responsable: Equipo XP	
Descripción: Creación del prototipo de interfaz que permitirá la gestión de configuración del TDM, para comprobar que su diseño es acorde con las normas de gui/greeter.dtd a la hora de su implementación.	

Tarea	
Número tarea: 7	Número historia: U-TD-04
Nombre tarea: Estudio del fichero xml que define la estructura del ScreenSplash.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 7 noviembre 2007	Fecha fin: 21 noviembre 2007
Programador responsable: Equipo XP	
Descripción: Estudio de los diferentes howto para crear o personalizar el GDM, el cual servirá de base para la creación de un howto para la configuración del TDM.	

Tarea	
Número tarea: 8	Número historia: U-TD-04
Nombre tarea: Creación del prototipo de interfaz para gestionar la configuración del ScreenSplash.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 11 febrero 2008	Fecha fin: 25 febrero 2008
Programador responsable: Equipo XP	
Descripción: Creación del prototipo de interfaz que permitirá la gestión de configuración del ScreenSplash, para comprobar que su diseño es acorde con las normas de gui/greeter.dtd a la hora de su implementación.	

Tarea	
Número tarea: 9	Número historia: U-TD-05
Nombre tarea: Estudio del fichero xml que define la estructura del Escritorio.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 21 noviembre 2007	Fecha fin: 28 noviembre 2007
Programador responsable: Equipo XP	
Descripción: Estudio de los diferentes howto para personalizar el Escritorio.	

Capítulo 3: Planificación y Definición del Sistema.

Tarea	
Número tarea: 10	Número historia: U-TD-05
Nombre tarea: Creación del prototipo de interfaz para gestionar la configuración del Escritorio.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 12 mayo 2008	Fecha fin: 19 mayo 2008
Programador responsable: Equipo XP	
Descripción: Creación del prototipo de interfaz que permitirá la gestión de configuración del Escritorio, para comprobar que su diseño es acorde con las normas de gui/greeter.dtd a la hora de su implementación.	

Tarea	
Número tarea: 11	Número historia: U-TD-06
Nombre tarea: Estudio del fichero xml que define la estructura del Panel.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 28 noviembre 2007	Fecha fin: 5 diciembre 2007
Programador responsable: Equipo XP	
Descripción: Estudio de los diferentes howto para crear o personalizar el panel, el cual servirá de base para la creación de un howto para la configuración del mismo.	

Tarea	
Número tarea: 12	Número historia: U-TD-6
Nombre tarea: Creación del prototipo de interfaz para gestionar la configuración del Panel.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 19 mayo 2008	Fecha fin: 26 mayo 2008
Programador responsable: Equipo XP	
Descripción: Creación del prototipo de interfaz que permitirá la gestión de configuración del panel, para comprobar que su diseño es acorde con las normas de gui/greeter.dtd a la hora de su implementación.	

Tarea	
Número tarea: 13	Número historia: U-TD-07
Nombre tarea: Estudio del fichero xml que define la estructura de los temas de Metacity.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 5 diciembre 2007	Fecha fin: 12 diciembre 2007
Programador responsable: Equipo XP	
Descripción: Estudio de los diferentes howto para crear o personalizar los temas de metacity, el cual servirá de base para la creación de un howto para la configuración del mismo, con las particularidades de Tocororo - Desktop.	

Tarea	
Número tarea: 14	Número historia: U-TD-07
Nombre tarea: Creación del prototipo de interfaz para gestionar la configuración de los temas de Metacity.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 26 mayo 2008	Fecha fin: 9 junio 2008
Programador responsable: Equipo XP	
Descripción: Creación del prototipo de interfaz que permitirá la gestión de configuración de los temas Metacity, para comprobar que su diseño es acorde con las normas de gui/greeter.dtd a la hora de su implementación.	

Tarea	
Número tarea: 15	Número historia: U-TD-08
Nombre tarea: Estudio de la estructura para modificar el fondo de escritorio.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 5 diciembre 2007	Fecha fin: 12 diciembre 2007
Programador responsable: Equipo XP	
Descripción: Estudio de los diferentes howto para crear o personalizar el fondo de escritorio, el cual servirá de base para la creación de un howto para la configuración del mismo.	

