



Universidad de las Ciencias Informáticas
Facultad X



Título: Solución para hacer de Guano un entorno de escritorio usable.

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

Autor(es): Miguel Albalat Aguila.
Barbarita Ramírez Pérez.

Tutor(es): Ing. Anielkis Herrera González
Msc. Graciela González Pérez.

Ciudad de la Habana
Curso 2008 – 2009



DECLARACIÓN DE AUTORIA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 10 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Barbarita Ramírez Pérez

Miguel Albalat Aguila.

MSc. Graciela González Pérez

Ing. Anielkis Herrera González

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado “Solución para hacer de Guano un entorno de escritorio usable”, fue realizado en La Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

Totalmente Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a _____.

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____.

Representante de la entidad

Cargo

Firma

Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: “Solución para hacer de Guano un entorno de escritorio usable”.

Autores: Barbarita Ramírez Pérez.

Miguel Albalat Aguila.

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan:

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de _____.

Firma del Tutor

MSc. Graciela González Pérez

Firma del Tutor

Ing. Anielkis Herrera González

Fecha

AGRADECIMIENTOS.

DEDICATORIA.

RESUMEN

La flexibilidad y usabilidad de las aplicaciones es un tema que se aborda cada día más en el ámbito del Software Libre. Así mismo han ido surgiendo poco a poco guías que marcan las pautas de un diseño amigable. En la UCI el proyecto Guano (entorno de escritorio libre y de código abierto) se ha dado a desarrollar una interfaz gráfica ligera que a su vez se nutra de estas guías amigables para la interacción con los usuarios.

Este trabajo aborda el proceso de desarrollo de herramientas que permitan al usuario de Guano interactuar amigablemente con su computadora, brindando un arma poderosa al proceso de migración de Software Libre en Cuba.

Palabras claves:

Entorno de escritorio, herramientas de configuración de escritorio, Guano.

ÍNDICE

INTRODUCCIÓN	12
1. FUNDAMENTACIÓN TEÓRICA	16
1.1. Entorno de Escritorio.....	16
1.1.1. Entornos de escritorio pesados.....	17
1.1.2. Entornos de escritorio ligeros.....	18
1.2. Características de Guano.....	18
1.2.1. Componentes de Guano.....	18
1.2.2. Encuesta realizada.....	20
1.3. Estado actual de las funcionalidades requeridas.....	21
1.3.1. Menú de aplicaciones.....	21
1.3.2. Aplicaciones de configuración de escritorio.....	26
1.3.3. Administrador de sesión de usuarios.....	30
1.3.4. Complementos o plugins.....	33
1.4. Lenguajes de programación y bibliotecas.....	33
1.5. Herramientas de desarrollo.....	38
1.6. Las Metodologías de Desarrollo de Software.....	40
1.6.1. Metodología Ágil.....	41
1.6.1.1. Programación Extreme (eXtreme Programming).....	42
1.6.1.2. Scrum.....	44
1.6.1.3. SXP o Metodología ágil para proyectos de software libre.....	46
2. ANÁLISIS Y DESARROLLO DE LA SOLUCIÓN PROPUESTA	48
2.1. Valoración crítica de las herramientas de Guano.....	48
2.2. Análisis y Desarrollo ágil de la Solución.....	49
2.2.1. Planificación y Definición del sistema.....	49
2.3. Lista de reserva del producto (LRP).....	52
2.4. Las historias de usuarios y los prototipos de interfaz.....	53
2.5. Lista de riegos.....	57
2.6. Planificación de las historias de Usuarios.....	59
2.6.1. Plan de Releases.....	59
2.7. Tareas de ingeniería.....	60
Iteración 1.....	60
Iteración 2.....	61
Iteración 3.....	62
2.8. Diseño con metáforas.....	64
2.9. Diagrama de paquetes.....	65
3 VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	67
3.1. Casos de Prueba.....	67
3.1. Resultados obtenidos.....	77
3.1.1. Acerca de las funcionalidades obtenidas.....	77
CONCLUSIONES GENERALES	79
REFERENCIAS BIBLIOGRÁFICAS	80
BIBLIOGRAFÍA	82
ANEXOS	83
ANEXO 1: ENCUESTA SOBRE MIGRACIÓN DE ESCRITORIO.....	83
ANEXO 3: MENÚ POR DEFECTO DE GNOME Y KDE.....	84

ÍNDICE.

ANEXO 3: MENÚ DE INICIO DE WINDOWS XP	84
ANEXO 4: HERRAMIENTAS DE CONFIGURACIÓN DE GNOME	85
ANEXO 5: HERRAMIENTA DE CONFIGURACIÓN DE XCEWM	86
ANEXO 6: HERRAMIENTA DE CONFIGURACIÓN DE ENLIGHTENMENT	87
ANEXO 7: HERRAMIENTAS DE CONFIGURACIÓN DE XFCE	87
ANEXO 8: HERRAMIENTAS DE CONFIGURACIÓN DE KDE	88
ANEXO 9: ADMINISTRADOR DE SESIÓN DE GNOME	89
ANEXO 10: APLICACIÓN DE ADMINISTRACIÓN DE SESIONES DE KDE	89
ANEXO 11: PRINCIPIOS DEL MANIFIESTO ÁGIL	90
ANEXO 12: PLANTILLA DE CONCEPCIÓN DEL SISTEMA	90
ANEXO 13: HERRAMIENTA DE CONFIGURACIÓN DE ESCRITORIO DEL PCMANFM	93
ANEXO 14: PROTOTIPO DE LA SOLUCIÓN PROPUESTA	93

ÍNDICE DE FIGURAS

FIGURA 1: GRÁFICO DE CLASIFICACIÓN DE LOS ENTORNOS DE ESCRITORIOS SEGÚN EL CONSUMO DE RECURSOS.	17
FIGURA 2. RESULTADO DE LA ENCUESTA.	20
FIGURA 3: GNOMENU, MENÚ ALTERNATIVO PARA GNOME.	24
FIGURA 4: KBFX, MENÚ ALTERNATIVO PARA KDE.	24
FIGURA 5: KICKOFF, MENÚ ALTERNATIVO PARA KDE.....	25
FIGURA 6. COMPARACIÓN DE LAS HERRAMIENTAS PARA UML EN LINUX.....	40
FIGURA 7: MODELO DE DOMINIO.	51
FIGURA 8. DIAGRAMA DE PAQUETES.	65

ÍNDICE DE TABLAS

ÍNDICE DE TABLAS

TABLA 1: CONSUMO DE MEMORIA DE LOS MENÚS.	25
TABLA 2: DEPENDENCIAS DE LOS MENÚS.	26
TABLA 3: CONJUNTO DE DEPENDENCIAS DE LAS HERRAMIENTAS DE CONFIGURACIÓN DE ESCRITORIO.	30
TABLA 4. DEPENDENCIAS QUE ACARREAN LA INSTALACIÓN DE LOS ADMINISTRADORES DE SESIÓN.	32
TABLA 5: LISTA DE RESERVA DEL PRODUCTO.....	53
TABLA 6. LISTA DE RIESGOS.....	58
TABLA 7: PLAN DE RELEASES.....	59

INTRODUCCIÓN

En la actualidad se está produciendo una profunda transformación social, económica y política motivada por el desarrollo de las Tecnologías de la Información y la Comunicación (TIC). Se está asistiendo a un proceso de grandes cambios tecnológicos que permiten configurar la denominada Sociedad de la Información.

Cuba, a pesar de no ser un país desarrollado, está llevando a cada rincón de la isla la revolución informática, desde los más pequeños hasta los más avanzados en edad, juegan un gran papel en el desarrollo de las TIC para el país, donde el capital humano es la mayor riqueza que posee.

El país, en esta lucha por desarrollar la informática ha llevado a cabo un gran proceso de compra de computadoras, con el fin de ponerlas al alcance de la mayoría de la población, pues el denominado Proyecto Futuro, que engloba en gran medida este esfuerzo, es una iniciativa en aras de mejorar la calidad de la educación y de la vida en general.

Dentro de este amplio escenario se encuentra la instalación de equipos de computación en todas las escuelas del país, incluso en lugares donde la electricidad aun no ha llegado. A esto se agrega la producción de abundantes softwares educativos, que cubren una amplia gama de necesidades en la educación primaria, secundaria, preuniversitaria y tecnológica, así como la ampliación de los estudios de formación en esta área, en especial con el incremento del número de institutos tecnológicos de Informática y la creación de la Universidad de las Ciencias Informáticas (UCI).

Producto del bloqueo impuesto por los Estados Unidos, el país ha estado limitado en la compra de computadoras, por lo que el hardware desplegado en la isla en muchos casos tiene más de cinco años de uso, por lo cual se consideran computadoras de bajas prestaciones. Al ser estas computadoras de bajas prestaciones, el país ha tenido un gran problema a la hora de escoger un sistema operativo, ya que con el avance de las tecnologías el desarrollo de los sistemas operativos es cada vez más grande, por lo que se necesita de un sistema operativo que permita adaptarlo a las necesidades y condiciones del hardware presente en el país.

Una solución a este problema es la migración al software libre, y con este la instalación del sistema operativo GNU/Linux, software que permite adaptar el sistema a nuestras necesidades.

En la Universidad de las Ciencias Informáticas, el proyecto Nova se ha dedicado a la creación de un sistema operativo libre cubano, que se pueda adaptar a las necesidades del país, condiciones de la máquina y requerimientos que solicite el centro donde está destinada para su futuro uso. A la hora de realizar la migración no solo se trata de encontrar un sistema operativo que de solución al problema, también la elección y optimización del entorno de escritorio, así como modificar la apariencia y la

INTRODUCCIÓN.

funcionalidad de este, es el punto clave para lograr que el usuario se sienta a gusto con el cambio.

En la actualidad existen muchos entornos de escritorio libres, pero son entornos de escritorios que en algunos casos consumen mucha memoria para poder ejecutarse o no están lo suficientemente humanizados para que el usuario se sienta a gusto a la hora de trabajar con este. Dada esta situación el proyecto Nova se ha dado la tarea de desarrollar Guano, un entorno de escritorio ligero, que pueda funcionar con un rendimiento adecuado en computadoras de bajas prestaciones y que a su vez sea una vía para apoyar la migración del país al software libre.

Guano en su arquitectura se define como un compendio de funcionalidades agregadas a aplicaciones ya existentes, es decir está basado en una arquitectura modular. Este está compuesto por un administrador de sesión, el manejador de ventanas, el manejador de ficheros y el panel.

Actualmente Guano no tiene una interfaz amigable para que los usuarios finales se sientan a gusto trabajando con él, este necesita de una serie de funcionalidades que permitan una mejor interacción entre el usuario y la máquina.

A partir de aquí surge la siguiente **situación problémica**: Guano, el entorno de escritorio alternativo para Nova no cuenta con las funcionalidades necesarias para una interacción simple y cómoda del usuario con el entorno de escritorio, y carece de herramientas simples y cómodas que permitan modificar su comportamiento.

Lo que conduce al siguiente **problema científico**: ¿Cómo lograr la integración de la usabilidad y las funcionalidades requeridas por el usuario en Guano conservando un bajo consumo de recurso de los ordenadores?

Para la resolución del problema científico se asume como **objeto de estudio**, las herramientas de configuración de escritorio y las aplicaciones para mejorar la interacción del usuario con el ordenador, y como **campo de acción** las herramientas de configuración de escritorios de software libre.

Se trazó como **objetivo general**, desarrollar el paquete de herramientas para la configuración y personalización del entorno de escritorio Guano.

Desglosándose así los siguientes **objetivos específicos**:

- Sistematizar el estado actual de las herramientas para configurar los entornos de escritorios.
- Analizar, diseñar y reutilizar soluciones viables a la problemática expuesta en el objetivo general.
- Desarrollar aplicaciones para configurar y personalizar el escritorio Guano.
- Automatizar el proceso de instalación de las aplicaciones desarrolladas.

INTRODUCCIÓN.

Las **preguntas científicas** que se plantean:

- ¿Cuáles son los antecedentes y estado actual de las herramientas de configuración?
- ¿Cuáles son los antecedentes y estado actual de las herramientas de configuración de escritorio en los entornos libres?
- ¿Cómo desarrollar el paquete de herramientas de configuración del escritorio para Guano?

Para dar cumplimiento a los objetivos general y específicos planteados se han definido una serie de **tareas de investigación**, las cuáles se muestran a continuación:

- Sistematizar los antecedentes y estado actual de las herramientas de configuración.
- Sistematizar los antecedentes y estado actual de las herramientas de configuración en los entornos libres.
- Desarrollar el paquete de configuración de escritorio para el entorno de escritorio Guano.

Esta investigación tuvo como soporte procedimientos que permitieron estudiar las características del objeto de la investigación que no son observables directamente y dar cumplimiento a las tareas propuestas con anterioridad. Los métodos científicos y las técnicas empleados en el desarrollo de este estudio se describen a continuación:

- El **método analítico – sintético** permitió descomponer el problema en sus partes y realizar un mejor análisis del funcionamiento de las herramientas de configuración de escritorio, para así entender mejor sus funcionalidades y características, y luego en su conjunto llegar a generalizaciones.
- **Análisis bibliográfico:** Este método permitió realizar una revisión y análisis ordenado de las diferentes bibliografías referentes a las herramientas de configuración de escritorios existentes para posteriormente extraer y sintetizar los elementos más significativos teniendo en cuenta el objeto de estudio.
- **Histórico-lógico:** Este método posibilitó conocer los antecedentes y tendencias más actuales de las herramientas de configuración de escritorios en el mundo del software.
- El **hipotético – deductivo** para inferir conclusiones y establecer predicciones sobre las herramientas que permiten personalizar el escritorio.
- **Observación:** Esta técnica permitió tener un registro visual del comportamiento de algunas herramientas de configuración de los entornos de escritorios.
- **Encuesta:** Esta técnica se utilizó para obtener información relevante para el desarrollo de la

INTRODUCCIÓN.

investigación consultando a personas para tener conocimiento y dominio de cuáles son las funcionalidades requeridas por los usuarios finales.

El presente trabajo consta de tres capítulos los cuales en su conjunto abordarán todo el contenido referente al estudio y desarrollo de una solución para hacer de Guano un entorno de escritorio usable.

Capítulo 1: “FUNDAMENTACIÓN TEÓRICA”: Contiene la fundamentación teórica de la investigación, se exponen las técnicas, tecnologías, metodologías y software en los que se apoya la solución propuesta para el problema. Se realiza un análisis crítico y valorativo de las tendencias actuales y el estado del arte de las herramientas de configuración en los entornos de escritorios.

Capítulo 2: “ANÁLISIS Y DESARROLLO DE LA SOLUCIÓN PROPUESTA”: En este capítulo se realiza el desarrollo ágil de la solución. Se explica toda la dinámica del proyecto en forma de historias de usuarios, prototipos de interfaz de usuario y algunos modelos auxiliares además del plan de liberación para las entregas intermedias.

Capítulo 3: “VALIDACIÓN DE LA SOLUCIÓN PROPUESTA”: En este capítulo se detallan las pruebas realizadas al software, como de tipo caja negra, así como el impacto que tendrá el sistema una vez implantado.

1. FUNDAMENTACIÓN TEÓRICA.

El software libre es un proyecto basado en una filosofía de cooperación y aportación desinteresada que promueve la libertad y pretende elevar el concepto de la creación de software a los niveles más altos de las virtudes humanas como son el altruismo y la cooperación. Este, al dar acceso al código es el único que puede garantizar esta transparencia y brinda además independencia tecnológica.

La filosofía que rodea todo el movimiento de Software Libre permite un cambio de mentalidad dentro de la sociedad que afianza en el individuo el compartir, el desprenderse y el poder interactuar con otras personas para mejorar los problemas, poniendo los conocimientos y las posibilidades tecnológicas existentes al alcance de todos en la sociedad.

En los últimos años con el surgimiento de Linux y su combinación con el sistema de GNU, el software libre ha adquirido una nueva dimensión de índole social. Es incluso posible afirmar que el Software Libre ya no es solamente un paradigma tecnológico, sino también un paradigma social. La unión de ambos resultó en un sistema operativo libre, completo y usable. Por lo tanto, los sistemas Linux en la actualidad se suelen denominar GNU/Linux, es decir, un sistema operativo libre formado por un núcleo Linux y las aplicaciones GNU.

Producto de las libertades que brinda el software libre podemos encontrar una gran variedad de softwares que comparten el mismo fin, esta diversidad de softwares se debe a que están orientados a diferentes usuarios o computadoras con diferentes prestaciones. En la actualidad existen muchas distribuciones de GNU/Linux orientadas a grupos específicos de usuarios, dando así origen a ediciones hogareñas, empresariales y para servidores. Cada una de ellas suele traer incorporado un entorno de escritorio, donde el sistema operativo brinda la posibilidad en muchos casos de utilizar varios entornos de escritorio, según el rendimiento de la computadora, al público que estará dirigido, así como las tareas que se quiere realizar con el sistema operativo.

Escoger un entorno de escritorio adecuado a las condiciones de la máquina es un paso muy importante, ya que este define el rendimiento de una distribución.

1.1. Entorno de Escritorio.

Según www.wikipedia.org, se define que: *“Un entorno de escritorio es un conjunto de software que ofrece a los usuarios de una computadora una interacción amigable y cómoda. El software es una solución completa de interfaz gráfica de usuario o GUI, ofrece iconos, barras de herramientas,*

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

programas e integración entre aplicaciones con habilidades como, arrastrar y soltar (drag&drop). En general cada entorno de escritorio se distingue por su aspecto y comportamiento particular, aunque algunos tienden a imitar características de otros escritorios ya existentes. Actualmente el entorno más conocido es el ofrecido por la familia Windows aunque existen otros como los de Macintosh (Classic y Cocoa) y de código abierto (o software libre): GNOME, KDE, CDE o Xfce.”

Para una mejor comprensión del presente trabajo se decidió hacer una clasificación de los entornos de escritorios. Basándose en el rendimiento, los entornos de escritorios pueden ser agrupados en dos grandes grupos: **Entornos ligeros** y los **Entornos pesados**.

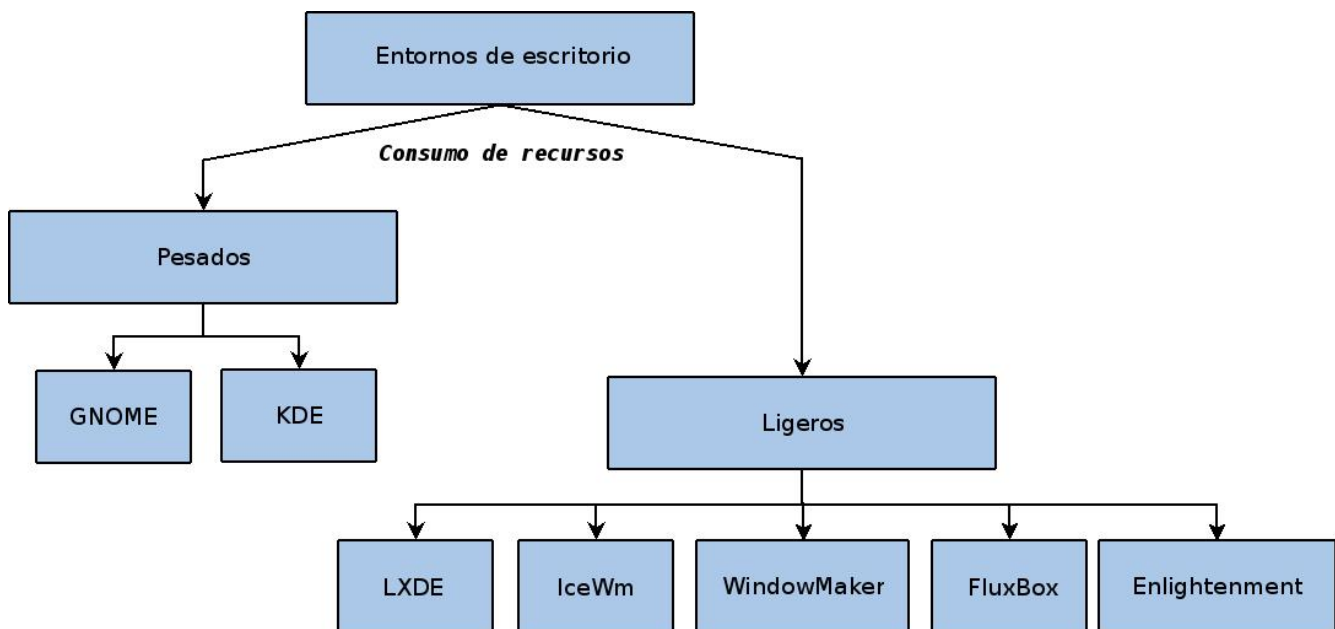


Figura 1: Gráfico de clasificación de los entornos de escritorios según el consumo de recursos.

1.1.1. Entornos de escritorio pesados.

Además del manejador de ventana traen numerosas herramientas integradas entre sí que proporcionan todo lo necesario para trabajar en el entorno gráfico, como iconos, el administrador de sesión (que se encarga de guardar el estado de las aplicaciones cuando se sale de la sesión y restaurarlas cuando el usuario entra de nuevo), el manejador de ficheros, un framework¹ y aplicaciones como editores de textos, navegador, multimedia, gráficos, etc.

Los entornos de escritorio pesados necesitan más recursos para ejecutarse que los entornos de escritorio ligeros (al menos 128 MB de RAM), se configuran mediante herramientas gráficas y los más

¹ Framework: Conjunto de librerías y componentes que permite diseñar, construir e implantar aplicaciones corporativas de forma más uniforme, rápida, y con mayor calidad.

utilizados son:

- GNOME
- KDE

1.1.2. Entornos de escritorio ligeros.

Constan del manejador de ventana y pocas aplicaciones gráficas. Son rápidos, ligeros y estables, con un consumo muy bajo de recursos (sólo necesitan 32-64 MB de RAM), la mayoría no tienen iconos y normalmente se configuran editando manualmente los archivos de configuración. Algunos son:

- IceWm.
- FluxBox y otros derivados del BlackBox, como OpenBox y Waimea.
- WindowMaker y otros similares (Twm, Vtwn, Fvwm, Fvwm-Crystal, AfterStep).
- Enlightenment.
- Lxde.
- Xfce

Debido a la gran diversidad de entornos de escritorios que existen actualmente en el mundo del software libre y que ninguno de los antes mencionados se adapta a las necesidades que debe cubrir un entorno de escritorio para las características muy peculiares con que cuenta el hardware en el país y los que son ligeros no están dirigidos a los usuarios finales, usuarios que no poseen muchos conocimientos de los entornos libres; en la Universidad de las Ciencias Informáticas se está desarrollando Guano, un entorno de escritorio ligero que será usado en la distribución Cubana de GNU/Linux, Nova, que actualmente se encuentra desarrollando el proyecto del mismo nombre (1).

