

Arquitectura para Entorno Gráfico enfocado a Multimedia

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

**Autor(es): Eiger Mora Moredo
Yunier Vega Rodríguez**

**Tutor: Juan José Galano Frutos
Co-Tutor: Roberto Rodríguez Montoya**



xeiba

*“Revolución es sentido del momento histórico; es cambiar
todo lo que debe ser cambiado...”*

Fidel Castro Ruz

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter no exclusivo.

Para que así conste firmo la presente a los ___ días del mes de abril del año 2009

Yunier Vega Rodríguez

Firma del Autor

Eiger Mora Moredo

Firma del Autor

Lic. Juan José Galano Frutos

Firma del Tutor

Ing. Roberto Rodríguez Montoya

Firma del Co-Tutor

Datos de Contacto

Nombre y apellidos del tutor: Juan José Galano Frutos

Email: juanjog@uci.cu

Currículum Vitae: Graduado en Licenciatura en Ciencias Químicas en el año 2004. Profesor asistente de la UCI. Imparte cursos optativos de herramientas sobre software libre. Investigador en metodologías de migración a software libre.

Nombre y apellidos del co-tutor: Roberto Rodríguez Montoya

Email: robertom@uci.cu

Currículum Vitae: Graduado en Ingeniería en Automática en el año 2004, profesor instructor de la UCI. Ha impartido la asignatura de Teleinformática I y II durante 3 años, e imparte el curso optativo de Servicios Telemáticos.

Agradecimientos

A nuestros tutores, Juan José Galano Frutos y Roberto Rodríguez Montoya,
por el tiempo invertido y la guía oportuna.

A Abel Meneses Abad, quien se convirtiera
en nuestro tercer tutor.

A todos los que de una u otra forma han colaborado
con la realización de este trabajo.

Dedicatoria

A nuestros seres queridos.

A la vida y los sueños.

RESUMEN

El negocio de la informática se ha extendido considerablemente en los últimos tiempos, llegando a los públicos más insospechados, con las costumbres más diversas, al tiempo que las interfaces gráficas de los sistemas operativos se mantienen prácticamente sin cambio alguno desde hace más de 30 años, requieren, ante esta nueva realidad, ser sustituidas por nuevos paradigmas que se ajusten a los usuarios y sus necesidades.

En ese escenario la combinación de tecnologías como la multimedia o el 3D, podrían ofrecer productos muy diferentes a los actuales, con un poder de interactividad superior, lo cual para Cuba resulta no solo deseable, sino también necesario en virtud de los numerosos planes que como parte de la Batalla de Ideas contribuyen a la universalización de la enseñanza y con ella a la masificación de la cultura.

La posibilidad que brinda el software libre de desarrollar productos nacionales, hacen que esta idea sea razonable, más allá incluso de las restricciones del bloqueo y la hegemonía de los monopolios, que como Microsoft Windows o Apple Inc. imponen una realidad ajena a los pueblos mediante sus estándares capitalistas.

En tal sentido, la presente investigación tiene como propósito definir una arquitectura que permita desarrollar un entorno gráfico orientado a multimedia, sobre el cual puedan implementarse diferentes metáforas visuales.

Palabras claves: arquitectura, entorno gráfico, software libre, multimedia

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Antecedentes del problema planteado en este trabajo.....	5
1.2 Interacción Persona-Ordenador	9
1.2.1 Estilos de interacción.....	9
1.2.2 Interfaz Gráfica de Usuario.....	14
1.2.3 Metáforas en la interacción persona-ordenador.....	14
1.3 Multimedia.....	19
1.4 Entornos Gráficos.....	21
1.5 El sistema de ventanas X.....	25
1.5.1 Elementos que conforman el sistema de ventanas X.....	25
1.5.2 Concepto funcional.....	34
1.5.3 Descripción del funcionamiento.....	35
1.5.4 Presente y futuro del sistema de ventanas X.....	37
1.5.5 Más allá del sistema de ventanas X.....	37
CAPÍTULO II. ARQUITECTURA PARA ENTORNO GRÁFICO ENFOCADO A MULTIMEDIA.....	40
2.1 Principios generales de la arquitectura.....	40
2.2 Herramientas, tecnologías y metodologías para el desarrollo.....	42
2.2.1 Lenguajes de programación.....	42
2.2.2 Tecnologías.....	45
2.2.3 Metodología de desarrollo.....	47
2.3 Descripción.....	49
2.3.1 Arquitectura cliente-servidor.....	49
2.3.2 Arquitectura en tres capas.....	51
2.4 Modelos arquitectónicos.....	53
2.4.1 Modelo físico.....	53
2.4.2 Modelo lógico.....	54
2.5 Sobre la arquitectura.....	55
CAPÍTULO III. IMPLEMENTACIÓN DEL GESTOR DE CONFIGURACIÓN DE	