Tarea	
Número tarea: 16	Número historia: U-TD-08
Nombre tarea: Creación del prototipo de interfaz para gestionar la configuración del fondo de escritorio.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 9 junio 2008	Fecha fin: 16 junio 2008
Programador responsable: Equipo XP	
Descripción: Creación del prototipo de interfaz que permitirá la gestión de configuración del fondo de escritorio.	

3.1.5 Plan de Lanzamiento.

En este apartado se define el plan de Lanzamientos e iteraciones para realizar las entregas intermedias y la entrega final. Tiene como entrada la relación de Historias de Usuario definidas previamente. Para colocar una historia en cada iteración se tiene en cuenta la prioridad que definió el cliente para dicha Historia. Además existen algunas tareas que se definen para lograr la realización satisfactoria de las Historias de Usuarios que se realizan en las primeras iteraciones y que no se colocan en el siguiente plan de Lanzamiento porque no constituyen Historias de Usuarios Independientes. Como resultado de la priorización de Historias se llegó a la siguiente planificación:

Lanzamiento	Historias de Usuarios	Tiempo estimado (semanas)
1		9
2		9
3	U-TD-04	9
4	U-TD-01 U-TD-02 U-TD-03 U-TD-05 U-TD-06 U-TD-07 U-TD-08	9

3.1.6 Tarjetas CRC.

Las tarjetas que definen las **Clase Responsabilidades Colaboradores (CRC)**, son una metodología para el diseño de software orientado por objetos creada por Kent Beck y Ward Cunningham. Como una extensión informal a UML, la técnica de las tarjetas CRC se puede usar para guiar el sistema a través de análisis guiados por la responsabilidad. Las clases se examinan, se filtran y se refinan basándose en sus responsabilidades con respecto al sistema, y las clases con las que necesitan colaborar para completar sus responsabilidades.

Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

En este sistema el uso de las tarjetas CRC no será necesario ya que los eventos de los botones de la interfaz no precisan la creación de clases. El sistema integrará en una sola interfaz elementos que se configuran en varias interfaces, haciendo llamados a métodos de

las clases definidas en la widget y que por tanto pertenecen a las librerías utilizadas, en este caso la librería GTK. Cada uno de los elementos que gestiona Tocororo –Desktop se encuentra especificado en la `gui/greeter/greeter.dtd` y en este caso solo se modificarán algunos valores de dicho archivo.

3.1.7 Prototipos de interfaz.

Los prototipos ayudan a identificar, comunicar y probar un producto antes de crearlo. Un prototipo es una representación limitada del diseño de un producto que permite a las partes responsables de su creación experimentar, probarlo en situaciones reales y explorar su uso. Puede ser desde un trozo de papel con sencillos dibujos a un complejo software. Son útiles para comunicar, discutir y definir ideas entre los diseñadores y las partes responsables, además de que apoyan el trabajo evaluando productos y clarificando las historias de usuario, además de ir definiendo alternativas.

Tocororo – Desktop es un sistema que posibilita la configuración de muchos de los componentes de un entorno de escritorio, y se hace necesario para ello crear interfaces que faciliten y amenicen esta tarea. Para cada una de estas interfaces se crea un prototipo, con el fin de ir sentando conceptos para facilitar el trabajo de los programadores del sistema. Por cada uno de las historias de usuario se tiene una interfaz. Las que se muestran a continuación.

Prototipo de interfaz del módulo principal:

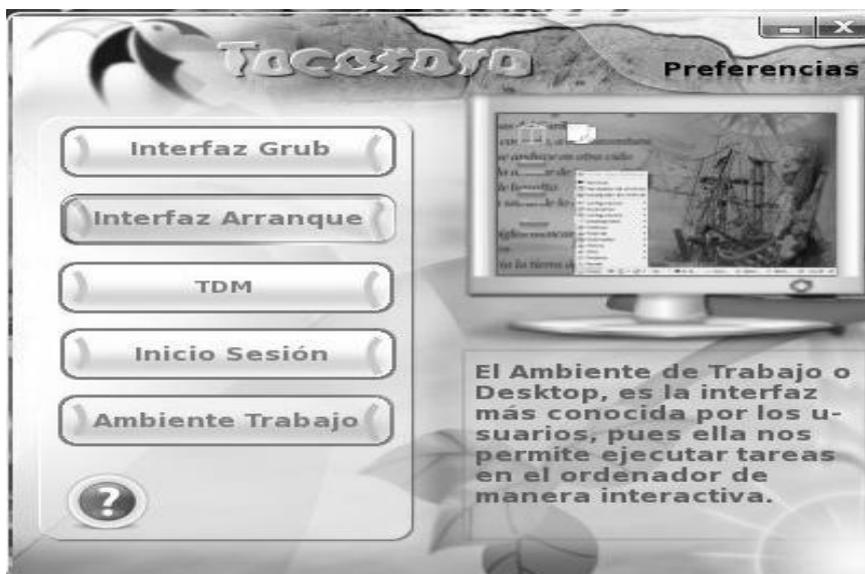


Imagen 2 Prototipo de interfaz del módulo principal.