1.2. Características de Guano.

Guano es un entorno de escritorio que está en pleno desarrollo, su diseño está basado en una arquitectura modular compuesta por un administrador de sesión, el manejador de ventana, el manejador de fichero y el panel. Debido al tipo de estructura que tiene, a este se le pueden insertar nuevos componentes que se detecten en la amplia gama de los proyectos libres que existen, atendiendo al rendimiento en hardware de bajas prestaciones, usabilidad y calidad de las funcionalidades requeridas.

1.2.1. Componentes de Guano.

Manejador de Ficheros.

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

Uno de los elementos más importantes en la construcción de un entorno de escritorio es la concepción de un manejador de ficheros ya que este es el encargado de explorar las carpetas, relacionar ficheros con programas para su reproducción, gestionar los eventos de arrastrar y soltar iconos además de administrar la papelera de reciclaje. En la actualidad existe una gran cantidad de manejadores de ficheros en el mundo del software libre, la mayoría de los que poseen una mayor potencia y funcionalidades ya pertenecen a un entorno de escritorio, por lo que Guano usa como manejador de fichero al PcmnFM debido a que este se encuentra independiente de cualquier entorno de escritorio y por ser una alternativa con una arquitectura ligera y características innovadoras.

Gestor de ventanas.

Otro elemento importante del entorno de escritorio es el manejador de ventanas, el cual es un programa que controla la ubicación y apariencia de las aplicaciones bajo el Sistema de Ventanas X², así como los múltiples espacios de trabajo virtuales. Hoy en día existen diversos gestores de ventanas, casi todos bastante minimalistas que resultan muy ligeros, prácticos y potentes pero rara vez resultan totalmente configurables, por lo que Guano escoge a Openbox, ya que posee una muy amigable e intuitiva forma de configuración. Este está diseñado para ser rápido y consumir una mínima cantidad de recursos. Ofrece menús generados dinámicamente capaces de ofrecer información variada. Posee la peculiaridad de permitir cambiar casi todos los aspectos de cómo interactuar con el escritorio y completamente inventar nuevas formas de uso y control.

Panel

En un entorno de escritorio estándar el panel es el encargado de suministrar un menú de aplicaciones, un espacio para el reloj, los iconos de notificaciones, accesos directos a aplicaciones, el paginado de escritorio, applets para el control de volumen, el acceso a red y la importantísima barra de tareas donde se sitúan las aplicaciones minimizadas. Los proyectos de paneles que no están ligados a un entorno de escritorio específico son prácticamente inexistentes, por lo Guano escoge el panel de Xfce, por ser un entorno de escritorio rápido y ligero para sistemas operativos basados en Unix, diseñado para Freedesktop³ y que posee un panel completo e intuitivo que ofrece una forma sencilla de añadir elementos mediante un cuadro de diálogo, y para mover elementos a través de diferentes paneles, además gracias a la concepción modular de Xfce el panel puede ser importado dentro del proyecto sin el acarreo de numerosas dependencias.

² El sistema de ventanas X (en inglés X Window System) fue desarrollado a mediados de los años 1980 en el MIT para dotar de una interfaz gráfica a los sistemas Unix.

³ Freedesktop: Es un proyecto que trabaja por la interoperabilidad y la compartición de la tecnología base de los entornos de escritorio para X Window System, tanto en Linux como en otros sistemas operativos Unix.

1.2.2. Encuesta realizada.

En el mundo real, cada usuario que trabaja sentado frente a un escritorio convencional lo organiza y mantiene sus pertenencias como mejor le sea posible para desempeñar su trabajo eficiente y cómodamente. En el mundo informático sucede lo mismo, cada cual organiza su estación de trabajo como mejor le sea posible para desarrollarse adecuadamente. El escritorio de un ordenador es una abstracción de la realidad, cada objeto visto y herramienta usada es un objeto virtual, que desempeña un rol diferente y específico siguiendo patrones de diseño así como de usabilidad.

Basándose en la encuesta realizada por el equipo de desarrollo de Guano, sobre cuáles son las funcionalidades que se quieren conservar en el entorno de escritorio, cuando migre al software libre (Ver [Anexo 1. Encuesta sobre migración de escritorios](#)) se obtuvieron los siguientes resultados.

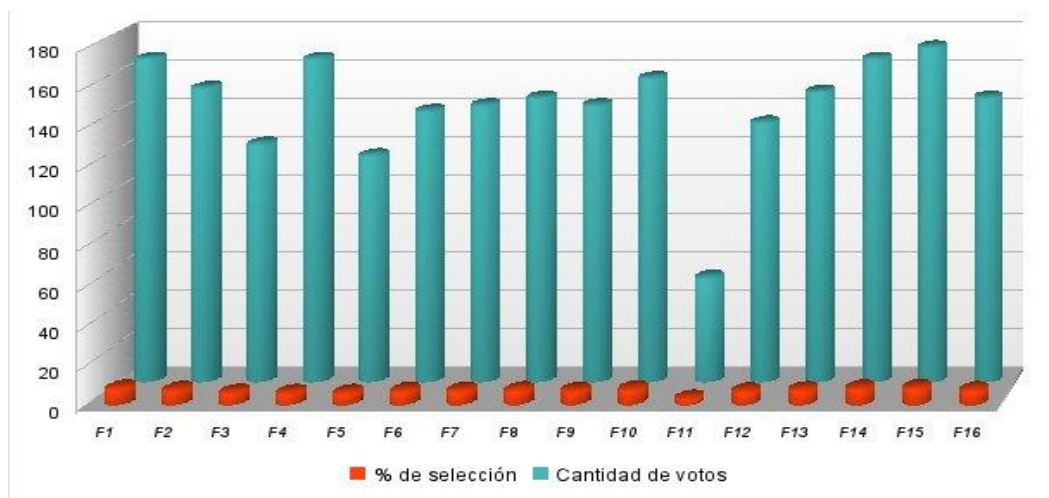


Figura 2. Resultado de la encuesta.

De 200 personas encuestadas, con un total de 2254 votos. Se hizo un mayor énfasis en las propuestas a las que el equipo de desarrollo de Guano por no tener mano de obra disponible no pudo dar respuesta inmediata:

- Poder cambiar la apariencia y configuración del escritorio (colores, fondo de pantalla, resolución del monitor, estilo, etc.) Con 7.49% (166 votos).
- Conservar la estructura del menú y que recuerde mis aplicaciones recientes y preferidas. Con 5.24% (166 votos).
- Tener la papelera de reciclaje. Con 5.24 % (116 votos).

Debido al desconocimiento de los usuarios finales sobre qué es un administrador de sesión de usuarios y a que no se incluía como parte de la encuesta realizada por el equipo de desarrollo de

Guano, se decide debido a su importancia; incorporarlo como requisito importante a tener en cuenta en el desarrollo del presente trabajo. Es necesario además aclarar que la propuesta de la necesidad de soporte de papelera de reciclaje se decide que solamente va a estar dirigido a mejorar la interacción de los usuarios con la papelera a través de plugins en el panel, debido a que la funcionalidad es una tarea a realizar por el equipo de desarrollo de Guano; encontrándose en la misma situación el manejo del demonio Guano-sesion encargado de gestionar el inicio de las sesiones de usuarios.

1.3. Estado actual de las funcionalidades requeridas.

Para realizar el estudio sobre el estado actual de las propuestas antes expuestas se decidió basarse en la arquitectura de la información⁴, como un marco de apoyo para realizar una crítica al diseño y la funcionalidad de las aplicaciones que se pudieran tomar como un punto de partida si fuera factible su utilización. Para esto se utilizaron los criterios expuestos en la HIG (Human Interfaces Guinelines), pautas por las que se guían el desarrollo de interfaces para usuarios del proyecto Nova y que son provechosas en la construcción de las aplicaciones para Guano.

1.3.1. Menú de aplicaciones.

Menú

En términos informáticos, un menú es una serie de opciones que el usuario puede elegir para realizar determinada tarea. Los menús se organizan siguiendo el principio de los árboles, esto quiere decir que un menú puede tener menús hijos y menús padres. Inicialmente, al crear una barra de menús, el identificador del objeto nuevo es un identificador de menús válido, además de que la barra de menús queda como la raíz o el nodo principal.

Los menús pueden contener los siguientes elementos:

- Submenús.
- Lanzadores o elementos de menú.

Una flecha situada a la derecha de un elemento de menú indica que se trata de un submenú. Al apuntar al submenú, éste se abre y se pueden seleccionar los elementos que contiene.

Al seleccionar un lanzador, éste arranca una aplicación o ejecuta un comando. También se puede

⁴ Arquitectura de la información: Es definida como el arte y la ciencia de organizar espacios de información con el fin de ayudar a los usuarios a satisfacer sus necesidades de información.

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

hacer clic con el botón derecho del mouse sobre un lanzador para abrir un menú emergente.

Menú emergente de elementos del menú

Los elementos de los menús llevan asociado un menú emergente que permite realizar las tareas relacionadas. Cuando se abre un menú emergente asociado a un elemento este permite añadir un elemento al panel, así como realizar otras tareas como:

- Agregar elementos de menú a paneles en forma de lanzadores.
- Quitar elementos de menús.
- Abrir el cuadro de diálogo.
- Ejecutar programa con el comando elemento de menú.
- Agregar menús a los paneles. Puede agregarse un menú como objeto de menú o como objeto cajón.
- Agregar nuevos elementos a los menús.
- Cambiar las propiedades de los submenús y elementos de menú.

Menú de aplicaciones o de inicio

El **menú de aplicaciones**, **menú Inicio** o **botón de inicio** como se les suelen llamar, son los elementos de la interfaz de usuario que sirven de punto central de lanzamiento de las aplicaciones y las tareas.

Tradicionalmente, el **menú de aplicaciones** provee una lista personalizable y anidada de programas que el usuario puede poner en marcha, así como una lista de la mayoría de los documentos abiertos recientemente; una manera de encontrar los archivos y obtener ayuda, así como tener acceso a la configuración del sistema.

Técnicamente, el menú Inicio no es necesario, ya que todos los programas y archivos se pueden abrir mediante la navegación a ellos en el manejador de ficheros de cualquier distribución o Sistema Operativo. Sin embargo, el menú Inicio proporciona una vía mucho más fácil y consolidada de ejecutar programas, incluso para usuarios experimentados. En el caso de Microsoft este "utiliza el menú Inicio en cada versión de Windows como una manera de proteger a los usuarios novatos de las complejidades del sistema operativo; por ejemplo, en Windows XP, la raíz, Archivos de programa y carpetas de Windows están ocultos al usuario por defecto, y el acceso a los programas se logra a través del menú Inicio (2), por otro lado en otros sistemas operativos se emplea con el mismo objetivo, no llamándose del mismo modo, aunque también cabe destacar que es debido a la

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

estandarización del mismo por parte de Microsoft.

Panorámica actual del menú de aplicaciones.

En la actualidad se ha hecho un estándar que todos los entornos de escritorios incluyan un menú de aplicaciones, ya que este es el principal mecanismo mediante el cual los usuarios pueden descubrir y ejecutar aplicaciones además de ser la forma más rápida de hacerlo. Para realizar el estudio de esta aplicación, se hizo hincapié en el menú de aplicaciones de los entornos de escritorios que cuentan con una mayor aceptación (GNOME y KDE) y siguen el patrón de diseño del menú de Windows XP (Ver [Anexo 3. Menú de inicio de Windows XP](#)), dejando a un lado los menús por defecto de estos entornos (Ver [Anexo 2. Menú por defecto de los entornos GNOME y KDE](#)), debido a que se quiere que esta aplicación sea lo más parecida a lo que los usuarios que migran por primera vez estén acostumbrados a ver, haciendo el proceso menos “doloroso”.

Se tomó en cuenta dos criterios para realizar el estudio:

- Arquitectura jerárquica.
- Funcionalidad que brinda.

Gnomenu.

Es un menú para el panel del entorno de escritorio GNOME, ideal para el usuario nuevo de Linux o que provienen del sistema operativo Windows. Tiene ciertas reminiscencias del panel de Windows Vista. El menú de aplicaciones es similar al original de GNOME, y aparece ahora en el lugar que ocupaban los programas más utilizados en el menú de Windows Vista. Ofrece acceso directo a las carpetas almacenadas en el directorio home y tanto el aspecto de la interfaz como el de los botones e iconos puede ser cambiado desde las preferencias de la aplicación, ofreciendo para cada elemento varias opciones a escoger. Mantiene los botones de acceso para salir de la sesión o bloquear (3).



CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

Figura 3: Gnomenu, menú alternativo para GNOME.

KBFX

Es un menú alternativo para el entorno de escritorio KDE que reemplaza el botón y menú con el logo K (similar a Windows 95) con una versión más grande y más visible, sin embargo, mantiene el menú tradicional de KDE como una opción. Posee una interfaz altamente personalizable. Tiene una barra vertical, llamada Spinx menú, donde pueden encontrar en su parte superior la imagen seleccionada previamente por el usuario al momento de registrarse mediante el login y, debajo de ella, todas las categorías de software instalado en el sistema. Mantiene la arquitectura jerárquica del menú original de KDE; al momento de elegir una categoría los nombres de los programas van apareciendo en su costado de modo ascendente, para después poder navegarlos a través del movimiento del scroll del mouse y así poden ser seleccionados con tan sólo un click. También posee una ventana de comandos en la parte inferior, en donde se pueden llamar a funciones mediante una línea de comandos o usar algunas de las que ya trae predefinidas. Otra característica muy buena es el primer apartado del menú, donde se ven los programas recientes que hemos usado, para así tener un acceso más rápido a ellos (4).



Figura 4: KBFX, menú alternativo para KDE.

Kickoff

OpenSUSE 10.2 ha rediseñado un nuevo menú de inicio para KDE mucho más potente y con características similares al menú de Windows Vista. Este está gobernado por una caja de búsqueda, y otras 4 pestañas: Favoritos (Programas, Carpetas, Archivos, Configuración), Recientemente usados, un explorador de archivos integrado, el típico menú con todos los programas y las opciones de apagado (5).



Figura 5: Kickoff, menú alternativo para KDE.

A continuación se hace una comparación del consumo de memoria de la computadora estimado de estos menús, así como las dependencias que estos acarrearán.

Menú	Entorno de escritorio	Memoria utilizada(Mb)
Gnomenu	GNOME	20
KBFX	KDE	30
Kickoff	OpenSuse	30

Tabla 1: Consumo de memoria de los menús.

Menú	Dependencias
Gnomenu	Gnomenuthemes python2.5 python-xdg pycairo python-gconf python-xlib deskbar-applet

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

KBFX	kdebase kdelibsc makestrigi
Kickoff	Kdnss Dlib Konqk Delib sbeagle

Tabla 2: Dependencias de los menús.

Como se puede observar, ninguna de las opciones antes mostradas corresponde una solución factible a usar en Guano, lo demuestra el alto consumo de memoria y el gran número de dependencias que acarrearán cada uno por separado. Ahora, la manera de mostrar y organizar sus principales elementos constituyen su principal fuerza, además de ser la forma predominante que se emplea, también por el parecido con el Menú de Inicio de Microsoft Windows, creándose el hecho de que no sean rechazados por parte de los usuarios que migran a plataformas libres. Por lo que como recomendación se plantea desarrollar un menú con las mismas prestaciones que requiere un menú de inicio como los antes expuestos siguiendo las características que se necesitan para las aplicaciones construidas para Guano.

1.3.2. Aplicaciones de configuración de escritorio.

Las herramientas de configuración de escritorio son aplicaciones primordiales, estas son las que brindan la posibilidad al usuario de personalizar el escritorio según sus gustos y preferencias. Se puede generalizar que el número de herramientas de configuración de escritorio son inherentes al número de escritorios que existen, debido a que cada escritorio tiene sus características configurables propias, por lo que existen una gran cantidad y variedad de las mismas. Ejemplos de las principales herramientas, con sus respectivos entornos de escritorios se describen a continuación:

- **GNOME** es un entorno de escritorio completo, con excelentes herramientas gráficas: **gnome-appearance-properties**, **gnome-session-properties** y **gnome-window-properties** (Ver [Anexos 4. Herramientas de configuración de GNOME](#)); que permiten configurar de forma fácil y rápida las diferentes opciones del escritorio. Estas aplicaciones de configuración están desarrolladas para ser usadas tanto por expertos como por usuarios noveles en el sistema operativo GNU/Linux. El principal inconveniente que

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

presentan estas herramientas es que necesitan de muchos recursos de memoria para poder funcionar satisfactoriamente, por lo que se vuelve tedioso trabajar con estas en máquinas de bajas prestaciones.

- **XceWM** es un entorno de escritorio ligero que posee varias herramientas gráficas que permiten configurar las opciones generales del escritorio como **IceConf** o **IcePref** (Ver [Anexos 5. Herramienta de configuración de XceWM](#)). Estas herramientas no tienen en cuenta varias de las pautas dictadas por la HIG, como hacer aplicaciones simples, intuitivas y bonitas, no abarcan todas las configuraciones posibles que se le pueden realizar al escritorio, y las opciones que no presenta se tienen que realizar manualmente, proceso que es muy complejo para usuarios sin mucho conocimiento sobre el sistema operativo GNU/Linux. Además estas aplicaciones tienen una interfaz muy rústica y antiestética, por lo que el usuario tiende a desanimarse a la hora de interactuar con ellas.
- **Enlightenment** es un entorno de escritorio ligero que contiene unos efectos gráficos espectaculares y se configura totalmente con herramientas gráficas, pero presenta varios puntos en su contra, cuando se desea cambiar el fondo de escritorio se vuelve tedioso trabajar con la herramienta destinada para esta función (Ver [Anexo 6. Herramienta de configuración de Enlightenment](#)) ya que para agregar uno nuevo se debe colocar primeramente en el directorio `/.enlightenment/backgrounds/` y luego reiniciar la sesión para que este los detecte. Para añadir un nuevo tema de ventana es el mismo proceso, primero debemos colocarlos en el directorio `/.enlightenment/themes/<tema>/` y después reiniciar la sesión para cargar los cambios.

Este proceso no es el más óptimo para los usuarios finales, usuarios que no tienen ningún conocimiento del software libre y es muy incómodo tener que reiniciar la sesión por algo tan insignificante como cambiar el fondo de pantalla y el tema de ventana.

La aplicación de configuración está enfocada solamente en configurar el entorno de escritorio Enlightenment.

- **Xfce** es un entorno de escritorio ligero que consta de varias herramientas de configuración ya sea para configurar el fondo de pantalla, cambiar el tema de ventana o de iconos. Un ejemplo de estas herramientas es **xfce-setting-show** (Ver [Anexo 7. Herramientas de configuración de Xfce](#)), una potente herramienta de este entorno de escritorio, pero su principal problema es que está confeccionada sólo para Xfce y no para otros manejadores

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

de ventanas, por lo que tiene muchas dependencias y no está guide-dificada⁵. Por otro lado la herramienta que presenta para configurar el tema de ventanas y de iconos no cumple cabalmente toda con su funcionalidad.

- **KDE** es un entorno de escritorio pesado que presenta una gran variedad de herramientas de configuración (Ver [Anexo 8. Herramientas de configuración de KDE](#)). Éstas herramientas de configuración son parte de una serie de módulos que pertenecen al paquete **kdelibs**, el cual brinda la posibilidad de utilizarlos por separado utilizando el comando **kcmshell** o embebidos en cualquier otra aplicación que los necesite sin necesidad de reimplementar dichas funcionalidades.

A continuación se presentan las dependencias que acarrearán la instalación de estas aplicaciones.

Entorno de escritorio	Aplicaciones de configuración de escritorio	Dependencias
Gnome	gnome-windows-properties gnome-appearance-properties	gstreamer liboil eel libgnomekbd metacity gst-plugins-base libcanberra gnome-settings-daemon
KDE	background clock colors componentchooser desktopbehavior desktop ebrowsing filetypes fonts icons installtheme kcmcgi kmcscs	poppler-utils qt-core qt-dbus qt-sql qt-test qt-scripts qt-gui qt-qt3support qt-opengl qt-svg qt-webkit automoc strigi

⁵ Guide-dificada: Que cumple con los estándares dictados por Gnome en la “Human Interface Guideline” (HIG).

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

	kcmfontinst kcmkded kcmkicker kcmlaunch kcmperformance kcsmserver keys kfmclient kfontview khotkeys kpersonalizer krandrtray ksplashthememanager kthememanager media mouse netpref opengl kcontrol kdm keyboard keyboard_layout khtml_fonts kmenuedit knetattach kpager kwindecoration kwinoptions kwinrules language panel screensaver style xserver	soprano qimageblitz phonon kdebase-data
Xfce	xfce-setting-show	libxfce4utils extutils-dependends extutils-pkgconfig libxfce4mcs

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

		glib-perl xfconf libxfcegui4 xfce-mcs-manager
IceWm	icecc	qt3 icewm-tools icets icewmcp iceked iceiconcvt icecursorscfg iceme icesndcfg icewoed icemergeprefs

Tabla 3: Conjunto de dependencias de las herramientas de configuración de escritorio.

Como se puede observar la lista de dependencias de cada una de las aplicaciones de configuración antes mostradas resulta extensa, cada una de ellas por si sola acarrea una gran cantidad de dependencias, por lo que no es conveniente reutilizar cualquiera de ellas. En el caso de Enlightenment tampoco resulta conveniente reutilizar sus herramientas de configuración debido a que solamente configura dicho entorno.

Debido a la problemática antes planteada se hace necesario el desarrollo de las aplicaciones de configuración del entorno de escritorio Guano desde cero.

1.3.3. Administrador de sesión de usuarios.

¿Qué es una sesión?