INTERFACES TOCORORO.....	56
3.1 Valoración crítica del diseño propuesto por el analista.....	56
3.2 Planificación del proyecto por roles.....	59
3.3 Historias de usuario y prototipos de interfaz de usuario.....	59
3.4 Tareas de Ingeniería.....	64
3.5 Plan de Versiones.....	69
3.6 Sobre el desarrollo del GCIT.....	69
CONCLUSIONES GENERALES.....	70
RECOMENDACIONES.....	71
REFERENCIAS BIBLIOGRÁFICAS.....	72
BIBLIOGRAFÍA.....	74
Fuentes electrónicas.....	74
ANEXOS.....	79
Tablas.....	79
Figuras.....	80
GLOSARIO DE TÉRMINOS.....	83

INTRODUCCIÓN

Resulta imposible separar la historia reciente de los ordenadores de las interfaces gráficas de usuario, de los denominados entornos gráficos, que han permitido el uso masivo de estos dispositivos por personas de todas las edades y profesiones, con las más disímiles costumbres e idiosincrasia. Estas interfaces, que obedecen en su mayoría a estándares definidos hace ya más de 30 años, cuando la computadora personal era destinada principalmente a trabajos de oficina, no contaba con la difusión que hoy tiene, menos con los más recientes avances en materia de hardware, que permiten mayores y mejores prestaciones imponen la necesidad de nuevos paradigmas.

Ideas como los escritorios 3D, mejoran notablemente la metáfora visual actual dotándola de una creciente gama de efectos, consecuencia quizás de la industria cinematográfica, más no ofrecen novedad en el diseño, al punto de que promuevan un cambio en ella con el propósito de ajustarse a las características propias de los usuarios. No obstante la tecnología subyacente ofrece el medio propicio, para combinada con el concepto de multimedia construir un entorno con más posibilidades de interacción y usabilidad, que los actuales, pues este se aplica fundamentalmente a programas especializados, de manera que no constituye una característica que distinga propiamente a estos.

El uso de la multimedia en la informática es un fenómeno reciente, cuya importancia radica en su similitud con la forma en la que los humanos perciben la información del medio ambiente que les rodea, lo cual es el resultado de la combinación de diferentes medios en función de la comprensión y el entendimiento, siendo su fin el de dotar de mayor interactividad al ordenador y en consecuencia proporcionar una mayor usabilidad del software. En el caso de Cuba, esta se ha extendido hacia todos los centros educacionales del país, constituyendo un elemento imprescindible para la enseñanza como medio de aprendizaje, lo cual avala las potencialidades de esta tecnología, incluida junto a la realidad virtual y la realidad aumentada, en la hasta el momento última de las generaciones de la Interacción Persona-Ordenador.

Esta **situación problemática** evidencia la necesidad de desarrollar entonces un entorno gráfico enfocado a multimedia, que permita implementar diferentes metáforas visuales, mas no existen experiencias similares en el tema, que puedan darle solución a esta problemática, a lo que se une el hecho de que la mayoría de los entornos gráficos existentes no son libres, pues constituyen productos insignias de sistemas operativos propietarios como Mac OSX y Microsoft Windows, cuyos precios en el mercado son inaccesibles para países del Tercer Mundo como Cuba.

En tal sentido el interés del estado y el gobierno cubanos por el uso de tecnologías libres en función del desarrollo socio-económico del país, motivado por la necesidad de soberanía e independencia tecnológicas, ha conllevado al establecimiento de una política de migración, que involucra a todos los sectores de la sociedad.