Esta interfaz contiene botones a todas las demás interfaces. Es una interfaz amigable e interactiva que incluye la posibilidad de brindar información al usuario sobre que puede hacer en cada botón, con solo sostener el mouse encima de este, esta información será mostrada en la parte inferior derecha. Además cuenta con botón de ayuda para aquellos usuarios que tengan dudas de como utilizar la aplicación.

Prototipo de Interfaz del GrubSplash:

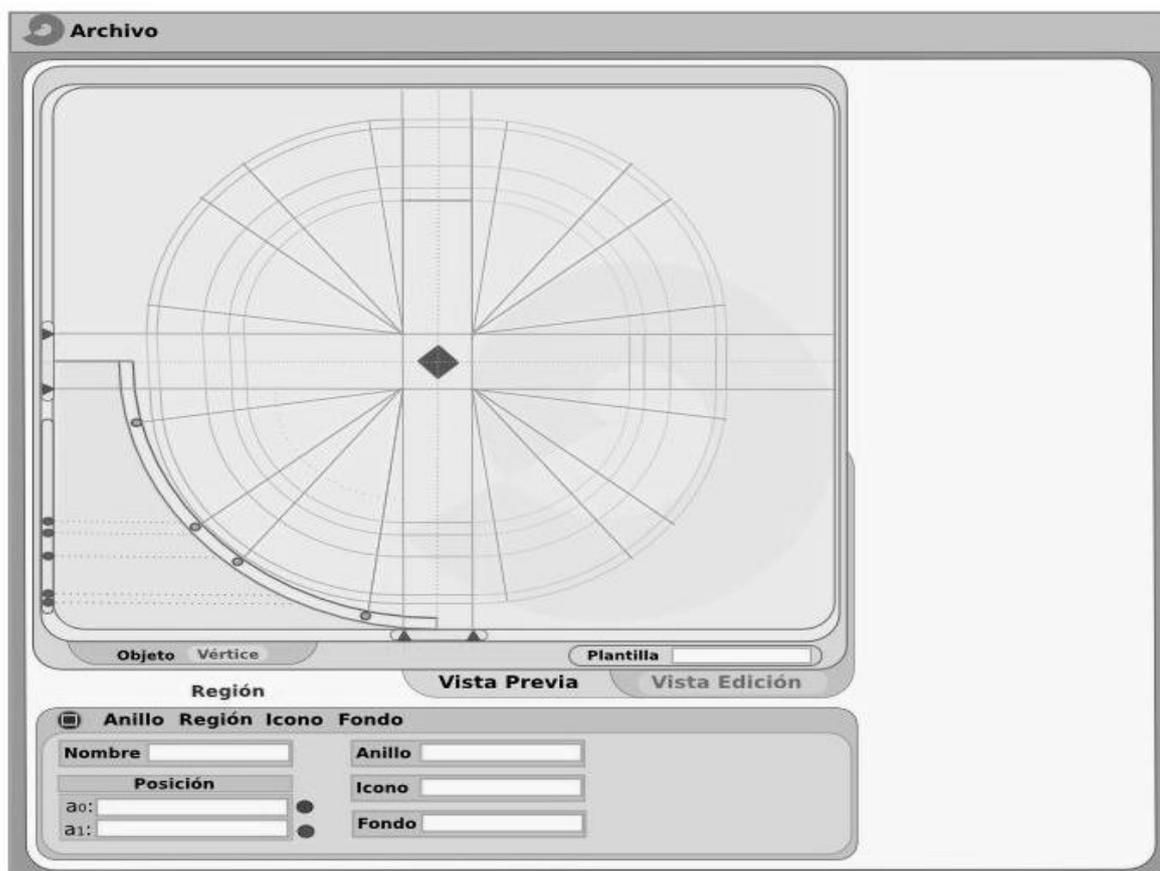


Imagen 3 Prototipo de interfaz del GrubSplash

El prototipo de interfaz del GrubSplash se carga al hacer clic sobre el botón que se encuentra en la interfaz principal con el nombre de Interfaz Grub. Este mostrará en la parte inferior un menú desplegable con todos los temas con los que cuenta la aplicación. Una vez seleccionado un tema del menú, se muestra en la pantalla de la primera computadora y, en la pantalla de la segunda computadora, el tema que actualmente está instalado. Luego de seleccionado el tema a instalar solo se necesitará dar clic en Añadir y el tema será instalado.

El resto de los prototipos de interfaces cumplirán con el mismo diseño del prototipo de interfaz del GrubSplash, con la diferencia de cargar los temas pertenecientes a las otras interfaces.

Prototipo de Interfaz del Gnus:



r0:
r1:

Imagen 4 Prototipo de interfaz del Gnus.

Al activar la interfaz de configuración del escritorio, se muestra la opción Configurar el Panel, responsable de la carga del prototipo de interfaz Gnus. Al aparecer la pantalla del prototipo se le permite al usuario cambiar las formas del panel lineal por una nueva representación, ya sea circular, triangular u otras; y elegir la posición en el escritorio y la forma de los iconos que se encuentran en él.

CONCLUSIONES.

La migración a Software Libre constituye un reto y una necesidad para Cuba, no solo desde el punto de vista moral y ético, sino también desde el punto de vista económico y social. En medio de la contienda de migrar a un sistema más justo para la libertad de usuario, es importante conseguir que las personas se familiaricen con este nuevo sistema.