Después de una búsqueda a fondo sobre que es una sesión se encontraron dos definiciones que se ajustan a la investigación:

Sesión: Es el marco espacio-temporal en el que se desarrolla una actividad concreta, en la cual se participa como actor o bien como espectador. Ejemplos pueden ser una sesión de cine, una sesión informática, una sesión de masaje, una sesión de estética, etc. (6)

Sesión: Espacio de tiempo comprendido entre el momento en que un usuario específico ingresa al sistema con su nombre de usuario y contraseña, y el momento en el que lo abandona. (7)

Por lo que se concluye que una sesión, no es más que el tiempo transcurrido en la que el usuario

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

inicia su sesión, donde el sistema brinda un conjunto de aplicaciones, configuraciones y recursos existentes en el escritorio del usuario; y el momento en que sale de esta.

La gestión de sesiones es un conjunto de convenios y protocolos que permiten al Gestor de sesiones guardar y restablecer la sesión de un usuario, para que cuando se vuelva a iniciar en el sistema tenga el mismo conjunto de aplicaciones en ejecución, configuraciones y recursos que los que disponía cuando finalizó la sesión anterior. Cuando un usuario inicia la sesión en el escritorio por vez primera, se carga una sesión inicial. Posteriormente, el Gestor de sesiones admite los conceptos de sesión actual y sesión de inicio (8).

El gestor de sesiones se inicia una vez que el nombre de usuario y la contraseña se han autenticado, este guarda y permite restaurar:

- Valores de aspecto y funcionamiento, por ejemplo, valores de tipos de letra, colores y del ratón.
- Aplicaciones que se estaban ejecutando como, por ejemplo, las ventanas del gestor de archivos y del editor de texto. No obstante, no se pueden guardar ni restaurar aplicaciones no gestionadas por el gestor de sesión. Por ejemplo, si inicia el editor Vim desde una línea de comandos en una ventana Terminal, el Gestor de sesiones no podrá restaurar la sesión de edición.

Los administradores de sesión más usados:

KSMServer: Es el administrador de sesiones de KDE, al iniciarse este se inician las aplicaciones auto-arrancables y se restauran las aplicaciones de la sesión anterior. Las aplicaciones auto-arrancables son las que se indican en los archivos .desktop en el directorio \$KDEDIR/share/autostart.

El administrador de sesiones puede ser configurado gráficamente mediante un módulo que posee el paquete **kdels** llamado **kcmsmserver** (Ver [Anexo 10. Aplicación de administración de sesión de KDE](#)), este puede ser ejecutado mediante el comando "**kmshell kcmsmserver**" desde una terminal.

Si se está ejecutando KDE por primera vez con un usuario concreto, se usará Kpersonalizer, software que ayuda con la primera configuración del escritorio. Si este no está presente, KDE iniciará, pero muchos valores predeterminados no se configurarán.

Algunas de las aplicaciones auto-arrancables por KSMServer por defecto son:

- Kdesktop: El escritorio de KDE.
- Kicker: El panel de KDE.
- Ktip: Un programa con el consejo del día.

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

- **Kwired:** Una utilidad para recibir mensajes del sistema enviados al usuario.
- **Klipper:** Una utilidad de portapapeles que se integra en el panel.
- **Kalarm:** Una utilidad que le avisa cuando se produce una cita.

Gnome-session: Es el administrador de sesión del entorno de escritorio GNOME. Al ejecutarse por primera vez se iniciará la sesión por defecto. Esto incluye el gestor de ventanas especificado (o el más apropiado si no se especificó), el panel, GNOME Midnight Commander, y el visor de ayudas. A medida que se use, el manejador de sesiones guardará el escritorio actual.

GNOME también presenta una aplicación de configuración de la sesión de usuario llamada **gnome-session-properties** contenida en el paquete **gnome-system-tools** (Ver [Anexo 9. Aplicación de administración de sesión de GNOME](#)).

En la tabla a continuación se hace referencia a las dependencias de cada una de las aplicaciones de administración de sesión antes mencionadas.

Entorno	Administrador de sesión	Dependencias
GNOME	gnome-session-properties	libnotify libgnomeui dbus-glib libXau xdpyinfo xrdp gnome-settings-daemon gnome-keyring
KDE	kcmsmsserver	kdelibs kdnssd-avahi qt3

Tabla 4. Dependencias que acarrearán la instalación de los administradores de sesión.

Debido a la cantidad de dependencias que presentan estas aplicaciones que provocan que se agregue software adicional al ya utilizado en la construcción de Guano, se considera viable la creación de un administrador de sesión reutilizando los componentes ya presentes en Guano.

1.1.1. Complementos o plugins.

Un complemento (o plug-in en inglés) es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de una API⁶.

Actualmente existe como una forma de expandir programas de forma modular, de manera que se puedan añadir sin afectar a las ya existentes ni complicar el desarrollo del programa principal.

Plugins del panel de Guano

El panel de Guano está basado en el Xfce4-panel del proyecto XFce, esto conlleva la incorporación de una modesta cantidad de plugins desarrollados por el proyecto del mismo nombre, que le brindan una serie de funcionalidades añadidas al panel. Muchos de estos plugins al ser desarrollados para el entorno de escritorio Xfce presentan problemas de incompatibilidad con su funcionamiento en Guano.

Por lo que se hace necesaria la modificación de algunos para su reutilización en el entorno de escritorio Guano.

Después del estudio realizado sobre las funcionalidades requeridas por los usuarios finales, se llegó a la conclusión que ningunas de las aplicaciones antes vistas corresponden una solución factible para Guano ya sea por las dependencias que acarrear, su alto consumo de memoria RAM⁷, no son muy intuitivas, ni fáciles de usar o no cumplen con los estándares de la HIG para la construcción de interfaces de usuarios.

Para desarrollar una solución informática que cumpla con las funcionalidades requeridas por los usuarios en la encuesta antes mencionada se hace necesario el estudio de las principales tecnologías, lenguajes así como herramientas que se utilizan en la confección de herramientas homólogas a las mismas en el mundo del software libre.

1.2. Lenguajes de programación y bibliotecas.

Para el desarrollo de cualquier software informático, seleccionar el lenguaje de programación es una

⁶ Una interfaz de programación de aplicaciones o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

⁷ Memoria de acceso aleatorio, (en inglés: Random Access Memory cuyo acrónimo es RAM) es la memoria desde donde el procesador recibe las instrucciones y guarda los resultados.

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

tarea primordial así como los componentes o librerías a utilizar, ya que de estos dependen la rapidez, la consistencia y la curva de desarrollo de la aplicación.

Al ser Guano un entorno de escritorio cuya construcción no parte de cero sino un compendio de componentes más un número de funcionalidades añadidas, las herramientas que se creen para la creación del mismo deben ajustarse al uso de los lenguajes y librerías propuestos por los primeros componentes añadidos al proyecto, reduciendo el estudio de los lenguajes así como las bibliotecas que se pueden utilizar para la construcción de la solución a los que a continuación se citan.

❖ Lenguaje C.

C es un lenguaje de programación creado en 1969 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B, a su vez basado en BCPL⁸. Es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones. Se trata de un lenguaje fuertemente tipado⁹ de medio nivel pero con muchas características de bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos. Proporciona facilidades para realizar programas modulares y/o utilizar código o bibliotecas existentes. (9)

Entre las características de C las más importantes son:

- Un núcleo del lenguaje simple, con funcionalidades añadidas importantes, como funciones matemáticas y de manejo de ficheros, proporcionadas por bibliotecas.
- Es un lenguaje muy flexible que permite programar con múltiples estilos. Uno de los más empleados es el estructurado no llevado al extremo (permitiendo ciertas licencias rupturistas).
- Un sistema de tipos que impide operaciones sin sentido.
- Acceso a memoria de bajo nivel mediante el uso de punteros.
- Un conjunto reducido de palabras claves.
- Los parámetros se pasan por valor. El paso por referencia se puede simular pasando explícitamente el valor de los punteros.
- Punteros a funciones y variables estáticas, que permiten una forma rudimentaria de encapsulado y polimorfismo.

⁸ BCPL (Lenguaje de Programación Básico Combinado): Diseñado en 1966 debido a las dificultades experimentadas con el lenguaje de programación CPL durante los años 60.

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

- Tipos de datos agregados (struct) que permiten que datos relacionados se combinen y se manipulen como un todo.

❖ UML

La Tecnología de Orientación a Objetos constituye la base de la reutilización de código por medio de componentes. UML (Lenguaje de Modelación Unificado o Unified Modeling Language, en inglés) es el lenguaje estándar adoptado por el OMG (Object Management Group) y mundialmente aceptado para la descripción de los "planos" de software. UML es un lenguaje gráfico para visualizar, especificar, construir y documentar cada una de las partes que comprende el desarrollo de software. UML posee formas de modelar conceptos como lo son procesos de negocio y funciones de sistema, además de aspectos concretos como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. (10)

▪ Diagramas UML

Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones. En concreto, un diagrama ofrece una vista del sistema a modelar. Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas. A continuación se citan algunos de ellos:

- Diagrama de clases: Muestra una colección de elementos de modelado declarativo (estático), tales como clases, tipos, sus contenidos y relaciones.
- Diagrama de Componentes: Representa los componentes que componen una aplicación, sistema o empresa. Los componentes, sus relaciones, interacciones y sus interfaces públicas.
- Diagrama de Objeto: Un diagrama que presenta a los objetos y sus relaciones en un punto de tiempo.
- Diagrama de paquete: Un diagrama que representa cómo se organizan los elementos de modelado en paquetes y sus dependencias entre ellos.

❖ Gtk+

GTK+ o The GIMP Toolkit (conjunto de rutinas para GIMP) es un grupo importante de bibliotecas o rutinas para desarrollar interfaces gráficas de usuario (GUI) para los entornos gráficos, principalmente GNOME, XFCE y ROX de sistemas UNIX. Inicialmente fue creado para desarrollar el programa de manejo de imágenes GIMP, sin embargo en la actualidad es muy usada por muchos otros programas

⁹ Un lenguaje de programación es fuertemente tipado si no se permiten violaciones de los tipos de datos

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

en los sistemas GNU/Linux. Este está escrito en C, pero puede ser usado desde otros lenguajes de programación como C++, C#, Java, Perl, PHP o Python (11).

A continuación se presentan algunas características:

- **Estabilidad**

Brinda un excelente rendimiento para el desarrollo de aplicaciones. Además esta soportado por una amplia comunidad de desarrolladores y los mantenedores principales de las compañías como son Red Hat, Novell, Imendio and Opened Hand.

- **Múltiple plataforma.**

Originalmente fue desarrollado para el Sistema de Ventanas X, pero ha crecido a lo largo de los años con el fin de incluir soporte a otros sistemas de ventanas. Actualmente es usado en GNU/Linux y Unix, Windows, Mac OS X.

- **Acomodando**

Abastece una serie de características que hoy en día los desarrolladores buscan:

- Apariencia agradable.
- Soporte de temas.
- Soporte para multi-hilos.
- Enfoque Orientado a Objeto.
- Internacionalización.
- Localización.
- Accesibilidad.
- Soporte bidireccional de texto.
- Soporte UTF8¹⁰.
- Documentación.

- **Interfaces**

Tiene una amplia colección de componentes básicos que incluyen:

- Ventanas (ventanas normales, de diálogos, asistentes y de información).

¹⁰ UTF8 (8-bit Unicode Transformation Format): Es una norma de transmisión de longitud variable para caracteres codificados utilizando Unicode.

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

- Componentes visuales (etiquetas, imágenes, barras de progreso, barras de estado).
- Botones con múltiples funcionalidades.
- Textos de entrada numéricos (con o sin completamiento).
- Editor de texto multi-líneas.
- Árboles, listas y vistas cuadriculadas de iconos.
- Cajas de diálogos.
- Menús (con imágenes y botones de elección).
- Barras de herramientas.
- Manejadores de interfaces de usuarios (UI).
- Selectores (selectores de colores, archivos y fuentes).
- Iconos de estado (área de notificación en Linux).
- Soportes de impresión.

❖ Glib

Es una biblioteca de propósito general que se usa para implementar muchas funciones no gráficas. Uno de los mayores beneficios de usar Glib es que provee una interfaz de plataforma independiente que permite que el código pueda ser usado en muchas plataformas como UNIX, Windows, OS/2 y BeOS. Otro aspecto de este es la amplia gama de tipo de datos que deja disponible al desarrollador:

- Incluye la gestión de estructuras de datos como listas enlazadas, arreglos dinámicos, tablas de búsqueda, árboles binarios, o la gestión de Entrada/Salida asíncrona. Provee estructuras y funciones para el manejo de listas simple y doblemente enlazadas
- Provee independencia con el sistema operativo, mejorando la portabilidad.
- Los segmentos de memoria son una forma eficiente de crear secciones de memoria que tienen todos los mismos tamaños. Pueden ser usados para crear arreglos aún de elementos de tamaño conocido.
- Manejo de plugins.
- Manejo dinámico de memoria.
- Soporte para multi-hilos (threads).

Además de proporcionar varios tipos de datos, también dispone de numerosos tipos de funciones.

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

Entre estas se encuentran funciones de manipulación de archivos, soporte de internacionalización, cadenas de caracteres, advertencias, banderas de depuración, carga dinámica de módulos, entre otras (12).

❖ GDK

Es una biblioteca de gráficos que actúa como un intermediario entre gráficos de bajo nivel y gráficos de alto nivel. Las siglas GDK corresponden a GIMP Drawing Kit, este originalmente se desarrolló para el Sistema de Ventanas X, este actúa entre el servidor X y la biblioteca GTK+, manejando las prestaciones básicas como primitivas de gráficos, mapas de bits, cursores, fuentes, eventos de ventanas y funcionalidades de arrastrar y soltar.

GDK es una parte importante de la portabilidad de GTK+. Como GLib provee funcionalidades independientemente de la arquitectura a las demás bibliotecas, todo lo que se necesita para hacer que GTK+ corra en cualquier plataforma es definir la capa de gráficos de un determinado sistema operativo en GDK (13).

Independientemente de los lenguajes de programación y las librerías a utilizar, escoger las herramientas de desarrollo es otro punto a tener en cuenta debido a que estas influyen en la curva de aprendizaje¹¹ y el tiempo de desarrollo de la aplicación de manera significativa.

1.3. Herramientas de desarrollo.

Para realizar la elección de las herramientas a utilizar en el desarrollo de la solución, se tuvieron en cuenta los siguientes criterios:

- La curva de aprendizaje de las herramientas debe ser rápida.
- Completamiento de código.
- No adicionar código innecesario.
- El consumo de recursos sea el mínimo necesario, sin comprometer la comodidad del usuario.

Por lo que se eligieron las siguientes herramientas de desarrollo:

❖ Glade.

Es una herramienta de desarrollo visual de interfaces gráficas mediante GTK/GNOME. Esta es independiente del lenguaje de programación y predeterminadamente no genera código fuente sino un

¹¹ Curva de aprendizaje: Describe el grado de éxito obtenido durante el aprendizaje en el transcurso del tiempo.

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

archivo XML, a partir del cual los programas pueden hacer uso de estos separando la interfaz de su funcionalidad (14).

❖ Geany.

Geany es un pequeño IDE para GNU/Linux basado en GTK2 y con muy pocas dependencias. Es un editor de gráficos muy estable, sencillo y rápido. Tiene resalte de sintaxis, autocompletado, atajos de teclado, muestra información de las funciones, tabulación automática, anidado de funciones según la apertura y cierre de paréntesis que se correspondan.

Las funciones son agrupadas en un árbol ordenado alfabéticamente, el que a la vez actúa como un acceso directo a las mismas. Dispone además de una consola de comandos mediante una terminal VTE integrada, con el intérprete que se quiera, bash¹² por defecto (15).

❖ DIA

El programa está basado en Windows Visio¹³ buscando para este mismo una alternativa de software libre, aunque más orientado hacia esquemas informales para uso ocasional. Se puede utilizar para dibujar diferentes tipos de diagramas. En la actualidad cuenta con objetos especiales para ayudar a crear diagramas entidad-relación, diagramas UML, organigramas, diagramas de red, y muchos otros esquemas. También es posible añadir soporte para nuevas formas de escritura de archivos XML, utilizando un subconjunto de SVG para dibujar la forma (16).

Para justificar la selección de esta herramienta se uso como marco de apoyo la comparación sobre Las Herramientas para UML en Linux (17) , donde se puede apreciar que con esta herramienta de modelado se pueden crear casi todos los diagramas de UML.

¹² Bash es un shell de Unix (intérprete de órdenes de Unix) escrito para el proyecto GNU.

¹³ Windows Visio: Programa del SO Windows que permite realizar dibujos, diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas y más.

	Umbrello	DIA	BoUML	ArgoUML
De clases	✓	✓	✓	✓
De componentes	✓	✓	✓	✗
De objetos	✗	✗	✗	✗
De estructura compuesta	✗	✗	✗	✗
De despliegue	✓	✓	✓	✓
De paquetes	✗	✓	✓	✗
De actividad	✓	✓	✗	✓
De casos de uso	✓	✓	✓	✓
De estado	✓	✓	✗	✓
De secuencia	✓	✓	✓	✓
De comunicación/colaboración	✓	✓	✓	✓
De tiempo	✗	✗	✗	✗
De vista de interacción	✗	✗	✗	✗

Figura 6. Comparación de las herramientas para UML en Linux.

Luego de haber caracterizado algunas de las herramientas de desarrollo puede pasarse a tratar las metodologías de desarrollo de software.

1.4. Las Metodologías de Desarrollo de Software.

Las metodologías de desarrollo de software surgieron a raíz de la necesidad de controlar y documentar proyectos cada vez más complejos. La implantación de una metodología es necesaria si se quieren gestionar adecuadamente los proyectos, por lo que aplicar una metodología de desarrollo de software en una organización es una tarea difícil y compleja. Las características de cada proyecto exigen que la metodología sea configurable y flexible a sus necesidades.

Ventajas de tener una metodología (18).

Mejora de los procesos de desarrollo.

- Todas las personas del proyecto trabajan bajo un marco común.
- Estandarización de conceptos, actividades y nomenclaturas.
- Actividades de desarrollo apoyadas por procedimientos y guías.
- Resultados del desarrollo predecibles.
- Uso de herramientas de Ingeniería Software.
- Planificación de actividades en base a un conjunto de tareas definidas y a la experiencia en otros proyectos.

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

- Recopilación de mejores prácticas para proyectos futuros.

Mejora de los productos.

- Se asegura que los productos cumplen con los objetivos de calidad propuestos.
- Detección temprana de errores.
- Se garantiza la trazabilidad de los productos a lo largo del desarrollo.

Mejora de las relaciones con el cliente.

- El cliente percibe el orden en nuestros procesos.
- Facilita al cliente el seguimiento de la evolución del Proyecto.
- Se establecen mecanismos para asegurar que los productos desarrollados cumplen con las expectativas del cliente.

Después de haber evidenciado la importancia de las metodologías para el desarrollo del software y las ventajas que conllevan el uso de las mismas, el siguiente paso a seguir es caracterizar qué se entiende por Metodologías Ágiles ya que se quiere que el ciclo de desarrollo de la aplicación sea lo más rápido posible.

1.4.1. Metodología Ágil.

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término “ágil” aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas (19).

El Manifiesto Ágil

Según el Manifiesto se valora:

- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- Desarrollar software que funciona más que conseguir una buena documentación. La regla a

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental.

- La colaboración con el cliente más que la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

Los valores anteriores inspiran los doce principios del manifiesto (Ver [Anexo 11. Principios del Manifiesto Ágil](#)). Son características que diferencian un proceso ágil de uno tradicional. Los dos primeros principios son generales y resumen gran parte del espíritu ágil. El resto tienen que ver con el proceso a seguir y con el equipo de desarrollo, en cuanto metas a seguir y organización del mismo.

Entre los principales métodos ágiles tenemos el XP (eXtreme Programming), Scrum, entre otras. Después de tener un conocimiento previo de las metodologías ágiles el siguiente paso es exponer las características que identifican a las antes mencionadas.

1.4.1.1. Programación Extreme (eXtreme Programming).

Las raíces de la programación extrema o Extreme Programming (XP) yacen en la comunidad de Smalltalk¹⁴, y en particular la colaboración cercana de Kent Beck¹⁵ y Ward Cunningham¹⁶. Ambos refinaron sus prácticas en numerosos proyectos a principios de los 90, extendiendo sus ideas de un desarrollo de software adaptable y orientado a la gente.

XP constituye una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizada para proyectos a corto plazo y pequeños equipos. Consiste en una programación rápida (extrema), cuya particularidad es tener como parte del equipo al usuario final pues es uno de los requisitos para llegar al éxito del proyecto.

Las características fundamentales de la metodología son:

¹⁴ Smalltalk: Lenguaje de programación considerado el primero en utilizar el paradigma orientado a objetos (OOP). Tuvo gran influencia en la creación de otros lenguajes como Java o Ruby.

¹⁵ Kent Beck: Uno de los creadores de la metodología ágil para el desarrollo de software conocida como Programación Extrema. Fue el alma mater del Manifiesto Ágil y creó, junto con Erich Gamma el framework de pruebas unitarias para los lenguajes de programación Java, JUnit.

¹⁶ Ward Cunningham: Programador de patrones para programación y del WikiWikiWeb (Sitio Web dedicado a programación). Es coautor (con Bo Leuf) del libro The Wiki Way (2001). Ward es conocido por sus contribuciones en la práctica del desarrollo de programación orientada a objetos, la metodología de desarrollo de software XP y las comunidades alojadas en su WikiWikiWeb.

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

- Desarrollo iterativo e incremental.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto.
- Frecuente integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento.
- 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.
- Simplicidad en el código: Es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo en caso necesario, que realizar algo complicado y quizás nunca utilizarlo.

Ventajas

- Apropiado para entornos volátiles.
- Estar preparados para el cambio, significa reducir su coste.
- Planificación más transparente para nuestros clientes, conocen las fechas de entrega de funcionalidades. Vital para su negocio.
- Permitirá definir en cada iteración cuales son los objetivos de la siguiente.
- Permite tener realimentación de los usuarios muy útil.
- La presión esta a lo largo de todo el proyecto y no en una entrega final

Desventajas

- Delimitar el alcance del proyecto con nuestro cliente.