Los beneficios para Cuba de la utilización del software libre en sus producciones son apreciables, especialmente para el desarrollo de la industria de los gráficos por ordenador en los sistemas operativos, debido a que como ninguna otra filosofía, ofrece la posibilidad de crear herramientas propias, a pesar del control ejercido por las multinacionales de la informática, cuyos estándares obedecen a modelos capitalistas.

Partiendo de esta realidad surge como **problema científico**: ¿Cómo desarrollar un entorno gráfico enfocado a multimedia, que permita implementar diferentes metáforas visuales?, siendo el **objeto de estudio** el proceso de interacción persona-ordenador en los entornos gráficos, y el proceso de interacción persona-ordenador en los entornos gráficos de software libre el **campo de acción**.

El **objetivo general** de la investigación: Diseñar una arquitectura para entorno gráfico enfocado a multimedia, que permita implementar diferentes metáforas visuales; del que se derivan los siguientes **objetivos específicos**:

- Determinar las características de los componentes que conforman los entornos gráficos actuales en software libre y sus relaciones.

- Diseñar arquitectura para un entorno gráfico enfocado a multimedia, que permita implementar diferentes metáforas visuales.
- Implementar el Gestor de Configuración de Interfaces Tocororo (GCIT).

Partiendo de lo anterior se formula entonces la siguiente **idea a defender**:

El diseño de una arquitectura enfocada a multimedia, que permita la implementación de diferentes metáforas visuales, proporcionaría un marco para el desarrollo de un entorno gráfico enfocado a multimedia que haría más efectiva la interacción persona-ordenador, mejorándose la usabilidad del software.

A fin de cumplir los objetivos antes mencionados se hace necesario acometer las siguientes **tareas**:

1. Realizar estudio de la arquitectura de los entornos gráficos existentes en software libre, que permita comprender el funcionamiento de estos.
2. Diseñar arquitectura para entorno gráfico enfocado a multimedia, que permita implementar diferentes metáforas visuales.
3. Implementar el Gestor de Configuración de Interfaces Tocororo (GCIT).

Los **métodos científicos** empleados son:

Métodos teóricos

- **Análisis Histórico-Lógico:** El método se emplea con el propósito de realizar un análisis del devenir histórico de la interacción persona-ordenador, haciendo énfasis en los entornos gráficos, que permita comprender las raíces del problema objeto de

investigación.

- **Analítico-Sintético:** El método tiene el objetivo de, por medio del análisis de los componentes de un sistema gráfico de software libre, establecer las características generales de su funcionamiento.
- **Inductivo-Deductivo:** Se utiliza con la finalidad de, partiendo del análisis de los elementos generales que caracterizan a la interacción persona-ordenador, llegar al conocimiento de los entornos gráficos.
- **Genético:** Permite deducir, mediante el análisis de los entornos gráficos de software libre, las características fundamentales de los entornos gráficos.

Capítulo 1. Fundamentación Teórica: En el capítulo se trata el tema de la interacción persona-ordenador y su vínculo con los entornos gráficos, así como se analizan las características funcionales de los componentes de software que conforman el sistema de ventanas X, el cual constituye el entorno gráfico de los ordenadores que operan con software libre. Se muestran finalmente las tendencias y retos futuros en el desarrollo del sistema antes mencionado.

Capítulo 2. Arquitectura para entorno gráfico enfocado a multimedia: Es objetivo de este capítulo diseñar la arquitectura que se propone para un entorno gráfico enfocado a multimedia, que permita implementar diferentes metáforas visuales.

Capítulo 3. Implementación del Gestor de Configuración de Interfaces Tocatoro: En este capítulo se verán los procesos propios de la fase de elaboración del Gestor de Configuración de Interfaces Tocatoro (GCIT), a fin de validar de manera indirecta la arquitectura propuesta.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

Es objetivo del presente capítulo exponer los antecedentes del problema planteado, así como los conceptos relacionados con la interacción persona-ordenador, tales como estilo de interacción, metáfora visual, multimedia y entorno gráfico, que permitan una mayor comprensión de este. Se hace además un análisis de la arquitectura del sistema de ventanas X, sistematizándose en su funcionamiento, tendencias en su desarrollo y futuro.