Como resultado de este trabajo, se cuenta con una propuesta de diseño para la personalización de distribuciones de GNU/Linux en la enseñanza primaria cubana: Tocororo – Desktop que permitirá ayudar a nuestro país en el proceso de migración a plataformas libres. Esta propuesta ayudara también a incentivar en los niños la motivación por la tecnología.

Se investigaron minuciosamente los entornos de escritorios CDE, KDE, Xfce y Gnome de los sistemas operativos GNU/Linux, para demostrar la factibilidad del uso de este ultimo en la implementación de Tocororo –Desktop, esto permitió identificar todos los componentes configurables en el sistema operativo que afectan la imagen visual y diseñar el gestor de configuración de interfaces: Tocororo – Desktop para la personalización de los sistemas GNU/Linux en las primarias cubanas.

Por tanto, se concluye que los objetivos propuestos para el presente trabajo de diploma, han sido cumplidos satisfactoriamente.

RECOMENDACIONES.

Se hacen las siguientes recomendaciones.

- ✓ Se recomienda que el trabajo sea tomado como material de estudio para aquellas personas que realicen un sistema similar o que desarrollen versiones posteriores del mismo.
- ✓ Implementar Tocororo - Desktop sobre las especificaciones planteadas en el trabajo de diploma.
- ✓ Definir procedimiento y estrategia de trabajo para lograr la óptima utilización del servicio una vez implementado.
- ✓ Que las escuelas primarias y por consiguiente los organismos del estado, después del proceso de migración se acojan a este servicio.
- ✓ Utilizar la idea de Tocororo –Desktop para ser implementado sobre cualquier concepto o esfera del país.

REFERENCIAS BIBLIOGRÁFICAS.

[1] [Santiesteban, 2006] Santiesteban, Ma. Luisa. Educación Primaria. [En Línea] [Citado el: 20 de enero de 2008] [Disponible en: <http://www.rimed.cu/primaria/primaria.asp>.]

[2] [Olivé, 2005] Olivé, Jenisel. "Software libre: una opción para Cuba". Radio Habana Cuba [En Línea] [Citado el: 12 de diciembre de 2007] [Disponible en: <http://www.radiohc.cu/espanol/galerias/galeriainformatica2005/mayo05/11may.htm>.]

[3] [Stallman, 2000] Stallman, Richard Matthew. Por qué las escuelas deberían usar exclusivamente software libre. [En Línea] [Citado el: 28 de enero de 2008] [Disponible en: <http://www.gnu.org/philosophy/schools.es.html>.]