Para mitigar esta desventaja se plantea definir un alcance a alto nivel basado en la experiencia.

Fases de XP.

Fase de la exploración: En esta fase, los clientes plantean los requisitos del usuario para la primera

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se define un prototipo del sistema. Esta fase toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

Fase del planeamiento: Se priorizan los requisitos de los usuario y se acuerda el alcance de las liberaciones del producto. Los programadores estiman cuánto esfuerzo requiere cada requisito y a partir de allí se define el cronograma. El cronograma fijado en la etapa de planeamiento se realiza a un número de iteraciones, cada una toma de una a cuatro semanas en ejecución. La primera iteración crea un sistema con la arquitectura del sistema completo. Las pruebas funcionales creadas por el cliente se ejecutan al final de cada iteración. Al final de la última iteración el sistema está listo para producción.

Fase de producción: Requiere de pruebas prueba y comprobación extra del funcionamiento del sistema antes de que éste se pueda liberar al cliente. En esta fase, los nuevos cambios pueden todavía ser encontrados y debe tomarse la decisión de si se incluyen o no en la liberación actual del producto. Las ideas y las sugerencias pospuestas se documentan para una puesta en práctica posterior por ejemplo en la fase de mantenimiento.

Fase de mantenimiento: Esta fase requiere de un mayor esfuerzo para satisfacer también las tareas del cliente. Así, la velocidad del desarrollo puede desacelerar después de que el sistema esté en la producción. La fase de mantenimiento puede requerir la incorporación de nueva gente y cambiar la estructura del equipo.

Fase de muerte: Es cuando el cliente no tiene más requisitos para ser incluidos en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

1.4.1.2. Scrum.

Es una forma de gestionar proyectos de software. No es una metodología de análisis, ni de diseño, como podría ser RUP (unido a UML), es una metodología de gestión del trabajo. Una de las características más importantes es que es muy fácil de explicar y de entender, lo que ayuda mucho a su implantación.

El desarrollo se realiza en forma iterativa e incremental, donde cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nueva funcionalidad. Las iteraciones en general tienen

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

una duración entre 2 y 4 semanas. Scrum se utiliza como marco para otras prácticas de ingeniería de software como RUP o XP.

Este se focaliza en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. Está diseñado especialmente para adaptarse a los cambios en los requerimientos. Al principio del proyecto se define el Product Backlog, que contiene todos los requisitos funcionales y no funcionales que deberá satisfacer el sistema a construir. Los mismos estarán especificados de acuerdo a las convenciones de la organización ya sea mediante: características, casos de uso, diagramas de flujo de datos, incidentes, tareas y otros.

Como método, Scrum 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.

Debido a que XP está dirigida solamente al desarrollo de la aplicación, se hace necesario el uso de Scrum que más que una metodología de desarrollo software, es una forma de auto-gestión del proyecto sin caer en tecnicismos; obteniéndose de la unión de ambas una meta-metodología con las características que se necesitan para esta empresa.

En la Universidad de las Ciencias Informática, se realizó una investigación que arrojó como resultado la propuesta SXP (20) , una nueva metodología de desarrollo de software que toma los mejores aspectos de XP y Scrum, adaptándolo a las características de los proyectos productivos de la facultad

10.

1.4.1.3. SXP o Metodología ágil para proyectos de software libre.

Es una nueva estructura para el expediente de proyecto, que responde a las características de la metodología utilizada actualmente en los proyectos productivos, del Polo de Software Libre de la Facultad 10. Esta metodología no es más que la unión de XP y Scrum, para el logro de un buen desarrollo de software, propuesta en el 2007 por la Ing. Malay Rodríguez Villar, y probada en los proyectos que trabajan con Software Libre, obteniéndose buenos resultados.

La metodología está dividida en cuatro fases, que son precisamente la base de la estructura del nuevo expediente de proyecto, estas son:

- Planificación-Definición.
- Desarrollo.
- Entrega.
- Mantenimiento.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, Diseño, Implementación, Pruebas, entre otras; de donde se generan artefactos que quedan incluidos en el nuevo expediente de proyecto.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. La metodología en un principio se llamaba MA-GMPUR2 pero como el nombre no era comercial se le dio el nuevo nombre de SXP.

Visto lo anterior, se puede concluir que se debe utilizar C como lenguaje principal en la construcción de las herramientas de configuración del escritorio Guano, debido a que la mayoría de las librerías utilizadas para desarrollar el entorno están escritas en este lenguaje, así como las aplicaciones que ya existen en el mismo. Concretando el hecho de no incorporar otras bibliotecas y la reutilización de las ya usadas (Gtk+, Gdk, Glib), haciéndose la consistencia del entorno más robusta además de no comprometer la rapidez del mismo.

Como herramienta de desarrollo o IDE¹⁷ se propone a Geany, debido a su ligereza pero consistencia como editor, utilizando su paquete de plugins que hacen que el programador haga mucho más

¹⁷ IDE: Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un

CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.

cómodo su trabajo sin hacer uso indiscriminado de los recursos de la computadora; combinándose con la herramienta Glade para la confección de interfaces de usuario basándose en los estándares de la GUI.

Resulta válida también la elección de SXP como metodología a emplear en la construcción del software debido a su sencillez y la cualidad de centrarse en la implementación, pruebas y la gestión del proyecto a desarrollarse sin caer en tecnicismos, sin dejar de lado que es definida como una política de desarrollo dentro del proyecto Nova; completando el proceso de ingeniería con el uso de DIA como herramienta de modelado basado en las características expuestas en el capítulo.

editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

2. ANÁLISIS Y DESARROLLO DE LA SOLUCIÓN PROPUESTA.

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

2.1. Valoración crítica de las herramientas de Guano.

Para lograr que los usuarios se sientan a gusto una vez que se realice la migración, se necesita que los aspectos de diseño de las herramientas así como la interacción con las mismas sean lo más semejantes posibles a los entorno de escritorio de más aceptación. Basándose en los resultados de la encuesta realizada a los usuarios, y de las funcionalidades que estos desean que se conserven, se hace una valoración de los componentes de Guano, para tener conocimiento de que podemos reutilizar.

El manejador de fichero PCmanFM contiene por defecto un módulo que permite configurar la apariencia del escritorio (Ver [Anexo 13. Herramienta de configuración de escritorio del PCmanFM](#)), pero no presenta todas las configuraciones que los usuarios de Windows están acostumbrados a realizar, como son la de escoger un tema de icono a usar, las fuentes tipográficas, etc. Uno de los aspectos que no amerita a este módulo es que cuando se va a cambiar el fondo de escritorio este no visualiza cual es la imagen que estamos usando, ni da la posibilidad de que los usuarios puedan ver cuál de las imágenes que tiene disponible es la que desea.

Otro de los componentes de Guano, que hay que estandarizar a los entornos de escritorios más solicitados, es el panel de Xfce, que en primer lugar no cuenta con un menú de las aplicaciones propio y aunque da la posibilidad de que se pueda importar el de GNOME sigue sin cumplir con el objetivo planteado ya que su instalación acarrea muchas dependencias que pueden sobrecargar al entorno de escritorio por lo que dejaría de ser ligero. También se presenta el inconveniente de que muchos de los lanzadores de aplicaciones no están bien configurados por lo que no se puede hacer uso de ellos. Un aspecto a destacar del panel de Xfce es que tiene un conjunto de plugins que se puede reutilizar para Guano como son:

- Xfce4-mixer: plugins que controla el volumen.
- Xfce4-xkb: plugins para el teclado.

CAPÍTULO 2. ANÁLISIS Y DESARROLLO ÁGIL DE LA SOLUCIÓN PROPUESTA

- Se utiliza el plugins del reloj que viene integrado al panel.

Es válido destacar la existencia de varios plugins que deben ser reutilizados y modificados dado el hecho de que han sido construidos y desarrollado para herramientas específicas que no comprenden la arquitectura de Guano, tales como:

- Plugin del panel para la papelera de reciclaje.
- Plugin del panel para gestionar la sesión de usuarios (bloquear y desbloquear).
- Plugin del menú de aplicaciones del panel.

2.2. Análisis y Desarrollo ágil de la Solución.

El resultado final de un proyecto de software es un producto que a lo largo del desarrollo del proyecto va tomando forma. En el mundo actual en el que se necesita obtener productos con alta calidad en un tiempo de elaboración mínimo, se impone mejorar el proceso de desarrollo de software, incrementando la productividad de los equipos involucrados en el desarrollo de los proyectos. Por tanto el objetivo primordial de los procesos de desarrollo de software es guiar a los desarrolladores para la obtención de sistemas que se ajusten a las necesidades de los clientes.

Partiendo de las características de la Metodología SXP que es la definida como marco de apoyo para documentar, analizar y desarrollar la solución propuesta se decide inicializar el análisis de la solución partiendo de la fase Planificación – Definición ya que esta define una serie de pasos a seguir que reflejan una vista general del producto a implementar partiendo de las necesidades del cliente.

2.2.1. Planificación y Definición del sistema.

La Plantilla de Concepción del Sistema es el primer documento generado en la fase Planificación – Definición, la realización de este es de gran importancia para la elaboración de la solución ya que después de la entrevista con los clientes del producto se obtiene una concepción inicial de la propuesta; también queda recogido los roles que intervendrán así como las responsabilidades que tendrán en dicho proceso, las herramientas que serán utilizadas para el desarrollo de la aplicación, entre otros aspectos (Más Información ver [Anexos 12. Plantilla Concepción del Sistema.](#))

Propuestas del sistema a realizar.

Después del estudio realizado sobre el estado actual de las herramientas que cumplen con las funcionalidades que los usuarios querían conservar en Guano una vez que se realizara la migración y

CAPÍTULO 2. ANÁLISIS Y DESARROLLO ÁGIL DE LA SOLUCIÓN PROPUESTA

las críticas a las herramientas que el mismo posee en estos momentos; se llego a la conclusión que ninguna de las expuestas en el Capítulo 1 son una solución factible para Guano y las que posee no siguen la línea que hasta estos momentos los usuarios requieren así como las políticas que el proyecto desea, por lo que se decide construir una solución de software que integre las funcionalidades requeridas por los clientes siguiendo la misma arquitectura de la información de las antes expuestas (Ver [Anexo 14. Prototipo de la Solución Propuesta](#)).

La Propuesta estará conformada por 3 aplicaciones, un menú de aplicaciones, una herramienta para configurar el escritorio y otra para manejar la sesión de los usuarios, además de plugins para el panel con el soporte de la papelera de reciclaje y otro para gestionar la sesión de usuarios (bloquear y desbloquear). El menú de las aplicaciones sigue las mismas pautas de los menús de los entornos de escritorios que más aceptación han tenido en el software libre; este brinda la posibilidad de poder acceder a las aplicaciones recientes así como mostrar todas las que están instaladas en el sistema por categorías y poder editar el menú para agregar, quitar o modificar una aplicación, además de insertarlo como plugin al panel. La herramienta encargada de configurar la apariencia del escritorio debe darle la posibilidad al usuario de que este pueda cambiar el fondo de pantalla, dando una vista previa de cómo quedaría la imagen; que se pueda configurar las fuentes tipográficas, el tema de ventana y de iconos. Por otra parte la herramienta que maneja la sesión de los usuarios debe permitir que los usuarios puedan decidir las aplicaciones que quieren que se auto-ejecuten una vez inicie su sesión. En cuanto a los plugin para el panel, estos no son más que una modificación a los ya existentes debido a que no se ajustan a la arquitectura de Guano. El plugin de la papelera de reciclaje debe darle la posibilidad a los usuarios que estos puedan vaciar la papelera desde el panel así como visualizar los archivos eliminados y el plugin para gestionar la sesión de los usuarios debe estar conectado al Guano-logout y permitir además que mediante las propiedades del mismo se pueda definir de su estado por defecto.

Después de tener una concepción inicial de la solución propuesta se hace más fácil comprender el negocio por lo que el siguiente paso a seguir es el modelado del negocio que permite tener una vista más clara de la interacción del sistema con los usuarios y facilita la captura de los requisitos ya que se tiene una noción del sistema y de lo que este va a realizar.

En este caso no se realiza un Modelo de Negocio debido a que en el mismo no están bien definidos los procesos a realizar, por lo que se realiza un Modelo de Dominio.

¿Qué es un Modelo de Dominio?

CAPÍTULO 2. ANÁLISIS Y DESARROLLO ÁGIL DE LA SOLUCIÓN PROPUESTA

El modelo del dominio se puede definir como un modelo conceptual que describe las principales entidades presentes y sus relaciones; usualmente es capaz de identificar los métodos y atributos de las entidades fundamentales, por lo que se puede decir que provee una primera vista estructural del sistema. Es creado para documentar los conceptos claves y el vocabulario que será utilizado por el equipo de desarrollo al referirse al sistema, y puede ser utilizado para garantizar o aumentar la comprensión del problema por los propios clientes por lo que es muy útil como herramienta de comunicación en caso de que el proyecto implemente una división en equipos técnicos y de negocio (21).

A partir del concepto antes mencionado, se obtuvo la realización del modelo de dominio para dicha solución, el cual se muestra a continuación:

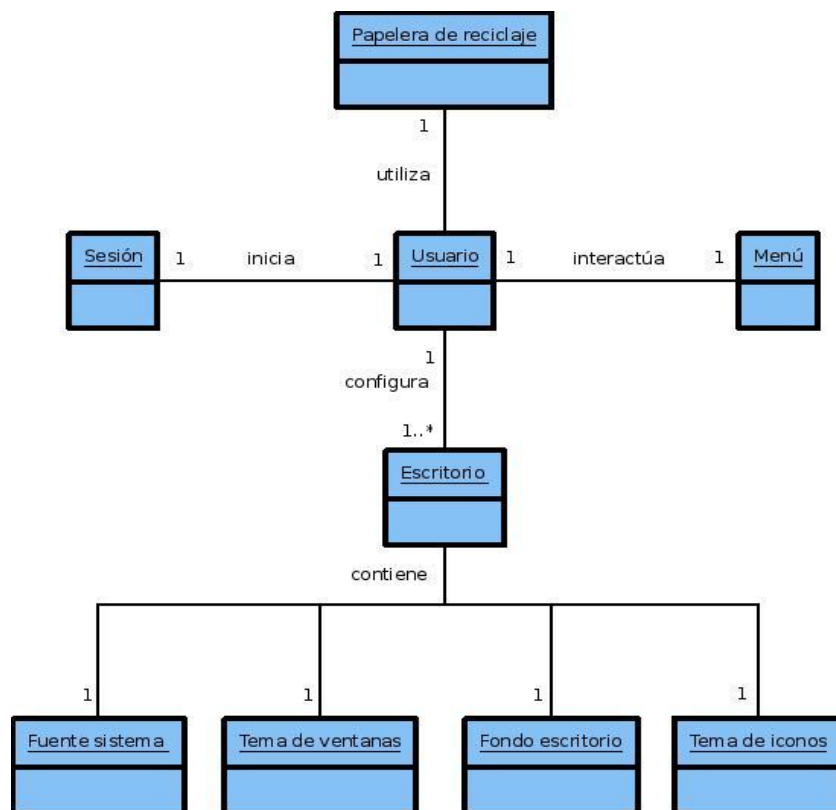


Figura 7: Modelo de Dominio.

El siguiente paso a seguir después de tener una primera vista conceptual de cómo quedaría el sistema a implementar, es la recolección de los requisitos que debe tener la solución propuesta; para esto se parte de la entrevista realizada con los clientes y recogida en la Lista de Reserva del producto. .

CAPÍTULO 2. ANÁLISIS Y DESARROLLO ÁGIL DE LA SOLUCIÓN PROPUESTA

2.3. Lista de reserva del producto (LRP).

La LRP es el primer artefacto generado en la etapa de Captura de requisitos, está conformada por una lista priorizada que define el trabajo que se va a realizar en el proyecto.

Asignado a	Item *	Descripción	Estimación
		Prioridad	Muy Alta
Miguel y Barbarita.	1	Tener una aplicación que permita modificar las configuraciones del escritorio y las fuentes tipográficas, iconos y tema de Gtk.	2 semanas.
		Prioridad	Alta
Miguel y Barbarita.	2	Tener un menú de aplicaciones, que muestre las aplicaciones instaladas en el sistema.	1 semana.
Miguel y Barbarita	3	Tener un menú de aplicación que recuerde las aplicaciones recientes.	1 semana.
Miguel y Barbarita	4	Poder editar el menú (para agregar aplicaciones al menú, quitar aplicaciones al menú, modificar una aplicación al menú).	1 semana
		Prioridad	Media
Miguel y Barbarita.	5	Agregar soporte para la papelerera de reciclaje que permita vaciar la papelerera así como acceder a ella y poder gestionar la sesión de los usuarios (ya sea para bloquear o salir) desde el panel.	2 semanas.
Miguel y Barbarita.	6	Tener una aplicación para el manejador de sesión que permita agregar y quitar una aplicación al inicio de sesión.	2 semanas.
RFN (Requisitos no funcionales)			
Miguel y Barbarita.	7	La solución debe ser óptima para cualquier hardware, preferentemente aquellos que se consideren hardware obsoletos.	
Miguel y Barbarita.	8	Las aplicaciones deben poseer una interfaz gráfica amigable y simple para los usuarios finales.	
Miguel y Barbarita.	9	Las aplicaciones deben funcionar bien sobre computadoras con al menos 800 MHz de velocidad de procesamiento y 256 MB de RAM	
Miguel y Barbarita.	10	El funcionamiento de las aplicaciones debe ser rápido, y el tiempo de respuesta no deberá exceder los 10 segundos.	

Tabla 5: Lista de reserva del producto.

Después de tener recogido los requerimientos de las aplicaciones a desarrollar, estos se desglosan en historias de usuarios que no son más que una abstracción de los requerimientos escritos en lenguaje informal llevados a un lenguaje donde el equipo de desarrollo pueda entender.

2.4. Las historias de usuarios y los prototipos de interfaz.

¿Qué son las historias de usuario?

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software, lo que equivaldría a los casos de uso en el proceso unificado. Las mismas son escritas por los clientes como las tareas que el sistema debe hacer y su construcción depende principalmente de la habilidad que tenga el cliente para definir las. Son escritas en lenguaje natural, no excediendo su tamaño de unas pocas líneas de texto. Estas guían la construcción de las pruebas de aceptación, elemento clave en XP y son utilizadas para estimar tiempos de desarrollo (22).

A continuación se darán a conocer las distintas historias de usuarios que están presente es en el sistema; así como su relación con los prototipos de interfaz usuario. Ésta es solo una planificación inicial, el proceso es cambiante para ir adecuándolo a las necesidades y nuevas propuestas.

Historia de Usuario	
Fecha:	Versión: 1.0
Número: U_SD_0.1	Dependencia: No
Nombre de historia: Configurar el escritorio.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Alto.
Descripción: La aplicación se encargará de modificar todo lo referente a configurar el entorno de escritorio. Se procederá a cambiar el fondo de escritorio, estilo de fuentes del sistema, color de la fuente, tema de iconos y temas de Gtk.	
Observaciones: Para que este proceso se lleve a cabo es necesario que se encuentre activa y abierta una sesión de usuario.	
Prototipo de interfaz de usuario.	



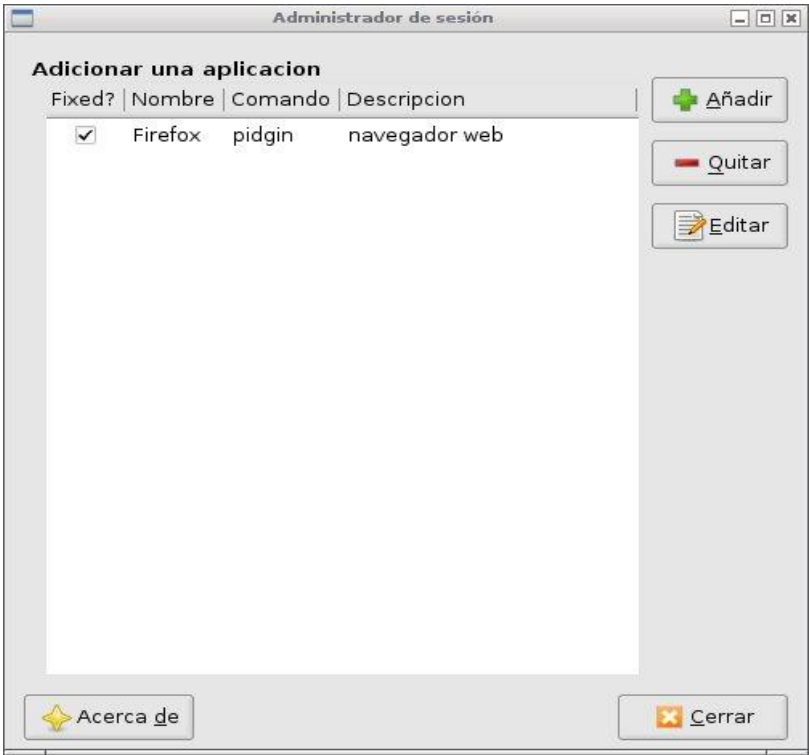
Historia de Usuario	
Fecha:	Versión: 1.0
Número: U_SD_0.2	Dependencia: No
Nombre de historia: Visualizar menú de aplicaciones.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Alto.
Descripción: Se muestra en el panel un botón de acceso al menú de aplicaciones de la sesión de usuario, con cada una de las aplicaciones instaladas.	
Observaciones: Debe estar iniciada y abierta una sesión de usuario, además de estar seleccionado el panel (conocido como barra de tareas por los usuarios de Windows).	
Prototipo de interfaz de usuario.	