1.1 Antecedentes del problema planteado en este trabajo

Generalmente, el proceso por el cual se llega a la solución de un problema en apariencia simple, termina resolviendo además, interrogantes que de otro modo parecerían no estar relacionadas con este. Así el problema planteado en este trabajo fue el resultado de la necesidad de desarrollar un entorno diferente para los niños y adolescentes cubanos, que hiciera uso de los conceptos de multimedia, extendiéndose más tarde hacia cualquier grupo de usuarios.

Primeras ideas. Tocatoro GNU/Linux

Las primeras ideas de desarrollar una herramienta informática para niños y adolescentes con software libre, vinieron de la mano de la Comunidad de Software Libre de la Universidad de las Ciencias Informáticas, con la propuesta de creación de una distribución cubana para la educación primaria y secundaria con el nombre de Tocatoro GNU/Linux. Con este propósito fueron concebidos los primeros esbozos, como distribución base, entorno de escritorio y otras características particulares que tendría, haciéndose hincapié en la personalización, especialmente de las interfaces gráficas de usuario, uno de los puntos identificados en ese momento como claves.

A tales efectos se realizaron los primeros contactos y jornadas de trabajo con los miembros que integrarían el equipo inicial, llevándose la propuesta al Ministerio de Educación, que en aquel entonces se debatía entre qué distribución emplear para la migración a nivel de país. Luego de varios encuentros se optó finalmente por Nova,

distribución que se venía desarrollando en la Universidad y que contaba con una trayectoria y proyecciones a largo plazo.

Esta nueva realidad llevaría a la realización de cambios en el proyecto sin perder la esencia materializada en su concepto, de sus objetivos iniciales, que desembocaron en lo que hoy es Tocatoro-Desktop, y cuyas premisas se resumen en la siguiente idea:

“ Lo importante es darle al usuario la posibilidad de hacer, no encerrarlo en el esquema de lo que debe ser, cuando lo que debe ser solo él lo sabe, aunque las más de las veces no sepa explicarlo; el usuario siempre sabe lo que quiere.”

Tocatoro-Desktop

Con el propósito de no dejar morir el noble fin detrás de la creación de una distribución para la enseñanza como Tocatoro GNU/Linux, surge en el curso 2007-2008 la idea de desarrollar un metapaquete de software educativo y temas de interfaz gráfica de usuario para Nova con el nombre de Tocatoro-Desktop. El metapaquete al instalarse debía permitir el acceso a juegos didácticos, multimedia, etc., desde el menú de un entorno de escritorio como Gnome - el entorno nativo de Nova-, así como añadir o activar temas de interfaz, estableciendo estos de manera significativa u obligando a crear un grupo de soporte para su empaquetamiento sistemático, que lógicamente necesitaría de conocimientos informáticos no triviales.

Todas estas tareas, definen el inicio del desarrollo de una aplicación de configuración denominada “Gestor de Configuración de Interfaces Tocatoro” (GCIT), pues tanto el software como los temas de interfaz podían ser empaquetados y luego instalados de manera separada mediante herramientas existentes, encargándose aquella solamente de la selección de los temas a activar y de la configuración propiamente del menú.

A las funcionalidades anteriores fueron añadidas otras, como los diseñadores de interfaz para cada una de las interfaces gráficas de usuario, con el propósito de integrar en una sola aplicación todas las actividades afines.

Sobre la base del anterior precedente se concibe el modelo actual de Tocaroro-Desktop, el cual tiene entre sus ventajas sobre el modelo anterior las siguientes:

- La creación de temas de interfaz por el usuario y la activación de estos mediante la aplicación, lo cual posibilita que en principio los profesores de los centros educacionales puedan adecuar las interfaces a su realidad.
- La configuración del menú con el software deseado, válido también para el contexto anterior, pues cada centro tiene sus propias particularidades, así como cada alumno las suyas, que requieren ser atendidas a fin de hacer eficiente el sistema de enseñanza.
- La gestión de toda la configuración de interfaces gráficas en un solo lugar y no en herramientas separadas; los entornos de escritorio actuales poseen aplicaciones de configuración de interfaces gráficas de usuario, pero estas las más de las veces no guardan relación entre sí o simplemente no existen.