[4] [Negroponte, 2006] Negroponte, Nicholas. El Gobierno ratificó la compra de un millón del laptops a u\$s100 [En Línea] [Citado el: 22 de febrero de 2008] [Disponible en: <http://www.canal-ar.com.ar/Noticias/NoticiaMuestra.asp?Id=2960>]

BIBLIOGRAFÍA.

APUNTES, Metodologías de Desarrollo de Software Ágiles, [En Línea], [Citado el: 1 de abril de 2008], [Disponible en:

<http://www.sel.unsl.edu.ar/ApuntesMaes/2004/Metodologias%20Agiles.doc>].

BLOD DE WORDPRESS, Entornos de Escritorio, [En Línea], [Citado el: 2 de febrero de 2008], [Disponible en: <http://jgutgil.wordpress.com/?s=entornos+de+escritorios>]

BULMA, Arranque gráfico con Bootsplash, debian y barra de progreso, [En Línea], [Citado el: 1 de abril de 2008], [Disponible en: <http://bulma.net/impresion.phtml?nIdNoticia=1807>].

CORE SYSTEM, Config, [En Línea], [Citado el: 19 de abril de 2008], [Disponible en: <http://www.bootsplash.org/Config>].

CUADERNO DE NOVATO EN DEBIAN, Crear tu propio tema GDM, [En Línea], [Citado el: 12 de mayo de 2008], [Disponible en: <http://diegosamuel.blogspot.com/2008/02/crear-tu-propio-tema-gdm.html>].

DESARROLLO DE SOFTWARE, Expendedora: Modelo de Dominio, [En Línea], [Citado el: 17 de junio de 2008], [disponible en: http://209.85.215.104/search?q=cache:ammxHKmupxYJ:iiie.fing.edu.uy/ense/asign/desasoft/practico/hoja8/ejemplos_clase2.pdf+modelo+del+dominio&hl=es&ct=clnk&cd=3&gl=cu]

DEVELOPER'S, Designing Metacity Themes, [En Línea], [Citado el: 3 de marzo de 2008], [Disponible en: <http://developer.gnome.org/doc/tutorials/metacity/metacity-themes.html>].

EL LOCO MUNDO DEL TECHNO, Howto GDM theme, [En Línea], [Citado el: 3 de mayo de 2008], [Disponible en: <http://www.techno.mayix.net/howto-gdm-theme/>].

ESPACIO LINUX, Como diseñar un Splash para Kde, [En Línea], [Citado el: 10 de mayo de 2008], [Disponible en: <http://www.espaciolinux.com/postt5138.html>].

FUNDACIÓN BIP BIP, Escritorios de Linux, [En Línea], [Citado el: 6 de febrero de 2008], [Disponible en: http://www.puntoorg.org/escritorios_de_linux].

GNOME HISPANO, GDM Theme, [En Línea], [Citado el: 27 de abril de 2008], [Disponible en: http://www.es.gnome.org/DanielGalleguillos/GDM_Theme].

LAPTOP, One Laptop per Child (OLPC), [En Línea], [Citado el: 20 de febrero de 2008], [Disponible en: <http://www.laptop.org/laptop/interface>].

LAS LARGADAS DE SNEB, Creación de temas para gdm, [En Línea], [Citado el: 25 de abril de 2008], [Disponible en: <http://largadasdesneb.homelinux.org/?p=31>].

LIBRARY, F. S. Visual Paradigm for UML Professional Edition Details, [En Línea], [Citado el: 12 de marzo de 2008], [Disponible en: http://textver.filehungry.com/spanish/product/windows_software/programming/specialized_tols/visual_paradigm_for_uml_professional_edition].

MANUAL DE FREEBSD, Capítulo 5. El sistema X Window, [En Línea], [Citado el: 9 de abril de 2008], [Disponible en: <http://www.freebsd.org/doc/es/books/handbook/x11-wm.html>].

MONOGRAFIAS.COM, El escritorio, [En Línea], [Citado el: 16 de mayo de 2008], [Disponible en: <http://www.monografias.com/trabajos13/windos/windos.shtml>].

PARADIGM, V. Visual Paradigm for UML 6.0 SP1, [En Línea], [Citado el: 10 de marzo de 2008], [Disponible en: <http://www.visual-paradigm.com/news/vpsuite30sp1/vpuml60sp1.jsp>].