Historia de Usuario	
Fecha:	Versión: 1.0
Número: U_SD_0.3	Dependencia: No.
Nombre de historia: Incluir plugin para el soporte de la papelera de reciclaje en el panel.	
Prioridad en negocio: Media.	Riesgo en desarrollo: Medio.
Descripción: Se procederá a acceder a la papelera desde el panel, así como vaciar la misma de archivos innecesarios para el usuario.	
Observaciones: Para que este proceso se lleve a cabo es necesario que se encuentre activa y abierta una sesión de usuario.	
Prototipo de interfaz de usuario.	
<p>The image shows a Linux desktop environment with a panel. On the left, there are icons for "NOVA" and "GIMP". On the right, there are system icons including a trash can, a power button, and the time "4:44 PM".</p>	

Historia de Usuario

CAPÍTULO 2. ANÁLISIS Y DESARROLLO ÁGIL DE LA SOLUCIÓN PROPUESTA

Fecha:	Versión: 1.0
Número: U_SD_0.4	Dependencia: No.
Nombre de historia: Gestionar el inicio de la sesión de usuario.	
Prioridad en negocio: Media.	Riesgo en desarrollo: Bajo.
Descripción: Se manejará el inicio de sesión de los usuarios: añadir aplicaciones al inicio de sesión, remover aplicaciones, así como modificar las características de las aplicaciones añadidas.	
Observaciones: Para que este proceso se lleve a cabo es necesario que se encuentre activa y abierta una sesión de usuario.	
Prototipo de interfaz de usuario.	
	

Historia de Usuario	
Fecha:	Versión: 1.0
Número: U_SD_0.5	Dependencia: No.

Nombre de historia: Incluir plugin para gestionar la sesión de los usuarios en el panel	
Prioridad en negocio: Media.	Riesgo en desarrollo: Medio.
Descripción: Se procederá a acceder al Guano-logout desde el panel así como poder definir su estado por defecto mediante las propiedades.	
Observaciones: Para que este proceso se lleve a cabo es necesario que se encuentre activa y abierta una sesión de usuario.	
Prototipo de interfaz de usuario.	
	

En el ciclo de desarrollo de cualquier software siempre hay situaciones que poner en peligro el desarrollo del sistema, para contrarrestar esto hay que tener una buena gestión de los riesgos que se pueden presentar. Esta tiene como objetivo marcar las estrategias y formas de actuar del equipo de trabajo frente a los riesgos: cómo evitarlos, monitorizarlos, gestionarlos y actuar ante contingencia, estableciendo las tareas principales a realizar en caso de que se llevaran a cabo.

Por tanto después de tener definida las historias de usuarios del sistema la siguiente acción es definir la lista de riesgo que pueden atrasar el desarrollo de la solución así como las tareas para mitigarlo.

2.5. Lista de riesgos

Riesgo	Tipo de Riesgo	Impacto	Descripción	Probabilidad	Efectos	Mitigación del riesgo
--------	----------------	---------	-------------	--------------	---------	-----------------------

CAPÍTULO 2. ANÁLISIS Y DESARROLLO ÁGIL DE LA SOLUCIÓN PROPUESTA

1	Personal	Retraso en la entrega del producto.	Ausencia de un programador, analista, documentador o un diseñador.	Media	Serios	El riesgo es resarcido planificando bien las tareas de cada roll sin crear dependencia extrema de un trabajo con otro.
2	Tecnológico	Retraso del trabajo incidiendo negativamente en la terminación del producto.	Rotura de computadora donde se encuentra información importante del proyecto.	Alta	Serios	Se evita revisando diariamente las computadoras para ver si todo se encuentra en condiciones óptimas para la realización del trabajo, así como realizar salvadas de la información importante.
3	Tecnológico	Desagrado por parte del equipo, amplía el período de culminación.	Afectación del fluido eléctrico en el laboratorio de producción.	Baja	Tolerable	Trabajar el mayor tiempo posible durante la noche ya que hay menos consumo de electricidad durante la noche y la red no se sobrecarga.
5	Personal	Retraso en las actividades que desarrolla esa persona dentro del proyecto.	Enfermedad de uno de los miembros del equipo.	Alta	Tolerable	Tener preparado a más de una persona que sepa realizar las estas actividades para que el trabajo del equipo no se atrase debido a estas consecuencias.
6	Personal	Retraso del trabajo	Falta de Experiencia	Media	Tolerable	Obtener entrenamiento y asesoría de personas expertas y negociar compromisos adecuados

Tabla 6. Lista de riesgos

Luego de tener definidas las historias de usuario a implementar así como los riesgos que pueden surgir durante en el ciclo de desarrollo de la solución, el siguiente paso es hacer una buena planificación de lo que se va a implementar para estimar el tiempo que se llevara el desarrollo de

CAPÍTULO 2. ANÁLISIS Y DESARROLLO ÁGIL DE LA SOLUCIÓN PROPUESTA

estas aplicaciones y de esta forma evitar sorpresa.

2.6. Planificación de las historias de Usuarios.

Planificar acertadamente el proyecto es necesario para estimar de forma ideal el tiempo que les tomará a los programadores la codificación de cada una de las historias de usuario. A partir de la prioridad de las historias se decide cuales de ellas se implementarán en las primeras iteraciones, las funcionalidades críticas del sistema deben ser codificadas en iteraciones tempranas del ciclo. Esta planificación queda recogida en la plantilla Plan de Liberación que es una de los artefactos que genera la metodología SXP.

2.6.1. Plan de Liberación.

Reléase	Descripción de la iteración	Orden de la HU a implementar	Duración total
Iteración 1	En esta iteración se desarrolla la aplicación que será la encargada de cambiar las configuraciones del escritorio según las preferencias del cliente.	U_SD_0.1	
Iteración 2	En esta iteración se desarrolla una aplicación que será capaz de mostrar todas las aplicaciones instaladas en el sistema así como dar la posibilidad de poder acceder de manera rápida a las recientemente ejecutadas, entre otras funcionalidades que permitan una mejor interacción del usuario con el sistema	U_SD_0.2	
Iteración 3	En esta iteración se desarrolla las condiciones necesarias para brindar soporte de papelera de reciclaje en panel, así como gestionar la sesión de los usuarios (bloquear o salir); también se desarrolla la aplicación que maneja la sesión de los usuarios para poder gestionar las aplicaciones que el usuario desea que se autoarranque una vez que se inicie la sesión.	U_SD_0.3 U_SD_0.4	

Tabla 7: Plan de Liberación.

Después de tener planificado cuales son las historias de usuarios que se van a implementar en cada iteración, lo que queda es determinar las tareas de ingeniería ya que estas permiten definir cada una de las actividades que estén asociadas a las historias de usuario y su implementación.

CAPÍTULO 2. ANÁLISIS Y DESARROLLO ÁGIL DE LA SOLUCIÓN PROPUESTA

2.7. Tareas de ingeniería.

Iteración 1

En esta iteración se expondrán las tareas de ingeniería que dan solución a la Historia de usuario U_SD_0.1 “Configurar el escritorio”.

Tarea de Ingeniería	
Numero Tarea: 1	Nombre de la Historia de Usuario: U_SD_0.1
Nombre de la Tarea: Estudiar las aplicaciones de configuración del escritorio.	
Tipo de Tarea: Estudio	Puntos Estimados: 1 semana
Fecha de Inicio: 12 / 01/ 2009	Fecha Fin: 17 / 01/ 2009
Programador Responsable: Miguel y Barbarita	
Descripción: Estudio de las aplicaciones de configuración de escritorio para saber cómo funciona y ver si alguna puede ser una solución factible para el entorno de escritorio Guano.	

Tarea de Ingeniería	
Numero Tarea: 2	Nombre de la Historia de Usuario: U_SD_0.1
Nombre de la Tarea: Desarrollar el Módulo de configuración del fondo de escritorio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2 semanas
Fecha de Inicio: 20 / 01/ 2009	Fecha Fin: 4 / 02/ 2009
Programador Responsable: Miguel	
Descripción: Se desarrollo el módulo que permite al usuario cambiar el fondo de pantalla así como dar una vista previa de la imagen que está usando. También debe dar a la posibilidad de que se pueda escoger en qué forma se quiere que se visualice ya sea centrado, mosaico, estirado o escalado; además de cambiar el color del fondo, de la letra y el color de la sombra del texto.	

Tarea de Ingeniería	
Numero Tarea: 3	Nombre de la Historia de Usuario: U_SD_0.1
Nombre de la Tarea: Desarrollar el Módulo configuración de temas de Gtk y fuentes tipográficas del sistema.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2 semanas

CAPÍTULO 2. ANÁLISIS Y DESARROLLO ÁGIL DE LA SOLUCIÓN PROPUESTA

Fecha de Inicio: 5/ 02/ 2009	Fecha Fin: 20 / 02/ 2009
Programador Responsable: Miguel	
Descripción: Se desarrollo el módulo que permite cambiar el tema de ventana de Gtk así como las fuentes tipográficas de las aplicaciones dando la posibilidad de que se pueda escoger el tipo de letra, estilo y tamaño.	

Tarea de Ingeniería	
Numero Tarea: 4	Nombre de la Historia de Usuario: U_SD_0.1
Nombre de la Tarea: Desarrollar el Módulo configuración de temas de iconos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2 semanas
Fecha de Inicio: 21/ 02/ 2009	Fecha Fin: 7 / 03/ 2009
Programador Responsable: Miguel	
Descripción: Se desarrollo el módulo que permite cambiar el tema de iconos.	

Iteración 2

En esta iteración se expondrán las tareas de ingeniería que dan solución a la Historia de usuario U_SD_0.2 “Visualizar menú de aplicaciones”.

Tarea de Ingeniería	
Numero Tarea: 5	Nombre de la Historia de Usuario: U_SD_0.2
Nombre de la Tarea: Estudio de los principales menús de aplicaciones.	
Tipo de Tarea: Estudio	Puntos Estimados: 1 semanas
Fecha de Inicio: 8/ 03/ 2009	Fecha Fin: 13 / 03/ 2009
Programador Responsable: Miguel y Barbarita	
Descripción: Estudio de los menú de los entorno de escritorios que más aceptación tienen.	

Tarea de Ingeniería	
Numero Tarea: 6	Nombre de la Historia de Usuario: U_SD_0.2
Nombre de la Tarea: Desarrollar generador de menú.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semanas

CAPÍTULO 2. ANÁLISIS Y DESARROLLO ÁGIL DE LA SOLUCIÓN PROPUESTA

Fecha de Inicio: 15/ 03/ 2009	Fecha Fin: 22 / 03/ 2009
Programador Responsable: Miguel	
Descripción: El generador del menú va hacer el encargado de visualizar las aplicaciones del sistema por categoría, así como adicionar automáticamente las nuevas que se instalen.	

Tarea de Ingeniería	
Numero Tarea: 7	Nombre de la Historia de Usuario: U_SD_0.2
Nombre de la Tarea: Desarrollar menú de aplicaciones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semanas
Fecha de Inicio: 8/ 03/ 2009	Fecha Fin: 13 / 03/ 2009
Programador Responsable: Miguel	
Descripción: Se desarrolla el menú de aplicaciones que debe mostrar las aplicaciones recientemente abiertas, así como los accesos a para salir de la sesión o bloquearla.	

Iteración 3

En esta iteración se expondrán las tareas de ingeniería que dan solución a las Historia de usuario U_SD_0.3 “Incluir plugin para el soporte de la papelera de reciclaje en el panel”, la U_SD_0.4 “Gestionar el inicio de la sesión de usuario” y la U_SD_0.3 “Incluir plugin para la gestión de la sesión de los usuarios desde el panel”

Tarea de Ingeniería	
Numero Tarea: 8	Nombre de la Historia de Usuario: U_SD_0.3
Nombre de la Tarea: Estudio del funcionamiento de los plugins del panel.	
Tipo de Tarea: Estudio	Puntos Estimados: 1 semanas
Fecha de Inicio: 13/ 03/ 2009	Fecha Fin: 17 / 03/ 2009
Programador Responsable: Barbarita y Miguel	
Descripción: Se estudia las principales plugins existentes en este caso el de la papelera y el de gestionar la sesión de los usuarios (bloquear o salir de la sesión)	

CAPÍTULO 2. ANÁLISIS Y DESARROLLO ÁGIL DE LA SOLUCIÓN PROPUESTA

Tarea de Ingeniería	
Numero Tarea: 9	Nombre de la Historia de Usuario: U_SD_0.3
Nombre de la Tarea: Desarrollo del plugin de la papelera de reciclaje en el panel.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semanas
Fecha de Inicio: 17/ 03/ 2009	Fecha Fin: 24 / 03/ 2009
Programador Responsable: Barbarita	
Descripción: Se desarrolla el plugin de la papelera de reciclaje en el panel con el soporte especificado.	

Tarea de Ingeniería	
Numero Tarea: 10	Nombre de la Historia de Usuario: U_SD_0.5
Nombre de la Tarea: Desarrollar Plugin para la gestión de la sesión de los usuarios desde el panel	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semanas
Fecha de Inicio: 17/ 03/ 2009	Fecha Fin: 24 / 03/ 2009
Programador Responsable: Barbarita	
Descripción: Se desarrolla el plugin para la gestión de la sesión de los usuarios que permita acceder al Guano-logout, así como definir su estado según las propiedades.	

Tarea de Ingeniería	
Numero Tarea: 11	Nombre de la Historia de Usuario: U_SD_0.4
Nombre de la Tarea: Estudio de los manejadores de Sesión.	
Tipo de Tarea: Estudio	Puntos Estimados: 1 semanas
Fecha de Inicio: 17/ 03/ 2009	Fecha Fin: 24 / 03/ 2009
Programador Responsable: Barbarita	
Descripción: Estudiar cómo funciona los manejadores de sesión en Linux, más específicó el de GNOME.	

Tarea de Ingeniería	
Numero Tarea: 12	Nombre de la Historia de Usuario: U_SD_0.4

CAPÍTULO 2. ANÁLISIS Y DESARROLLO ÁGIL DE LA SOLUCIÓN PROPUESTA

Nombre de la Tarea: Diseñar prototipo de la interfaz de la aplicación.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semanas
Fecha de Inicio: 17/ 03/ 2009	Fecha Fin: 24 / 03/ 2009
Programador Responsable: Barbarita	
Descripción: Diseñar un prototipo de interfaz de la aplicación con la herramienta Glade.	

Tarea de Ingeniería	
Numero Tarea: 13	Nombre de la Historia de Usuario: U_SD_0.4
Nombre de la Tarea: Conectar la interfaz con las funcionalidades	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 semanas
Fecha de Inicio: 17/ 03/ 2009	Fecha Fin: 24 / 03/ 2009
Programador Responsable: Barbarita	
Descripción: Crear las funcionalidad que permitan gestionar el script Startup desde la interfaz de la aplicación.	

2.8. Diseño con metáforas.

Debido a que la metodología usada como marco de apoyo para guiar el desarrollo de la solución está basada en XP, y dicha metodología plantea en una de las Doce Prácticas de XP que se tiene que guiar todo el desarrollo del sistema a través de una o varias Metáforas acerca de cómo trabaja (o debería trabajar) el Sistema.

Para definir la misma lo primero que se tiene que tener claro es que se entiende por metáfora:

Una metáfora es una historia compartida que describe cómo debería funcionar el sistema. Martin Fowler en *Is Design Dead?* explica que la práctica de la metáfora consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema. Este conjunto de nombres ayuda al diseño y métodos del sistema.

La **metáfora de la solución propuesta** es: “Conjunto de herramientas y funcionalidades que configuran y gestionan el entorno de escritorio Guano, permitiendo personalizar las configuraciones del entorno”

El Diseño con metáforas es sencillamente el diseño de la solución más simple que pueda funcionar y ser implementado en un momento dado del proyecto; lo cual genera el artefacto conocido como

CAPÍTULO 2. ANÁLISIS Y DESARROLLO ÁGIL DE LA SOLUCIÓN PROPUESTA

Modelo de Diseño, que a su vez está compuesto por un diagrama de paquetes, el cual expone dicho diseño.

2.9. Diagrama de paquetes.

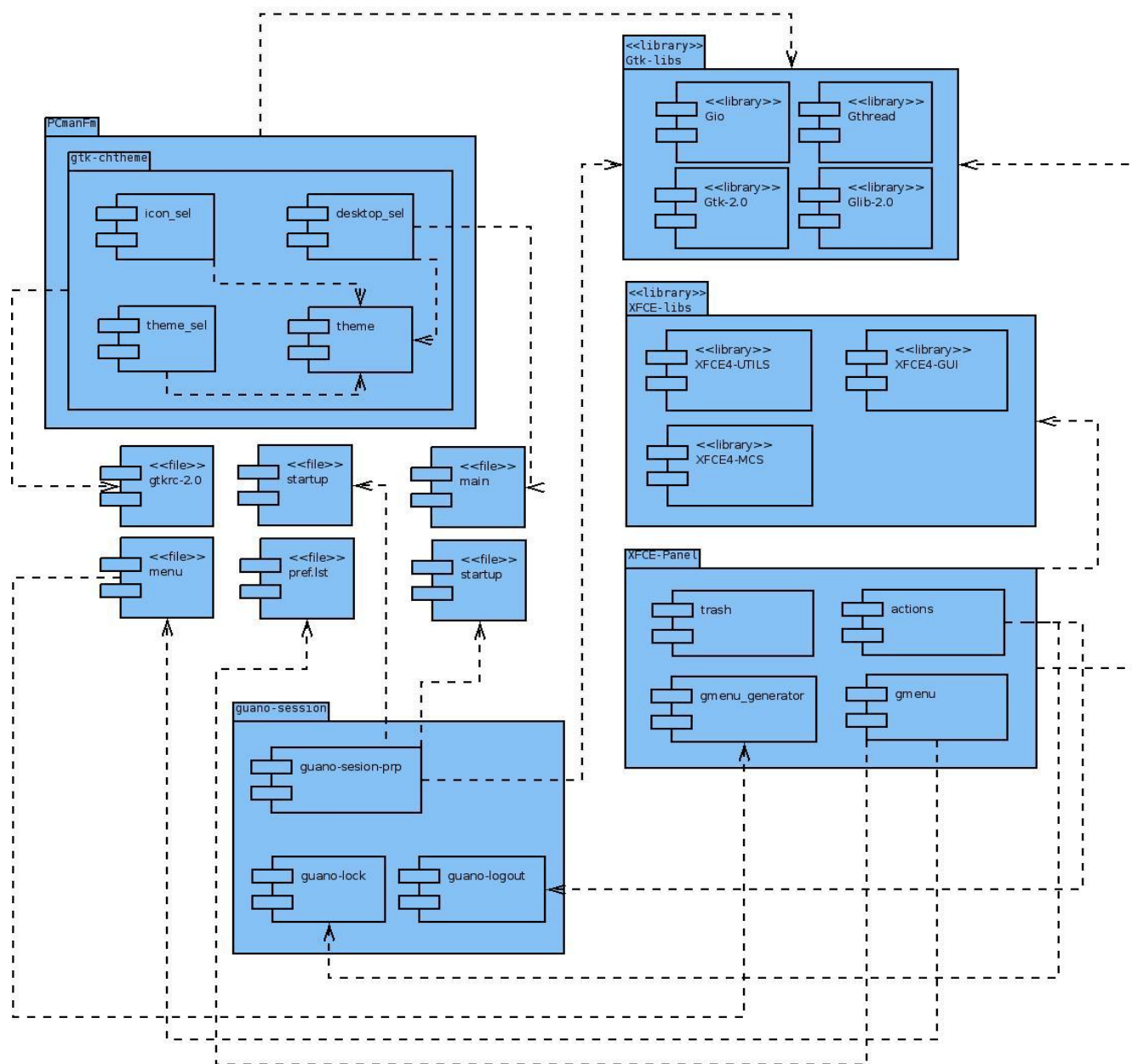


Figura 8. Diagrama de paquetes.

CAPÍTULO 2. ANÁLISIS Y DESARROLLO ÁGIL DE LA SOLUCIÓN PROPUESTA

Como las herramientas de Guano no cumplen con los requerimientos establecidos, en este capítulo, se definió y planificó el proceso de desarrollo de las herramientas requeridas, utilizando como base la metodología SXP.

Durante la planificación se realizó un modelo de dominio que facilitó el entendimiento de los principales conceptos utilizados durante el desarrollo de las aplicaciones. Así como la confección de la lista de los posibles riesgos y la planeación de la mitigación de los mismos. La elaboración del diagrama de paquetes dejó explícito la composición de las herramientas a desarrollar.

Todo esto arrojó que la solución final de esta investigación está centrada en el desarrollo 5 historias de usuario, desglosándose en 13 tareas ingenieriles. Definiendo un plan de liberación para precisar el tiempo de desarrollo por iteraciones, el cual permite obtener una estimación confiable de la conclusión del proceso de desarrollo de las herramientas.

A partir de este proceso se alcanzó como resultado una versión funcional de la solución propuesta, que debe ser validada por un proceso de pruebas, que verifiquen si se encuentra apta para su inclusión en Guano.

3 VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

Una parte muy importante en todo desarrollo de software es la etapa de pruebas, en la metodología SXP se requiere de una etapa de pruebas en cada iteración de desarrollo del producto. En estas etapas se le realizan pruebas al software con el objetivo de verificar si se cumplieron con los objetivos trazados en la iteración, haciéndose el proceso iterativo; e incremental, de pasar el proceso de prueba con éxito en cada iteración para proseguir con el próximo.

En la construcción de la solución se elaboraron un Plan de pruebas que en cada iteración contiene un conjunto de Casos de prueba a los que fue sometida la solución para comprobar el funcionamiento en correspondencia con las Historias de Usuarios definidas.

3.1. Casos de Prueba.

Se definieron casos de prueba para todas las historias de usuario. A continuación se muestran las pruebas que se realizaron mediante tablas definidas en la metodología utilizada para el desarrollo de la aplicación. Los principales elementos que se muestran son las condiciones que el sistema debe cumplir para que se ejecute el caso de prueba así como la entrada y salida de los pasos de ejecución del caso de prueba realizado. Por último se muestra el resultado esperado y la evaluación de la prueba en relación a si fue satisfactoria o no.

Caso de prueba para la historia de usuario U_IDes_01.

Esta sección cubre el conjunto de pruebas que se le realizaron a la historia de usuario: Configurar escritorio.

Este caso de prueba pretende comprobar que la aplicación cambia el fondo de pantalla, las fuentes tipográficas, los temas de iconos y ventana de forma correcta, así como permitir la instalación de nuevos.

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_0.1	Nombre Historia de Usuario: Configurar escritorio.
Nombre de la persona que realiza la prueba: Miguel Albalat Águila.	
Descripción de la Prueba: Se procederá a cambiar el fondo de pantalla.	