La concepción anterior marcaría sin dudas un punto de inflexión en el desarrollo subsiguiente de Tocaroro-Desktop, inclinando la balanza hacia las interfaces gráficas, en definitiva el tema más cercano a los niños y adolescentes, lo cual llevaría a una segunda idea:

¿Qué tal si las interfaces gráficas actuales no fueran las adecuadas, si no pudieran ajustarse a la realidad de los usuarios por más esfuerzo que se gaste, si el esquema no fuera el correcto?.

Pensar en niños y adolescentes requiere a todas luces una mentalidad distinta, requiere de ideas a las que no se está acostumbrado, pues cada uno de estos “príncipes enanos” es un mundo en germen, un mundo ajeno a las precondiciones similares de los adultos. La frase “Adecuar las interfaces a su realidad” señala de manera evidente el camino a seguir, aún cuando en el contexto en el que se emplea presenta un grupo de limitaciones

asociadas al propio esquema de las interfaces gráficas actuales. En el mundo de las TICs esta necesidad de adecuación a las características particulares de los usuarios se denomina desarrollo de software a la medida, algo que sin dudas distingue a los infantes de cualquier otro grupo social.

Con relación a lo anterior Graciela Alonso y Silvana Asteggiante, en su trabajo “Las metáforas en la interacción hombre-computador”, expresan:

“Si lo que se busca es alcanzar entonces una interfaz efectiva que logre ser consistente, intuitiva, estéticamente aceptable, clara y transparente, funcional, legible, y ante todo amigable -entre otras características- la metáfora deberá ser apropiada al tipo de usuario destinatario. Es decir, el lenguaje metafórico y el modelo mental que va a dotar de significado a los elementos de interacción en un programa de computación, deberá ser acorde a las destrezas de sus usuarios, a los fines para lo que éstos lo requieren, a las edades de los mismos, a su contexto cultural, etc., posibilitándoles entender y comprender el lenguaje.”[1]

Cambiar los esquemas.

Para Tocatoro-Desktop, así como también lo fue para Tocatoro GNU/Linux, aún cuando ninguna investigación conocida en Cuba se tenía para evaluar las necesidades de los niños y adolescentes, en definitiva usuarios que desde la perspectiva de sus premisas constituyen su centro de atención, era evidente de que el esquema de las interfaces gráficas, eran un elemento a modificar, pues su enfoque marcadamente dirigido a adultos, evoca ambientes que solo le son familiares a estos.

Desde el punto de vista técnico la idea de cambiar las interfaces gráficas existentes por otras más adecuadas al tipo de “usuario destinatario”, encontraría en el camino a su implementación un sinnúmero de dificultades asociadas a limitaciones y retos tecnológicos, que desembocarían, en la presente investigación, como el propio desarrollo del GCIT, cuyos problemas han servido como validación indirecta de la propuesta de este trabajo.

1.2 Interacción Persona-Ordenador

En términos generales, la Interacción Persona-Ordenador (IPO) o Human Computer Interaction (HCI por sus siglas en inglés), es la disciplina que estudia el intercambio de información entre personas y ordenadores. Esta se encarga del diseño, evaluación e implementación de los aparatos tecnológicos interactivos. El objetivo es que el intercambio sea más eficiente, para lo cual se vale de conocimientos asociados a otras ciencias como teoría de la comunicación, diseño gráfico e industrial, lingüística, ciencias sociales, psicología cognitiva y fisiología[2].