PARADIGM, V. Why Visual Paradigm for UML?, [En Línea], [Citado el: 10 de marzo de 2008], [Disponible en: <http://www.visual-paradigm.com/product/vpuml>].

SOFTWARE LIBRE VS SOFTWARE PROPIETARIO, Ventajas y desventajas, [En Línea], [Citado el: 21 de marzo de 2008], [Disponible en: <http://www.softwarelibre.cl/drupal/files/32693.pdf>].

SUN MICROSYSTEMS DOCUMENTATION, Common Desktop Environment (CDE), [En Línea], [Citado el: 15 de febrero de 2008], [Disponible en: <http://www.sun.com/software/solaris/cde/features.xml>].

UNIVERSIDAD POLITECNICA DE MADRID, Safety y Requisitos No Funcionales, [En Línea], [Citado el: 25 de abril de 2008], [Disponible en: <http://is.ls.fi.upm.es/doctorado/Trabajos20052006/Diez.pdf>].

VERSIONCERO, Visual Paradigm for UML, [En Línea], [Citado el: 3 de junio de 2008], [Disponible en: <http://www.versioncero.com/noticia/210/visual-paradigm-for-uml>]

WIKIPEDIA, DSL Compared to Microsoft Windows-es, [En Línea], [Citado el: 5 de abril de 2008], [Disponible en: http://www.damnsmalllinux.org/wiki/index.php/DSL_Compared_to_Microsoft_Windows-es].

WIKIPEDIA, Entornos de Escritorios, [En Línea], [Citado el: 15 de febrero de 2008], [Disponible en: http://es.wikipedia.org/wiki/Entorno_de_escritorio].

WIKIPEDIA, GRUB, [En Línea], [Citado el: 13 de abril de 2008], [Disponible en: <http://www.guia-ubuntu.org/index.php?title=Grub>].

GLOSARIO DE TÉRMINOS.

ARRANQUE DUAL: Doble booteo, Dual boot, Dual booting, Doble arranque o Arranque dual son distintas formas de llamar a la capacidad de una computadora para poder tener e iniciar con más de un sistema operativo funcionando en un mismo disco rígido o equipo.

CORBA: Es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.

DIAGRAMA: Es la representación grafica de un conjunto de elementos.

DOMINIO: Un dominio es un conjunto de ordenadores conectados en una red que confían a uno de los equipos de dicha red la administración de los usuarios y los privilegios que cada uno de los usuarios tiene en dicha red.

HERRAMIENTAS: Son los ambientes de apoyo necesario para automatizar las prácticas de Ingeniería de Software.

HERRAMIENTAS CASE: Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

IDE: Ambiente de desarrollo integrado. Es como se le llama al ambiente que proporciona al usuario una determinada herramienta de desarrollo.

INFORMACIÓN BURSÁTIL: adj. *Com.* Pertenciente o relativo a la bolsa, a las operaciones que en ella se hacen y a los valores cotizables. Se refiere a la información financiera de una empresa.

JAVA: Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 1990. Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible.

LICENCIAS GPL: Licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

MICROSOFT: Compañía de software más grande del mundo. Fue fundada en 1975 por Paul Allen y Bill Gates. Aunque también se conoce por sus lenguajes de programación y aplicaciones para computadores personales, el éxito sobresaliente de Microsoft se debe a sus sistemas operativos DOS y Windows.

METODOLOGÍA ÁGIL: Nuevo enfoque metodológico orientado a la gente y los resultados.

OPENOFFICE: OpenOffice.org es una suite ofimática de software libre, que incluye herramientas como procesador de textos, hoja de cálculo, presentaciones, herramientas para el dibujo vectorial y base de datos.

PERL: Lenguaje Práctico para la Extracción e Informe (ver abajo) es un lenguaje de programación diseñado por Larry Wall creado en 1987. Perl toma características del C, del lenguaje interpretado shell (sh), AWK, sed, Lisp y, en un grado inferior, muchos otros lenguajes de programación.

PYTHON: Lenguaje de programación creado por Guido van Rossum en el año 1990. Es comparado habitualmente con TCL, Perl, Scheme, Java y Ruby. En la actualidad Python se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation.

RAM: Se trata de una *memoria de semiconductor* en la que se puede tanto leer como escribir información. Es una memoria volátil, es decir, pierde su contenido al desconectar la energía eléctrica.

REQUERIMIENTO: Condición que el sistema debe cumplir o capacidad que debe tener.