CAPÍTULO 3.VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Condiciones de Ejecución: Debe encontrarse iniciada la aplicación PcmánFm.
Entrada / Pasos de ejecución: <ul style="list-style-type: none">❖ El usuario escoge la sección “Escritorio” en la aplicación de preferencias del escritorio.❖ Realiza un click izquierdo en el botón de la imagen.❖ Escoge la nueva imagen de fondo de escritorio y hace click en el botón “Aceptar”.❖ Realiza click izquierdo en el botón “Aplicar”.
Resultado Esperado: Cambio del fondo de escritorio.
Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_0.2	Nombre Historia de Usuario: Configurar escritorio.
Nombre de la persona que realiza la prueba: Miguel Albalat Aguila.	
Descripción de la Prueba: Se procederá a cambiar propiedades del fondo de pantalla.	
Condiciones de Ejecución: Debe encontrarse iniciada la aplicación PcmánFm.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none">• El usuario escoge la sección “Escritorio” en la aplicación de preferencias del escritorio.• Escoge un modo en el menú desplegable del “Modo”.• Realiza click izquierdo en el botón “Aplicar”.	
Resultado Esperado: Cambio de la propiedad del fondo de escritorio.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_0.3	Nombre Historia de Usuario: Configurar escritorio.
Nombre de la persona que realiza la prueba: Miguel Albalat Aguila.	
Descripción de la Prueba: Se procederá a cambiar las propiedades de la fuente del escritorio.	

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Condiciones de Ejecución: Debe encontrarse iniciada la aplicación PcmánFm.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • El usuario escoge la sección “Escritorio” en la aplicación de preferencias del escritorio. • Realiza click izquierdo en el botón del color de la fuente del escritorio. • Escoge un color. • Realiza click izquierdo en el botón “Aceptar”. • Realiza click izquierdo en el botón del color de la sombra de la fuente del escritorio. • Escoge un color. • Realiza click izquierdo en el botón “Aplicar”.
Resultado Esperado: Cambio en el color de la fuente.
Evaluación de la Prueba: Satisfactorio.

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_0.4	Nombre Historia de Usuario: Configurar escritorio.
Nombre de la persona que realiza la prueba: Miguel Albalat Aguila.	
Descripción de la Prueba: Se procede a cambiar el tema de Gtk de las aplicaciones.	
Condiciones de Ejecución: Debe encontrarse iniciada la aplicación PcmánFm.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • El usuario realiza click izquierdo en la pestaña “Tema”. • Escoge un tema en la lista de los posibles temas de Gtk a usar. • Realiza click izquierdo en el botón “Aplicar”. 	
Resultado Esperado: Cambio en el tema de Gtk.	
Evaluación de la Prueba: Satisfactorio.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_0.5	Nombre Historia de Usuario: Configurar escritorio.

CAPÍTULO 3.VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Nombre de la persona que realiza la prueba: Miguel Albalat Aguila.
Descripción de la Prueba: Se procede a cambiar las propiedades de la fuente del sistema.
Condiciones de Ejecución: Debe encontrarse iniciada la aplicación PcmánFm.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • El usuario realiza click izquierdo en la pestaña “Tema”. • El usuario realiza click izquierdo en el botón que indica la fuente y su tamaño. • Escoge una tipo de fuente y tamaño. • Realiza click izquierdo en el botón “Aceptar”. • Realiza click izquierdo en el botón “Aplicar”.
Resultado Esperado: Cambio en el tipo, tamaño y propiedades de la fuente en el sistema.
Evaluación de la Prueba: Satisfactorio.

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_0.6	Nombre Historia de Usuario: Configurar escritorio.
Nombre de la persona que realiza la prueba: Miguel Albalat Aguila.	
Descripción de la Prueba: Se procede a cambiar el tema de iconos.	
Condiciones de Ejecución: Debe encontrarse iniciada la aplicación PcmánFm.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • El usuario realiza click izquierdo en la pestaña “Iconos”. • Escoge un tema en la lista de los posibles temas de iconos a usar. • Realiza click izquierdo en el botón “Aplicar”. 	
Resultado Esperado: Cambio en el tema de iconos.	
Evaluación de la Prueba: Satisfactorio.	

Caso de prueba para la historia de usuario U_IDes_02.

Conjunto de pruebas que se le realizaron a la historia de usuario: Visualizar menú de aplicaciones.

CAPÍTULO 3.VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Este caso de prueba se pretende comprobar que la aplicación genera un menú con las aplicaciones instaladas automáticamente, que visualiza las aplicaciones del sistema por categorías y que se pueden ejecutar las aplicaciones desde el menú.

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_0.7	Nombre Historia de Usuario: Visualizar el menú de aplicaciones.
Nombre de la persona que realiza la prueba: Miguel Albalat Aguila.	
Descripción de la Prueba: Generación del menú de aplicaciones instaladas.	
Condiciones de Ejecución: Sesión de usuario iniciada.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none">▪ Ejecutar el comando gmenu_generator desde un terminal.	
Resultado Esperado: La generación del archivo ~/.config/guano/menu y la salida en la consola del cartel "Ok".	
Evaluación de la Prueba: Satisfactorio.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_0.8	Nombre Historia de Usuario: Visualizar el menú de aplicaciones.
Nombre de la persona que realiza la prueba: Miguel Albalat Águila.	
Descripción de la Prueba: Mostrar menú de aplicaciones.	
Condiciones de Ejecución: Debe encontrarse activa una sesión de usuario y la aplicación guano-panel.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none">▪ Adicionar el plugin "Guano Menú" al panel. Ejecutar el plugin.	
Resultado Esperado: Ejecución del Menú de aplicaciones, mostrándose las aplicaciones más usadas, menú categorizado de aplicaciones y links a carpetas del usuario.	
Evaluación de la Prueba: Satisfactorio.	

CAPÍTULO 3.VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_0.9	Nombre Historia de Usuario: Visualizar el menú de aplicaciones.
Nombre de la persona que realiza la prueba: Miguel Albalat Aguila.	
Descripción de la Prueba: Ejecución de aplicaciones del Menú de aplicaciones.	
Condiciones de Ejecución: Debe encontrarse activa una sesión de usuario y la aplicación guano-panel.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none">▪ Ejecutar el plugin “Guano Menú” en el panel. Realizar click izquierdo en cualquier link.	
Resultado Esperado: Ejecución de la aplicación elegida.	
Evaluación de la Prueba: Satisfactorio.	

Caso de prueba para la historia de usuario U_IDes_03.

Conjunto de pruebas que se le realizaron a la historia de usuario: Incluir plugin para el soporte de la papelera de reciclaje en el panel.

Este caso de prueba se pretende comprobar el plugin adicionado al panel, que permite vaciar la papelera de reciclaje así como visualizar los archivos

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_10	Nombre Historia de Usuario: Incluir plugin para el soporte de la papelera de reciclaje en el panel.
Nombre de la persona que realiza la prueba: Barbarita Ramírez Pérez.	
Descripción de la Prueba: Ejecución del plugin en el panel	
Condiciones de Ejecución: Debe encontrarse activa una sesión de usuario y la aplicación guano-panel.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none">▪ Añadir el plugin al panel.▪ Ejecutar el plugin “Papelera de reciclaje” en el panel.	

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Resultado Esperado: Mostrar la carpeta donde se encuentran los archivos eliminados.
Evaluación de la Prueba: Satisfactorio.

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_11	Nombre Historia de Usuario: Incluir plugin para el soporte de la papelera de reciclaje en el panel.
Nombre de la persona que realiza la prueba: Barbarita Ramírez Pérez.	
Descripción de la Prueba: Ejecución del plugin en el panel	
Condiciones de Ejecución: Debe encontrarse activa una sesión de usuario y la aplicación guano-panel.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none">▪ Añadir el plugin al panel.▪ Ejecutar click izquierdo sobre el plugin “Papelera de reciclaje” en el panel y seleccionar “Vaciar papelera”.	
Resultado Esperado: Eliminar los archivos de la papelera.	
Evaluación de la Prueba: Satisfactorio.	

Caso de prueba para la historia de usuario U_IDes_04.

Conjunto de pruebas que se le realizaron a la historia de usuario: Gestionar el inicio de la sesión de usuario.

Este caso de prueba se pretende comprobar que la aplicación Gestiona el inicio sesión de usuario permitiendo agregar, quitar y editar las aplicaciones que se auto-ejecutan cuando se inicia la sesión de los usuarios.

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_12	Nombre Historia de Usuario: Gestionar el inicio de la sesión de usuario.
Nombre de la persona que realiza la prueba: Barbarita Ramírez Pérez.	

CAPÍTULO 3.VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Descripción de la Prueba: Se procede a añadir una aplicación.
Condiciones de Ejecución: Debe encontrarse activa una sesión de usuario
Entrada / Pasos de ejecución: <ul style="list-style-type: none">▪ El usuario realiza click izquierdo sobre el botón Añadir.▪ Se muestra un diálogo para que introduzca el nombre, el comando y una descripción de la aplicación que se desea agregar.▪ Cuando el usuario proceda a dar click izquierdo en el botón aceptar del dialogo se debe mostrar la nueva aplicación adicionada en el treeview.
Resultado Esperado: Agrego una nueva aplicación.
Evaluación de la Prueba: Satisfactorio.

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_13	Nombre Historia de Usuario: Gestionar el inicio de la sesión de usuario.
Nombre de la persona que realiza la prueba: Barbarita Ramírez Pérez.	
Descripción de la Prueba: Se procede a eliminar una aplicación	
Condiciones de Ejecución: Debe encontrarse activa una sesión de usuario	
Entrada / Pasos de ejecución: <ul style="list-style-type: none">▪ El usuario debe tener previamente seleccionado una aplicación mostrada en el treeview.▪ Cuando realiza click izquierdo en el botón eliminar este elimina la aplicación seleccionada del treeview.	
Resultado Esperado: Agrego una nueva aplicación.	
Evaluación de la Prueba: Satisfactorio.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_14	Nombre Historia de Usuario: Gestionar el inicio de la sesión de usuario.

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Nombre de la persona que realiza la prueba: Barbarita Ramírez Pérez.
Descripción de la Prueba: Se procede editar los datos de una aplicación previamente añadida.
Condiciones de Ejecución: Debe encontrarse activa una sesión de usuario
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> ▪ El usuario debe tener previamente seleccionado una aplicación mostrada en el treeview. ▪ Cuando realiza click izquierdo en el botón editar, esta muestra un diálogo con los datos de la aplicación seleccionada dando la posibilidad de modificar los mismos. ▪ Cuando el usuario selecciona el botón Aceptar modifica los datos del treeview con los nuevos datos de la aplicación.
Resultado Esperado: Modificó los datos de la aplicación.
Evaluación de la Prueba: Satisfactorio.

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_15	Nombre Historia de Usuario: Gestionar el inicio de la sesión de usuario.
Nombre de la persona que realiza la prueba: Barbarita Ramírez Pérez.	
Descripción de la Prueba: Se procede editar los datos de una aplicación previamente añadida.	
Condiciones de Ejecución: Debe encontrarse entrarse nuevamente en la sesión del usuario	
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> ▪ Una vez que se entre a la sesión del usuario este debe auto-ejecutar las aplicaciones que estas guardadas en el manejador de sesión. 	
Resultado Esperado: Todas las aplicaciones guardas en el manejador de sesión se auto-ejecutaron satisfactoriamente una vez que el usuario entro en su sesión	
Evaluación de la Prueba: Satisfactorio.	

Caso de prueba para la historia de usuario U_IDes_05.

Conjunto de pruebas que se le realizaron a la historia de usuario: Incluir plugin para gestionar la sesión de los usuarios en el panel.

CAPÍTULO 3.VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Este caso de prueba se pretende comprobar que el plugin gestionar la sesión de los usuarios a partir del estado que tenga definido, bloquear la sesión, salir o bloquear y salir.

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_16	Nombre Historia de Usuario: Incluir plugin para gestionar la sesión de los usuarios en el panel.
Nombre de la persona que realiza la prueba: Barbarita Ramírez Pérez.	
Descripción de la Prueba: Se ejecuta el plugin para Bloquear la sesión.	
Condiciones de Ejecución: Debe encontrarse entrarse nuevamente en la sesión del usuario	
Entrada / Pasos de ejecución: <ul style="list-style-type: none">▪ Cuando se da click izquierdo en el plugin este procede a bloquear la sesión del usuario.	
Resultado Esperado: Se bloquea la sesión del usuario, donde para volver a entrar tiene que introducir nuevamente la contraseña.	
Evaluación de la Prueba: Satisfactorio.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_16	Nombre Historia de Usuario: Incluir plugin para gestionar la sesión de los usuarios en el panel.
Nombre de la persona que realiza la prueba: Barbarita Ramírez Pérez.	
Descripción de la Prueba: Se ejecuta el plugin para salir la sesión.	
Condiciones de Ejecución: Debe encontrarse entrarse nuevamente en la sesión del usuario	
Entrada / Pasos de ejecución: <ul style="list-style-type: none">▪ Cuando se da click izquierdo en el plugin hace un llamado a la aplicación Guano-logout.	
Resultado Esperado: Se ejecuta la aplicación Guano-logout.	
Evaluación de la Prueba: Satisfactorio.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: U_SDPC_16	Nombre Historia de Usuario: Incluir plugin para gestionar la sesión de los usuarios en el panel.
Nombre de la persona que realiza la prueba: Barbarita Ramírez Pérez.	
Descripción de la Prueba: Cambiar el estado del plugin.	
Condiciones de Ejecución: Debe encontrarse entrarse nuevamente en la sesión del usuario	
Entrada / Pasos de ejecución: <ul style="list-style-type: none">▪ Cuando se realiza click derecho sobre el plugin escoger la opción Propiedades▪ En el dialogo Acción de panel elegir el tipo de acción(bloquear, salir o salir y bloquear)	
Resultado Esperado: Se cambia el estado del plugin en el panel	
Evaluación de la Prueba: Satisfactorio.	

3.1. Resultados obtenidos.

En este epígrafe se relacionan los resultados alcanzados durante el proceso de desarrollo del presente trabajo.

Como fruto de este trabajo se han obtenido una serie de herramientas que permite al usuario una mejor interacción con el entorno de escritorio. Estas se encuentran en la versión 0.1 del entorno de escritorio Guano, que está siendo usado actualmente en algunos laboratorios de las Facultad 4, 3 y 10 de la UCI.

3.1.1. Acerca de las funcionalidades obtenidas.

Las herramientas obtenidas como producto del proceso de desarrollo llevado a cabo son:

- Herramienta de configuración de escritorio, que permite cambiar el fondo de escritorio, las fuentes tipográficas, los temas de iconos y los temas de Gtk.
- Menú de inicio o aplicaciones.
- Plugins del panel para el manejo de la papelera de reciclaje así como para la gestión de la sesión de usuario.
- Aplicación de gestión de inicio de sesión de usuario, que permite adicionar o eliminar aplicaciones en el inicio de la sesión.

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

A modo de conclusiones, en este capítulo se describieron las pruebas realizadas para validar la calidad de las herramientas desarrolladas obteniendo resultados exitosos. Siendo además avalado el funcionamiento de estas por la utilización de Guano en eventos nacionales e internacionales así como en los laboratorios de distintas facultades en la Universidad de las Ciencias Informáticas.

CONCLUSIONES GENERALES

Como resultado de la investigación se dio cumplimiento a los objetivos propuestos, dotando a Guano de herramientas que facilitan al usuario la interacción amigable con la computadora. Estas utilidades mejorarán la compenetración de los usuarios con Guano lo cual hará posible que se unan cada vez más personas al proceso de retroalimentación - desarrollo del producto.

Con el futuro desenvolvimiento de estas herramientas será cada vez más posible el proceso de migración de Software Libre en Cuba, incluso en aquellas máquinas donde por sus déficits de memoria y capacidad de procesamiento no se puedan ejecutar los entornos de escritorio estándares de Linux.

REFERENCIAS BIBLIOGRÁFICAS.

1. **Rodríguez Pino, Adisleydis y Firvida Donestevez, Abel Alfonso.** *Guano, entorno de escritorio cubano, libre y de código abierto.* C.Habana : s.n., 2008.
2. Start menu - Wikipedia, the free encyclopedia:. *Wikipedia, the free encyclopedia* . [En línea] [Citado el: 15 de marzo de 2009.] http://en.wikipedia.org/wiki/Start_menu#cite_note-0.
3. Descargar GnoMenu 1.6 gratis en español para Ubuntu - Uptodown.com. *Uptodown.com*. [En línea] [Citado el: 20 de Mayo de 2009.] <http://gnomenu.uptodown.com/ubuntu/>.
4. VivaLinux! » KBFX: El futuro menú de KDE. *Vivalinux!* [En línea] [Citado el: 15 de Marzo de 2009.] <http://www.vivalinux.com.ar/desktop/kbfx-0.4.9.2.html>.
5. Kickoff - openSUSE. *Discover it*. [En línea] [Citado el: 10 de Marzo de 2009.] <http://en.opensuse.org/Kickoff>.
6. Sesión - Wikipedia, la enciclopedia libre:. *Wikipedia, la enciclopedia de contenido libre* . [En línea] [Citado el: 18 de enero de 2009.] <http://es.wikipedia.org/wiki/Sesi%C3%B3n>.
7. Glosario - Alvarado:. *Alvarado*. [En línea] [Citado el: 18 de enero de 2009.] <http://www.alvarado-tolima.gov.co/glosario.shtml?apc=I----&s=b>.
8. 2. Configuración del Gestor de sesiones (Solaris Common Desktop Environment: Guía avanzada del usuario y del administrador del sistema) - Sun Microsystems:. *2. Configuración del Gestor de sesiones (Solaris Common Desktop Environment: Guía avanzada del usuario y del administrador del sistema) - Sun Microsystems*:. [En línea] [Citado el: 18 de enero de 2009.] <http://docs.sun.com/app/docs/doc/816-4016/6ma8otj1p?l=es&a=view>.
9. **Valdés Villarrubia, Alejandro.** Lenguaje C. *Sistema de Clonación y Distribución de Imágenes de Sistemas Operativos.* Ciudad de la Habana : s.n., 2007.
10. **Pilone, Dan.** *UML 2.0 in A nutshell.* s.l. : O'Reilly Germany, 2006.
11. GTK+ - About. *The GTK+ Project*. [En línea] The GTK+ Team. [Citado el: 22 de noviembre de 2008.] <http://www.gtk.org/index.php>.
12. GLib Reference Manual. *Gnome Documentation Library*. [En línea] [Citado el: 20 de enero de 2009.] <http://library.gnome.org/devel/glib/stable/>.
13. NationMaster - Encyclopedia: GDK. *NationMaster.com*. [En línea] [Citado el: 04 de enero de 2009.] <http://www.nationmaster.com/encyclopedia/GDK>.
14. Glade y libglade:. *Glade y libglade*. [En línea] [Citado el: 11 de abril de 2009.] <http://www.calcifer.org/documentos/libglade/>.

REFERENCIA BIBLIOGRÁFICA.

15. Geany : About:. *Geany*. [En línea] [Citado el: 20 de febrero de 2009.]
<http://www.geany.org/Main/About>.

16. Dia a drawing program:. *Dia*. [En línea] [Citado el: 17 de enero de 2009.]
<http://projects.gnome.org/dia/>.

17. Herramientas para UML en Linux ?? TechTear - Blog Magazine de Tecnología:. *Techtear.com*. [En línea] [Citado el: 19 de enero de 2009.] <http://www.techtear.com/2007/06/24/herramientas-para-uml-en-linux/>.

18. **Guillemí Martín, Jessie**. Las Metodologías de Desarrollo de Software. [aut. libro] Análisis y Modelado de la Solución Informática para el proceso de Inscripción en los Registros Mercantiles de la República Bolivariana de Venezuela. proceso. Caracas : s.n., 2007.

19. **Canós, José H, Letelier, Patricio y Penadés, M^a Carmen**. *Metodologías Ágiles en el Desarrollo de Software*.

20. **Peldeño Romero, Gladys Marsi**. *MA-GMPR-UR2 Metodología ágil para proyectos de software libre*. Ciudad de la Habana : Universidad de las Ciencias Informáticas, Facultad 10, 2008.

21. **Goñi Oramas, Angel y Hurtado Fedoróvich, Mijail**. Modelo de Dominio. *Herramienta gráfica para efectuar la gestión de paquetes en la distribución de GNU/Linux Nova*. Ciudad de la habana : s.n., 2008.

22. **Peñalver Romero, Gladys Marsi**. Modelo de historia de usuario. *Expediente para metodología SXP*. Ciudad de la Habana : s.n., 2008.

BIBLIOGRAFÍA.

Benson, Calum, y otros. *GNOME Human Interface Guidelines 2.0.* s.l. : Free Software Foundation, 2004.

Romero, Gladys Marsi Peñalver. *Expediente para metodología SXP.* Ciudad de la Habana : s.n., 2008.

Schmuller, Joseph. *Aprendiendo UML en 24 horas.*

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software.* 2000.