SUITE OFIMÁTICA: Conjunto de software para el uso en oficinas y entornos profesionales. No hay una norma estricta sobre los programas a incluir, pero la mayoría incluyen al menos un procesador de textos y una hoja de cálculo. De forma añadida, la suite puede contener

un programa de presentaciones, un sistema de gestión de base de datos, herramientas menores de gráficos y comunicaciones, un gestor de información personal (agenda y cliente de correo electrónico) y un navegador web.

UML: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad.

UNIX: Es un sistema operativo portable, multitarea y multiusuario; desarrollado en principio por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

XML: Es el acrónimo de eXtensible Markup Language (lenguaje de marcado extensible) desarrollado por el World Wide Web Consortium (W3C).

ANEXOS.

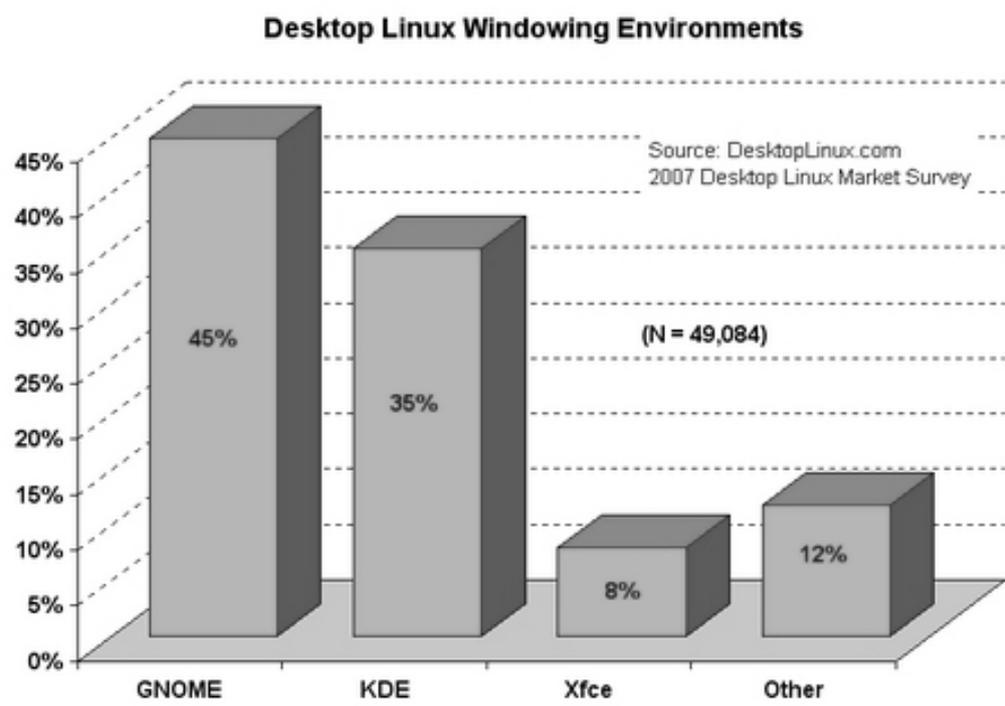
Anexo 1: AJUSTANDO LAS PROPIEDADES DE LA IMAGEN:



Anexo 2: HERRAMIENTAS DE ESCRITORIO:



Anexo 3: PORCENTAJE DE USABILIDAD DE GNOME:

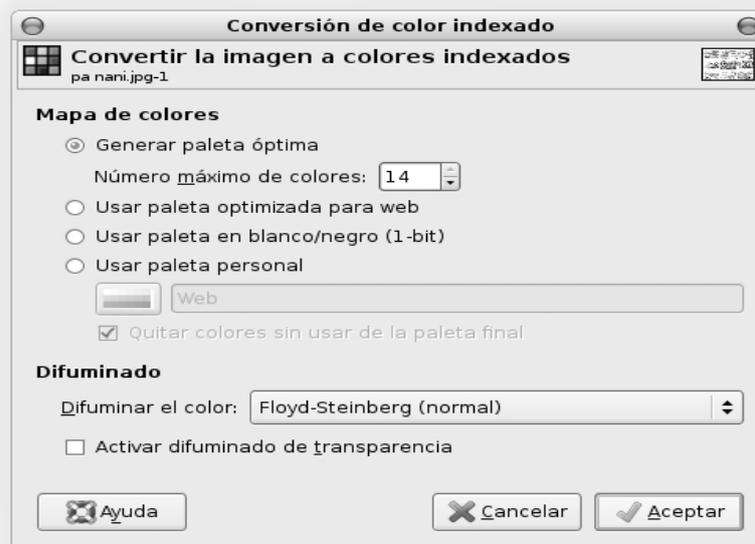


Anexo 4: AJUSTANDO LAS PROPIEDADES DE LA IMAGEN:

Formato: xpm



Número de colores: 14



Resolución: 640x480





Anexo 5: MODIFICANDO EL ARCHIVO MENU.LST:

```
splashimage=(hd0,#)/boot/grub/images/image.xpm.gz

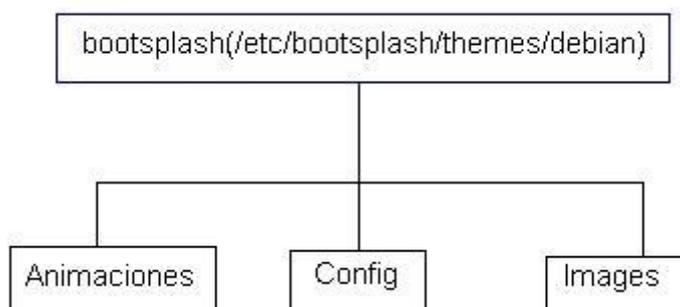
1. title Debian GNU/Linux, kernel
2.6.18-5-486
2. root (hd0,4)
3. kernel /boot/vmlinuz-2.6.18-5-486
root=/dev/sda5 ro vga=791
4. initrd /boot/initrd.img-
2.6.18-5-486
5. savedefault
```

Imagen del Grub que modifica y carga la imagen.

#: Es el número de la línea root, para el caso del ejemplo (hd0, 4)

Importante: Nótese que la línea que se adiciona está resaltada en otro color (verde).

Anexo 6: DIRECTORIO DEL BOOTSPASH



Anexo 7: PARTE DEL CONTENIDO DEL FICHERO DE BOOTSPASH:

```
# name of the picture file (full path recommended)
jpeg=/etc/bootsplash/themes/tocororo-desktop/images/bootsplash-1024x768.jpg
silentjpeg=/etc/bootsplash/themes/tocororo-desktop/images/silent-1024x768.jpg

progress_enable=0

# (ax, ay) are the (x, y) coordinates for playing boot/shutdown animations
ax=375
ay=145

# background
box silent noover 640 431 890 437 #04045498
box silent inter 640 432 640 436 #b6e3f0 #b6e3f0 #b6e3f0 #b6e3f0
box silent 640 432 890 436 #b6e3f0 #b6e3f0 #b6e3f0 #b6e3f0

# black border
box silent 640 431 890 431 #313234
box silent 640 437 890 437 #313234
box silent 640 431 640 437 #313234
box silent 890 431 890 437 #313234
```

Anexo 8: DECLARACIÓN DE LA IMAGEN EN EL FICHERO XML:

```
<item type="pixmap">
```

```
<normal file="background.jpg"/>
```

```
<pos x="0" y="0" with="100%" height="0"/>
```

</item>

Anexo 9: DEFINICIÓN DE IMÁGENES, RECTÁNGULOS Y ETIQUETAS CON LAS SIGUIENTES ETIQUETAS:

(Respectivamente)

<item type="pixmap">

<item type="rect">

<item type="label">

Anexo 10: USANDO LA ETIQUETA ITEM:

<item type="rect" id="language_button" button="true">

<item type="rect" id="session_button" button="true">

<item type="rect" id="system_button" button="true">

<item type="rect" id="disconnect_button" button="true">

<item type="rect" id="clock">

<item type="rect" id="caps-lock-warning">

<item type="rect" id="timed-rect">

<item type="rect" id="pam-prompt">

<item type="rect" id="user-pw-entry">

<item type="rect" id="pam-message">

<item type="rect" id="pam-error">

Anexo 11: REFERENCIA EN GDMGREETERTHEME.DESKTOP:

[GdmGreeterTheme]

Greeter=theme.xml

Name=Darkgno

Descriptio=A theme created with Blender

Author= Valia Cantillo y Yaneby Vázquez

Screenshot=GnomeGDM.jpg