Canós, José H., Letelier, Patricio y Penadés, Ma Carmen. *Métodologías Ágiles en el Desarrollo de Software.*

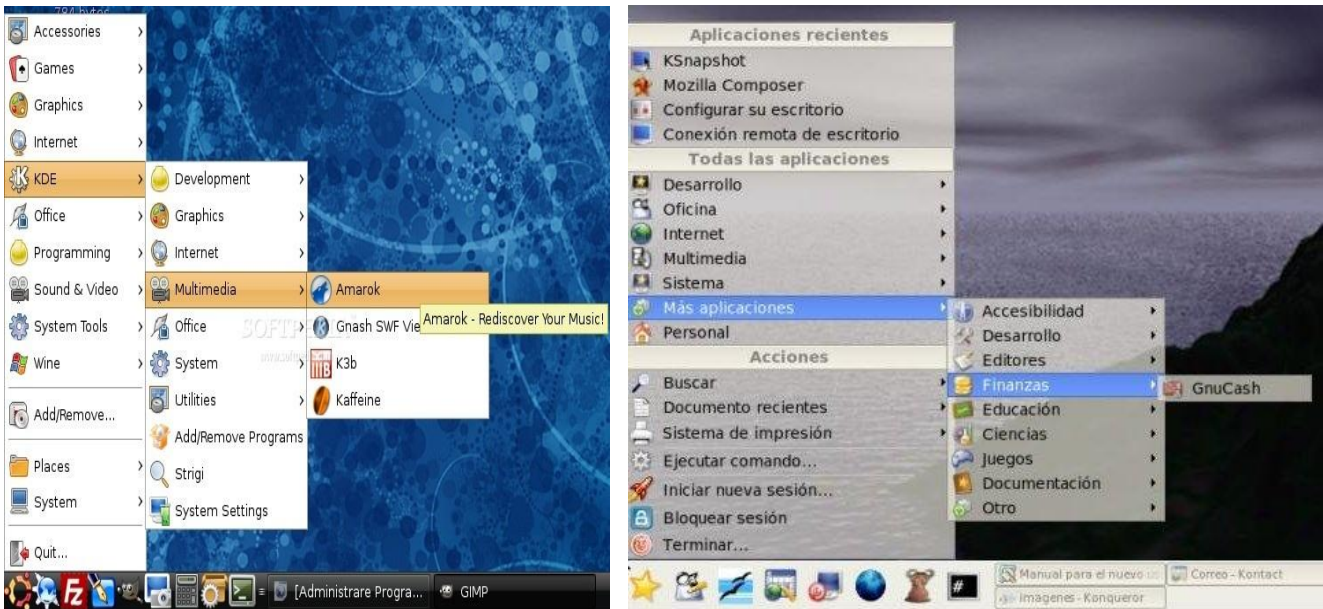
Beck, Kent. *Extreme Programming Explained. Embrace Change.* 1999.

ANEXOS

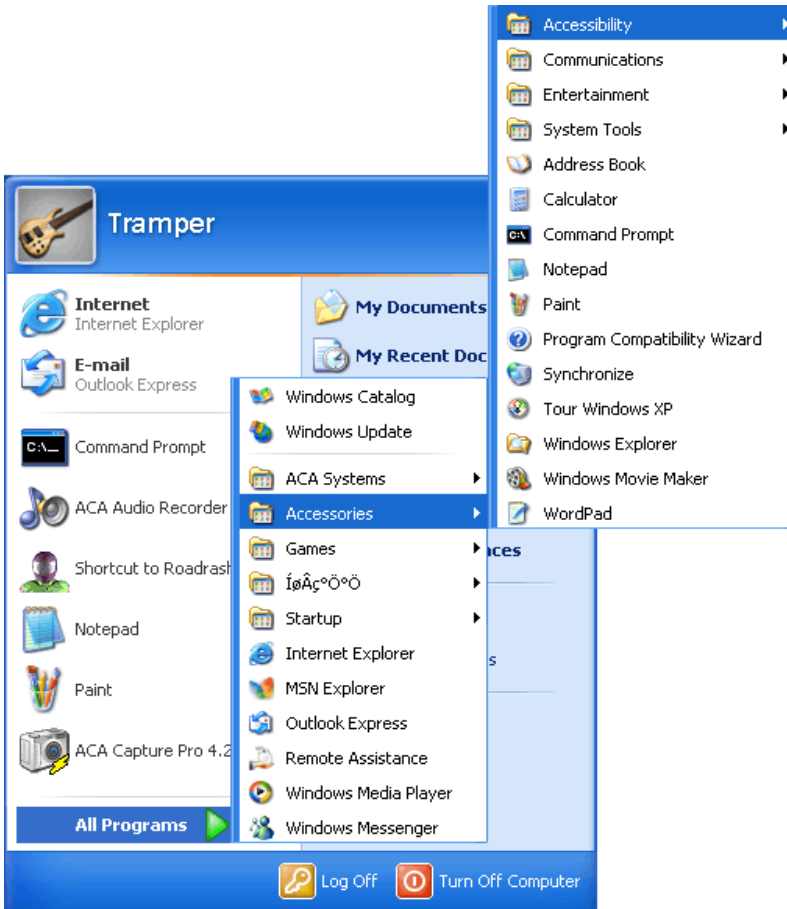
Anexo 1: Encuesta sobre migración de escritorio.

Funcionalidades mostradas en la encuesta.	
No.	Funcionalidad
F1	Poder cambiar la apariencia y configuración del escritorio.
F2	Conservar los accesos rápidos de teclado estilo Windows.
F3	Conservar los accesos directos en mi escritorio a Mi PC, así como poder añadir nuevos.
F4	Conservar la estructura del menú y que recuerde mis aplicaciones recientes y preferidas.
F5	Tener la papelera de reciclaje.
F6	Conservar la barra de accesos rápidos.
F7	Conservar aplicaciones para la conexión por escritorio remoto.
F8	Conservar funcionalidades para compartir carpetas e impresoras en mi red.
F9	Mantener las funcionalidades de vistas de carpeta o imágenes.
F10	Conservar funcionalidades para establecer conexiones con otras PC de mi red.
F11	Conservar los juegos predeterminados de Windows.
F12	Conservar la posibilidad de mantener archivos ocultos.
F13	Tener el Panel de Control.
F14	Navegar y explorar distintos protocolos de red con el mismo explorador (http, ftp, etc....).
F15	Mantener las funcionalidades para la interacción con dispositivos extraíbles.
F16	Conservar la funcionalidad de Buscar en mis carpetas.

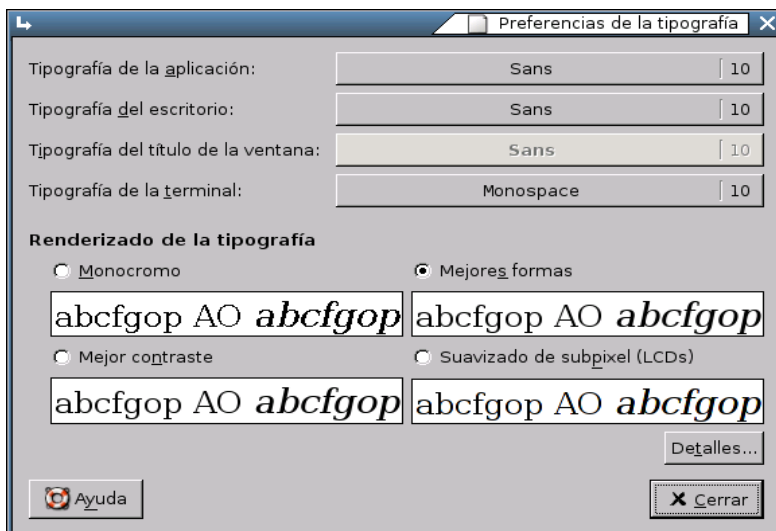
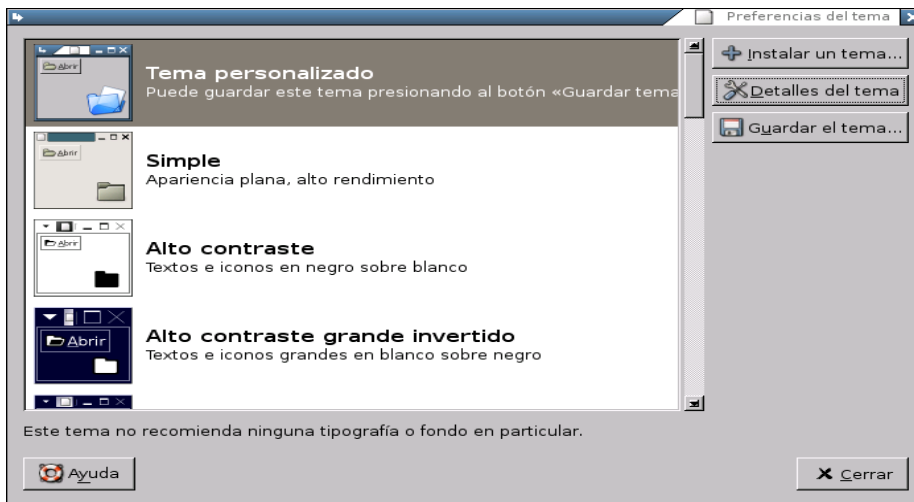
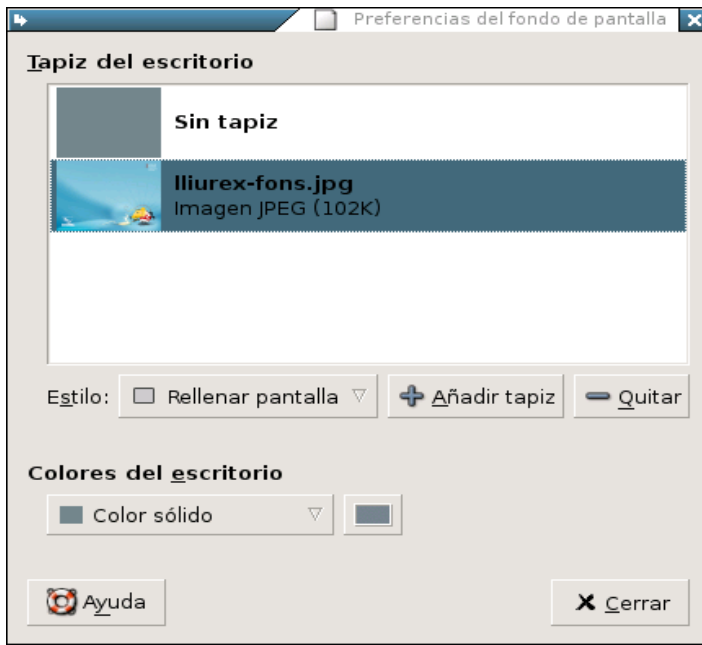
Anexo 2: Menú por defecto de GNOME y KDE.

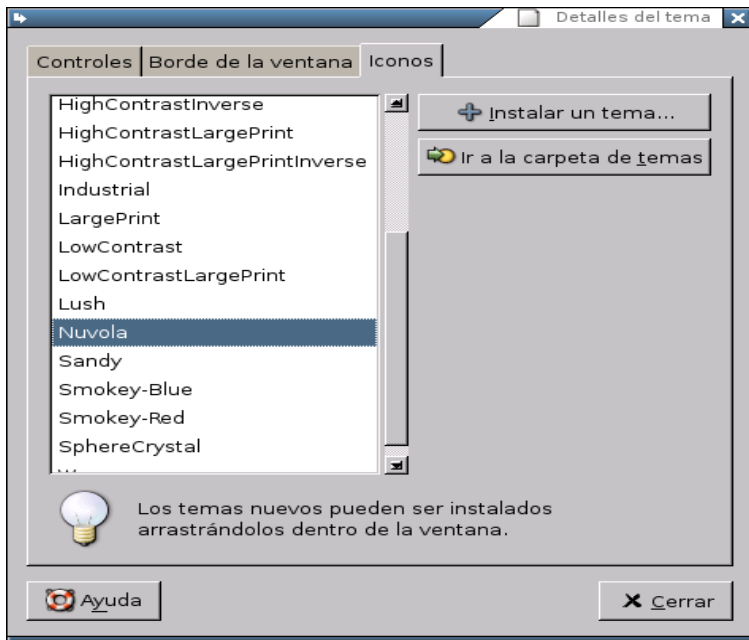


Anexo 3: Menú de inicio de Windows XP.

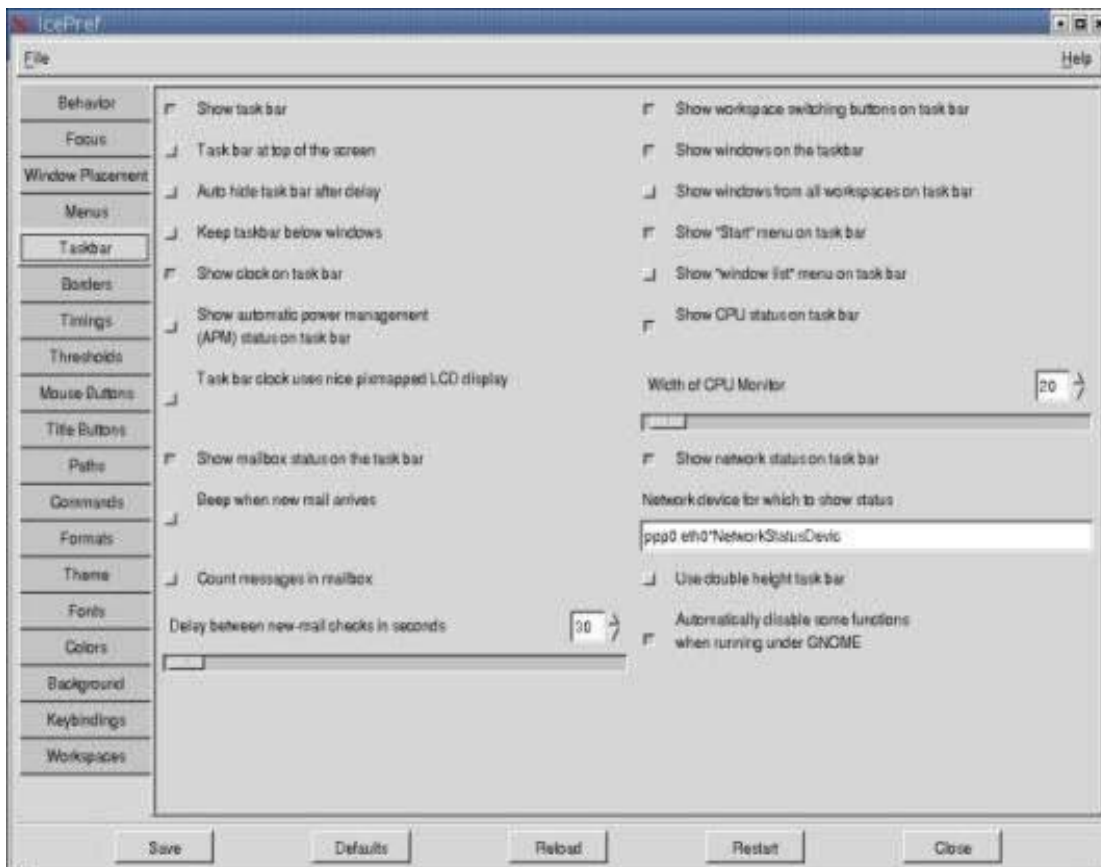


Anexo 4: Herramientas de configuración de GNOME.

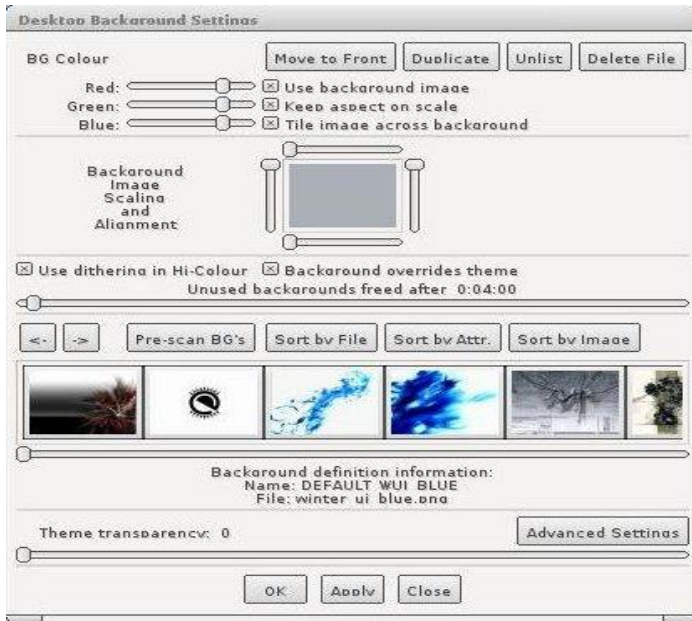




Anexo 5: Herramienta de configuración de XceWM.

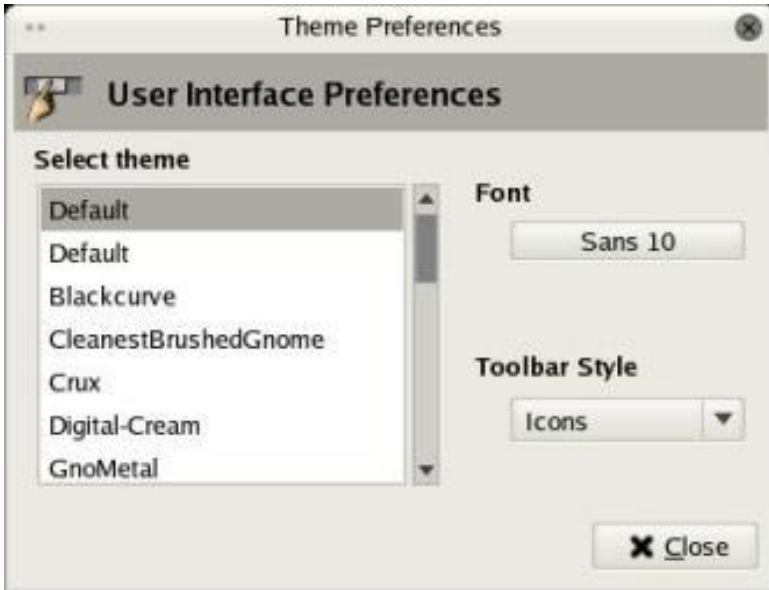


Anexo 6: Herramienta de configuración de Enlightenment.

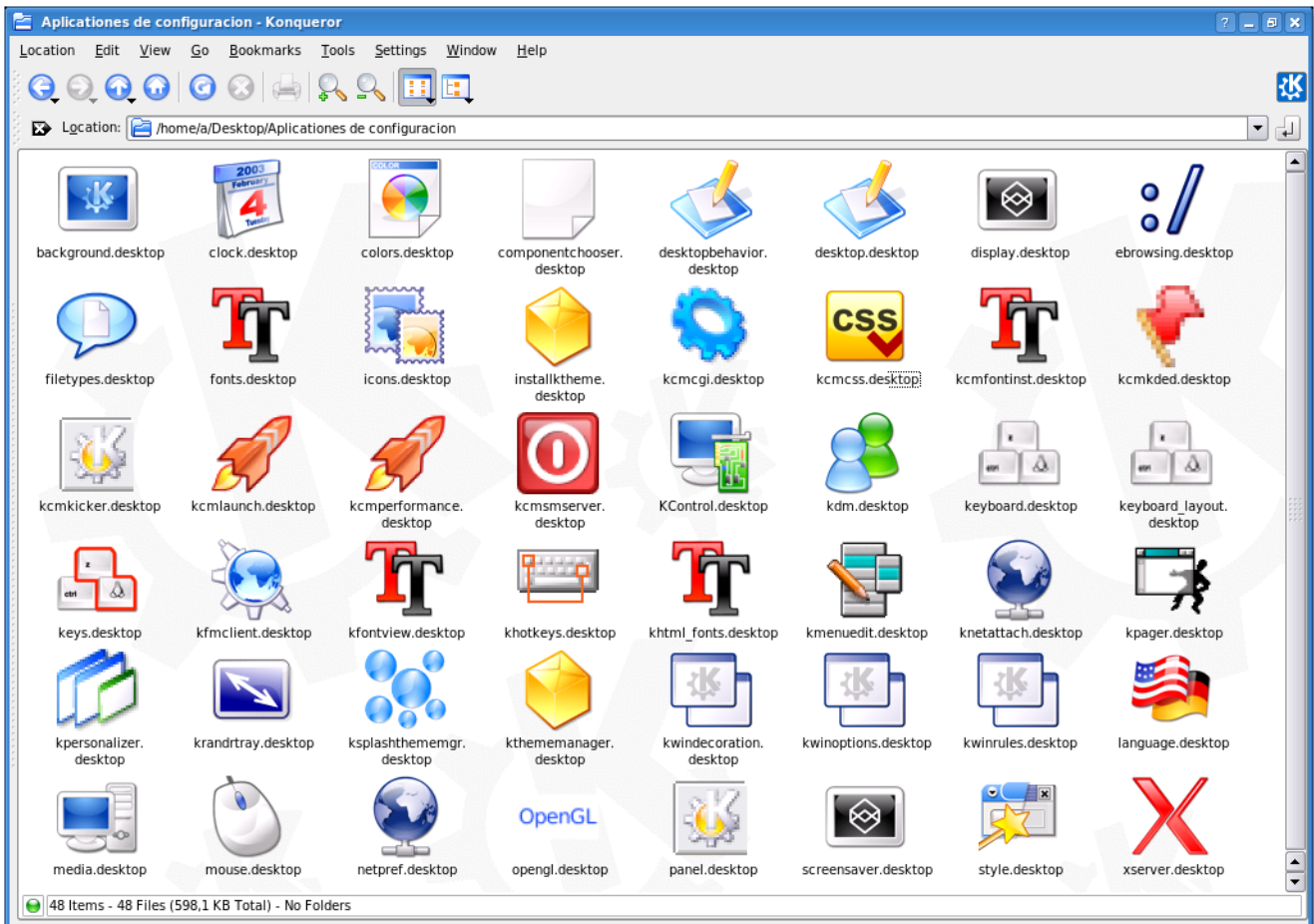


Anexo 7: Herramientas de configuración de Xfce.

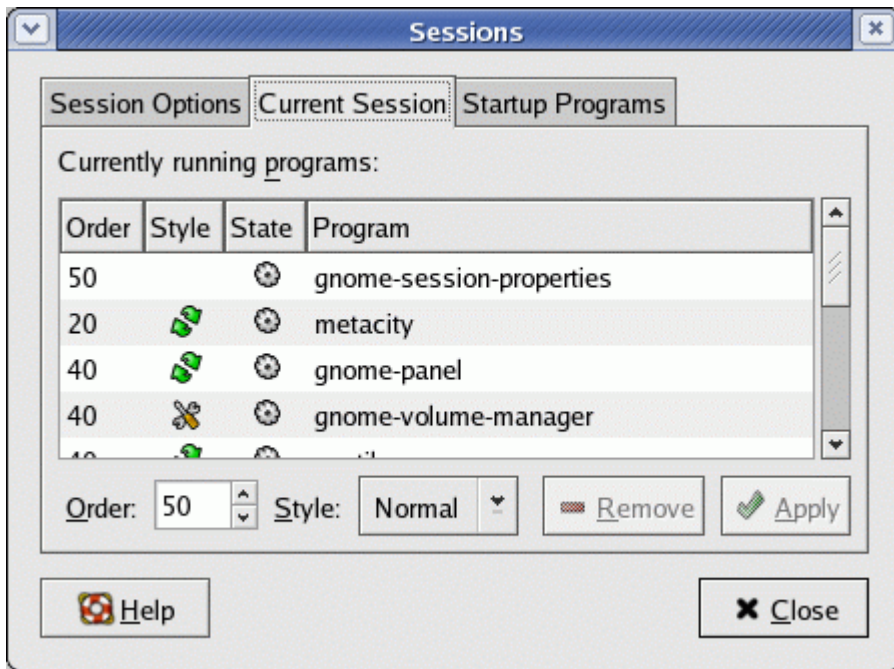




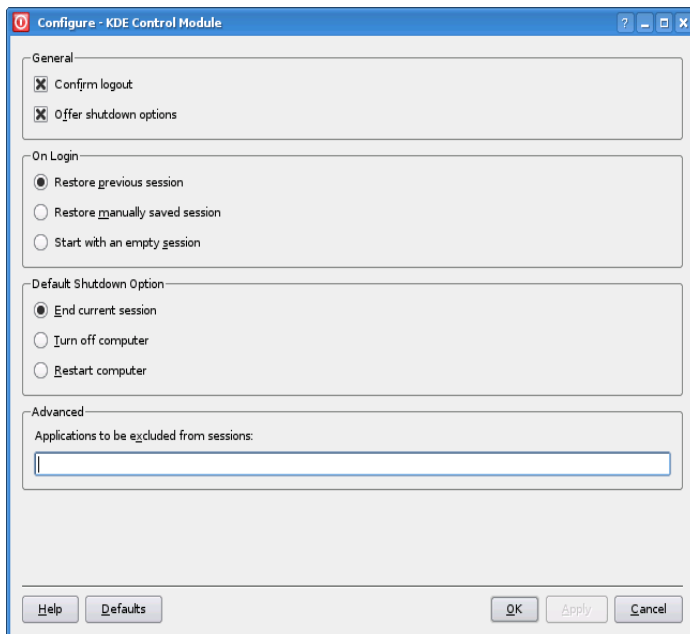
Anexo 8: Herramientas de configuración de KDE.



Anexo 9: Administrador de sesión de GNOME.



Anexo 10: Aplicación de administración de sesiones de KDE.



Anexo 11: Principios del Manifiesto Ágil.

- I. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- II. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- III. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- IV. El personal del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- V. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- VI. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- VII. El software que funciona es la medida principal de progreso.
- VIII. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- IX. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- X. La simplicidad es esencial.
- XI. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.

En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

Anexo 12: Plantilla de Concepción del sistema.

Concepción del sistema Metapaquete XXX			
ConFileM 0.01			
Control de versiones			
Fecha	Versión	Descripción	Autor
02/02/09	0,1		Barbarita Ramírez

- **Polo productivo.**
Software Libre
- **Clasificación del proyecto.**
Desarrollo de aplicación
- **Tipo de proyecto.**
Nacional
- **Resumen:**
Este documento tiene una breve descripción de la propuesta a desarrollar, los roles que va a tener el proyecto, las herramientas a utilizar, la visión y el alcance del mismo.
Palabras claves:
- **Surgimiento.**
En la UCI el proyecto Nova está desarrollando un entorno de escritorio, Guano, como una alternativa a usar en las maquinas de bajas prestaciones o obsoletas, este no cuenta con herramientas de configuración por lo que no está dirigido a usuarios finales, usuarios que no tienen mucho conocimiento de técnicas de programación o de los entorno libres. Para agilizar el desarrollo de Guano y de esta forma agilizar la migración se propuso la creación XXX , Metapaquete de configuración para el entorno de escritorio Guano.
- **¿Qué es?**
El Metapaquete XXX está conformado partiendo de los requisitos que los clientes querían para Guano, este está compuesto por un menú de aplicación que visualiza las aplicaciones del sistema por categorías, una aplicación para modificar las configuraciones del escritorio ya sea fondo de escritorio, las fuentes tipográficas del sistema, el tema de iconos y tema de Gtk, una aplicación que maneje la sesión de los usuarios dando la posibilidad de poder escoger que aplicaciones quiero que se auto-ejecuten una vez que se inicie la sesión y por ultimo agregar soporte para la papelera.
 - **Metodología a utilizar.**
La metodología a utilizar es SXP, es creada a partir de la unión de XP y Scrum, en la UCI, en el proyecto Unicornios, responde a las características particulares de los proyectos de la facultad 10 pues garantiza mayor organización en cuanto a los artefactos que sustentan la metodología, logrando un procedimiento ágil para proyectos de Software Libre.
 - **Involucrados.**
[Breve descripción del/los involucrado(s) y clasificar al/los involucrado(s) como experto, gurú, accionista, etc.]

▪ **Roles.**

Rol	Responsabilidad	Nombre
<i>Gerente (Management)</i>	<i>Es el responsable de tomar las decisiones finales, acerca de estándares y convenciones a seguir durante el proyecto.</i>	<i>Miguel Albalat</i>
<i>Cliente (Customer)</i>	<i>Contribuye a definir las historias de usuario y los casos de prueba de aceptación, para validar su implementación.</i>	<i>Equipo de desarrollo de Guano Encargado de los laboratorios de la facultad 3</i>
<i>Programadores (Programmers)</i>	<i>Define las tareas de ingeniería y produce el código del sistema. Además selecciona el estándar de programación a utilizar, controlando incluso la gestión de cambios.</i>	<i>Román Miguel Valdivia Genó Yuliette Tain Domínguez</i>
<i>Analista (Analyst)</i>	<i>Escribe la concepción del sistema y las historias de usuario. Crea el Modelo de historia de usuario del negocio y la LRP. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio</i>	<i>Barbarita Ramirez</i>
<i>Encargado de Pruebas (Tester)</i>	<i>Escribe los casos de prueba de aceptación. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas</i>	<i>Barbarita Ramirez Perez Miguel Albalat</i>

▪ **Misión.**

El Metapaquete XXX propuesto debe ser capaz de realizar las configuraciones del entorno Guano haciendo de este un entorno con más aceptación.

▪ **Visión.**

[Aspiraciones futuras para el sistema a desarrollar.]

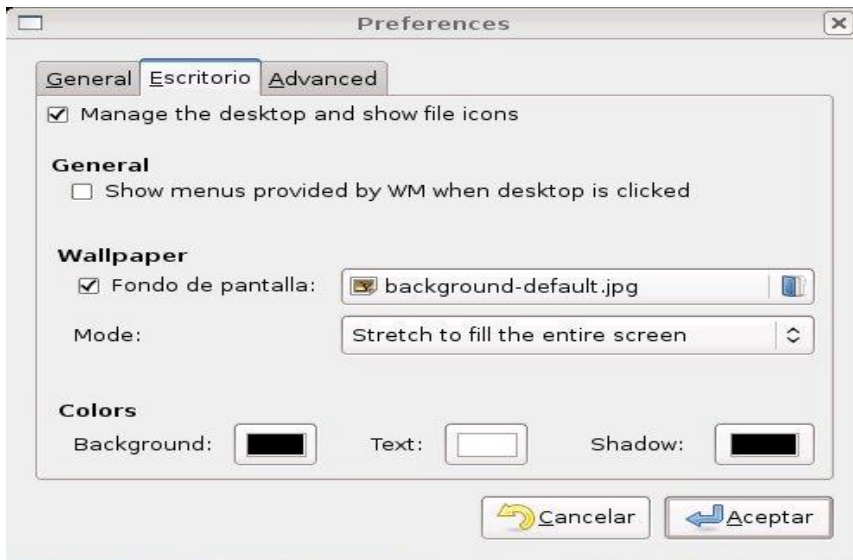
▪ **Alcance.**

▪ **Herramientas Utilizadas.**

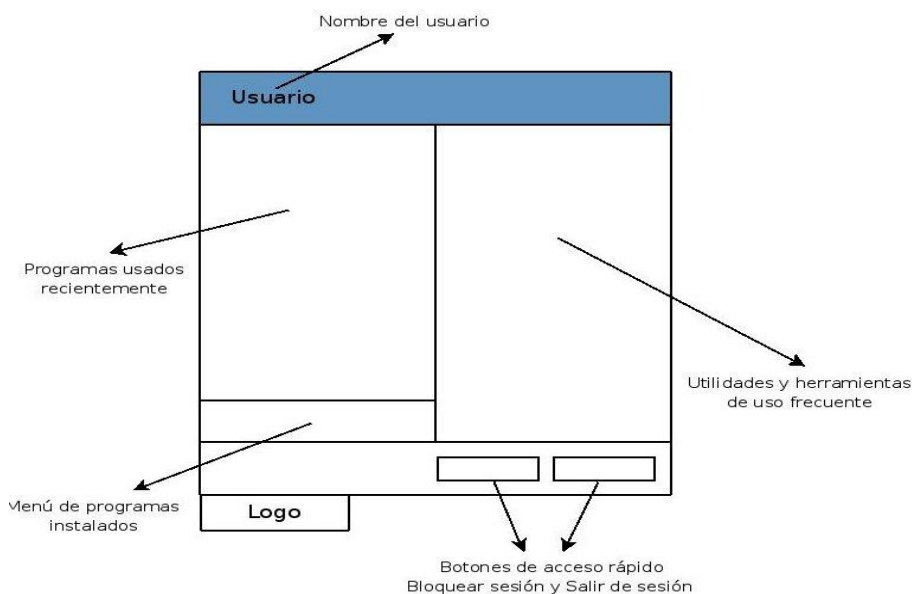
BoUML

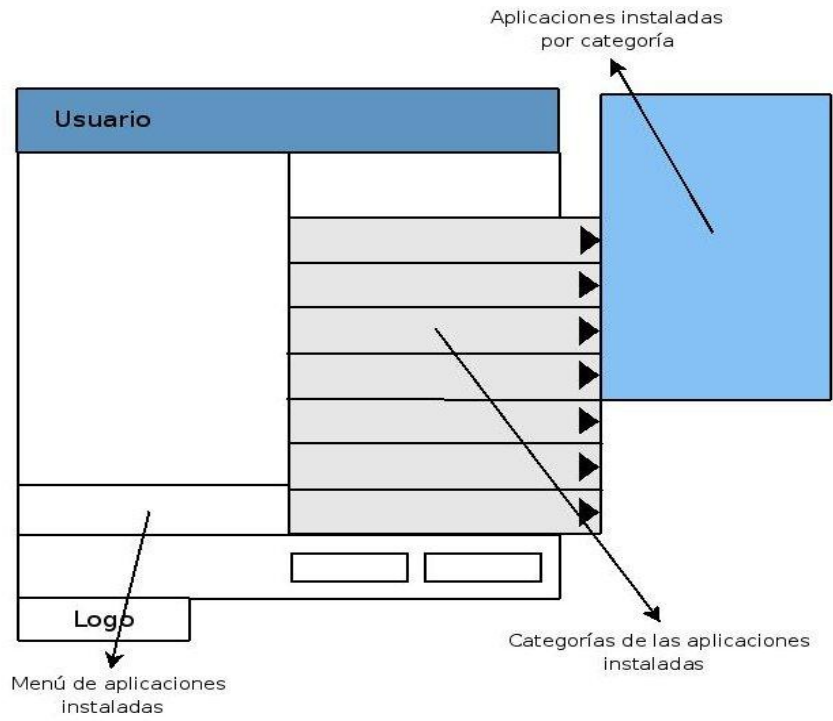
- UML
- Geany
- Glade
- **Solución propuesta.**

Anexo 13: Herramienta de configuración de escritorio del PCmanFM.

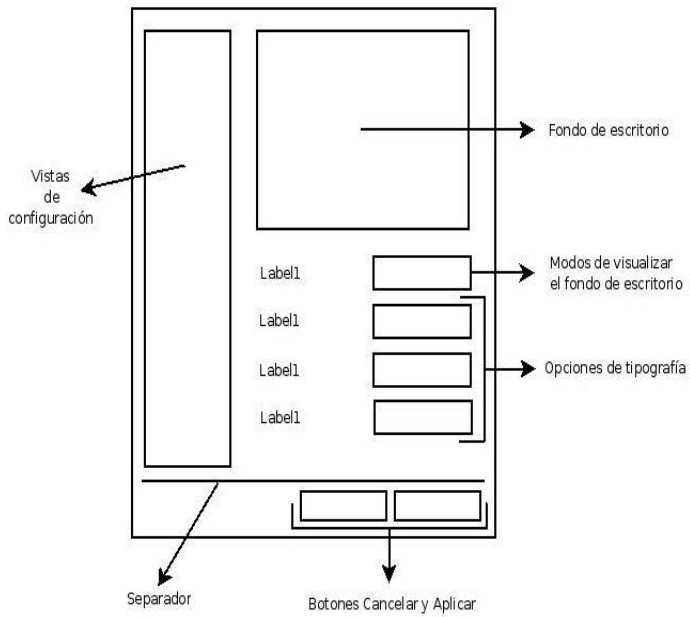


Anexo 14: Prototipo de la solución propuesta.

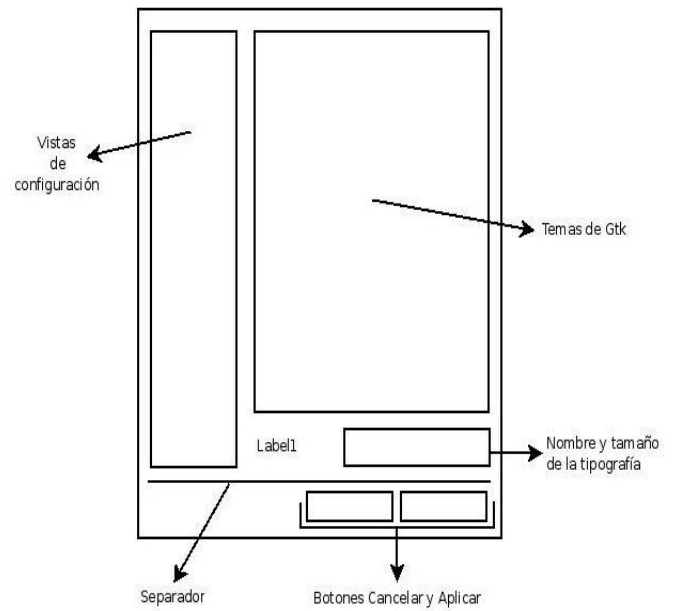




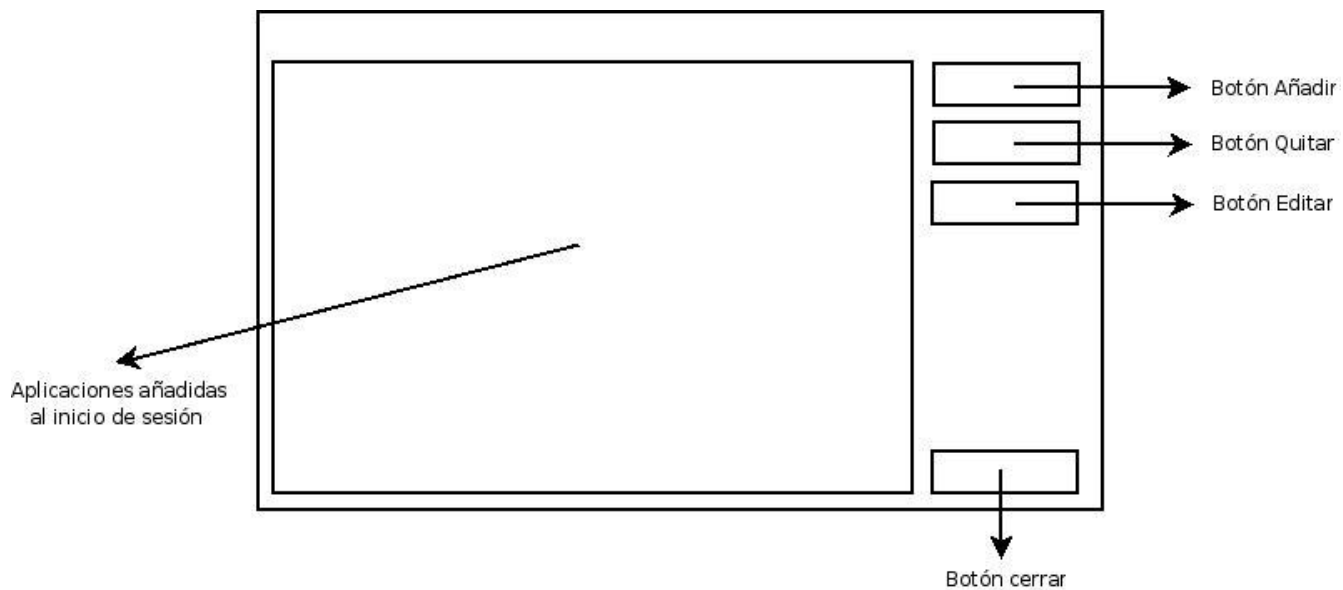
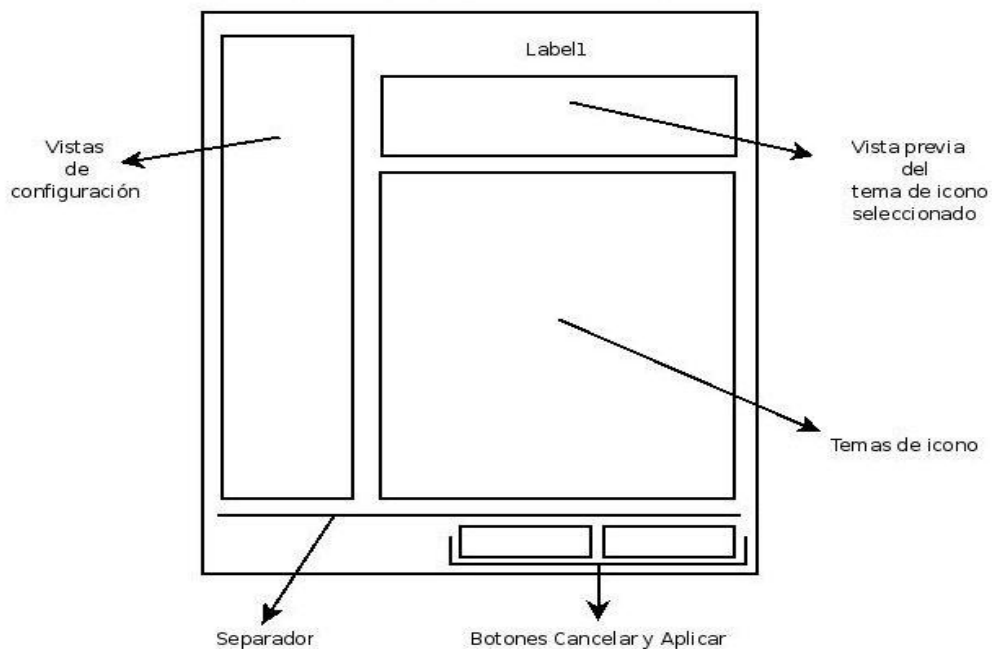
Vista de configuración del escritorio



Vista de configuración de temas de Gtk



Vista de configuración de temas de icono



GLOSARIO DE TÉRMINOS

Applet: Es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web. El applet debe ejecutarse en un contenedor, que lo proporciona un programa anfitrión, mediante un plugin, o en aplicaciones como teléfonos móviles que soportan el modelo de programación por applets.

Login: Identificación o nombre electrónico de un usuario de cualquier sistema compartido y en particular de los servicios de correo electrónico.

Scroll:

OpenSuse: Es el nombre de la distribución y proyecto libre auspiciado por Novell y AMD para el desarrollo y mantenimiento de un sistema operativo basado en Linux.

Microsoft: Microsoft Corporation (NASDAQ: MSFT) es una empresa multinacional estadounidense, fundada en 1975 por Bill Gates y Paul Allen. Dedicada al sector de la informática, que desarrolla, fabrica, licencia y produce software y equipos electrónicos.

Dispositivos periféricos: Dispositivos a través de los cuales la computadora se comunica con el mundo exterior, como a los sistemas que almacenan o archivan la información, sirviendo de memoria auxiliar de la memoria principal.

XML: Metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

SVG: Lenguaje para describir gráficos vectoriales bidimensionales, tanto estáticos como animados.

RUP: El Proceso Unificado Racional (Rational Unified Process en inglés) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Biblioteca: Término informático para referirse a las bibliotecas de vínculos dinámicos, conocidas también como librerías.

Scrum Master: Es uno de los Roles de Scrum. Es la persona responsable para asegurarse que el proceso de Scrum es utilizado correctamente.

B: Nombre de un lenguaje de programación desarrollado en los Bell Labs, predecesor del lenguaje de programación C.

LINUX: Núcleo presente en los sistemas GNU/Linux.

GNU/Linux: GNU/Linux es, a simple vista, un Sistema Operativo. Es una implementación de libre

GLOSARIO DE TÉRMINOS

distribución UNIX para computadoras personales, servidores, y estaciones de trabajo. Fue desarrollado para el i386 y ahora soporta los procesadores i486, Pentium, Pentium Pro y Pentium II, así como los clones AMD y Cyrix. También soporta máquinas basadas en SPARC, DEC Alpha, PowerPC/PowerMac, y Mac/Amiga Motorola 680x0. Como sistema operativo, GNU/Linux es muy eficiente y tiene un excelente diseño.

Human interface guidelines (HIG): Son documentos que ofrecen a los desarrolladores de aplicaciones un conjunto de recomendaciones destinadas a mejorar la experiencia para los usuarios, haciendo interfaces de uso más intuitivos, aprendibles, y constantes.

Minimalista: El término minimalista, en su ámbito más general, es referido a cualquier cosa que se haya desnudado a lo esencial, despojada de elementos sobrantes, o que proporciona solo un esbozo de su estructura, y minimalismo es la tendencia a reducir a lo esencial.

Sistema operativo: Un sistema operativo es un software de sistema, es decir, un conjunto de programas de computadora destinado a permitir una administración eficaz de sus recursos. Comienza a trabajar cuando se enciende el computador, y gestiona el hardware de la máquina desde los niveles más básicos, permitiendo también la interacción con el usuario.

SLIM: Es un gestor de acceso ligero.

UNIX: Unix (registrado oficialmente como UNIX®) es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T.