Dentro de la IPO, las interfaces gráficas de usuario y las metáforas visuales que las definen, juegan un papel fundamental, por cuanto constituyen el medio por el cual los usuarios se comunican con el ordenador, y caracterizan muchos de los diferentes estilos de interacción que pueden ocurrir entre el hombre y la máquina, estableciéndose por tanto una estrecha relación entre estos conceptos como se evidencia en la siguiente figura:

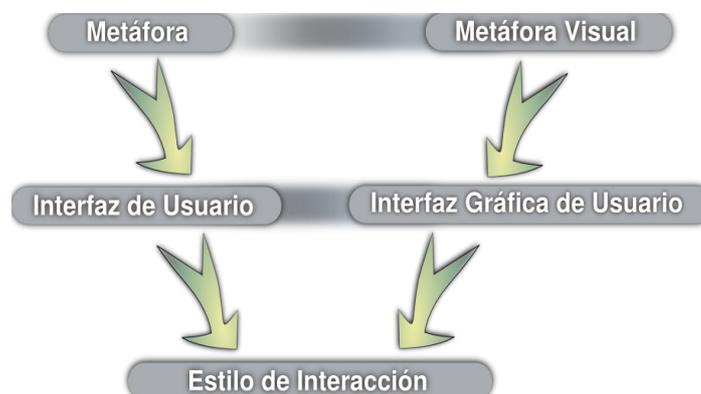


Figura 1. Relación entre metáfora visual, interfaz gráfica de usuario y estilo de interacción.

1.2.1 Estilos de interacción

Por estilo de interacción se entiende en informática un término genérico para agrupar las diferentes maneras en que los usuarios se comunican o interaccionan con el ordenador[3].

Los estilos de interacción predominantes según definen Jesús Lorés, Joan Manuel Gimeno y Ferran Pérdrix en el artículo usado como referencia para este apartado “*Metáforas, estilos y paradigmas*” [4], con los cuales además los autores concuerdan, pues aún cuando existen algunas diferencias asociativas que tienden a disminuir el número de estos en otras clasificaciones, constituyen esencialmente los mismos, son en orden de surgimiento:

- 1) La interfaz por línea de órdenes
- 2) Navegación por menús
- 3) Manipulación directa
- 4) Interacción asistida

Interfaz por línea de órdenes



Figura 2. La consola y su interfaz por línea de órdenes/comandos.

Es una manera de dar instrucciones directamente al ordenador. Pueden tener la forma de teclas de función, un carácter, abreviaciones cortas, palabras enteras o una combinación de las dos primeras, ofreciendo acceso directo a la funcionalidad del sistema.

La interfaz por línea de órdenes fue el primer estilo de interacción de uso generalizado y actualmente es muy utilizado, fundamentalmente en sistemas derivados de Unix[5].

Tabla I. Ventajas y desventajas de la Interfaz por línea de órdenes

Ventajas	Desventajas
<ul style="list-style-type: none"> ■ Flexibilidad ■ Permite la iniciativa del usuario ■ Es atractivo para usuarios expertos ■ Potencialmente rápido para tareas complejas ■ Capacidad para hacer macros 	<ul style="list-style-type: none"> ■ Requiere una difícil memorización ■ Requiere un entrenamiento importante ■ Gestión de errores pobre

Navegación por Menús



Figura 3. La interfaz del Midnight Commander, ejemplo de estilo de navegación por menús.

Un menú es un conjunto de opciones visualizadas en la pantalla, que suponen la ejecución de una orden subyacente y normalmente un cambio en el estado de la interfaz.

La estructura del menú de una aplicación se organiza de manera jerárquica. El número ideal de opciones en un menú está entre 3 y 8 [6].

Tabla II. Ventajas y desventajas de la Navegación por Menús

Ventajas	Desventajas
<ul style="list-style-type: none"> ■ No requiere memorización de sintaxis. ■ Es atractivo para usuarios noveles 	<ul style="list-style-type: none"> ■ Ocupa mucho espacio en la interfaz.

Manipulación directa

El término describe sistemas que tienen las siguientes características [7]:

- Representación continua de los objetos y de las acciones de interés.
- Acciones rápidas, incrementales y reversibles que provocan un efecto visible inmediatamente en el objeto seleccionado.
- Cambio de una sintaxis de comandos compleja por la manipulación de objetos y acciones.

Las pantallas gráficas de alta resolución y dispositivos como el ratón han permitido el desarrollo de los entornos de manipulación directa que crean una representación visual de objetos y acciones de interés. Estas ideas fueron aplicadas al ordenador Xerox Star y posteriormente al Apple Macintosh que fue el primer ordenador de manipulación directa que tuvo éxito comercial [8].

En la actualidad, el entorno más común de manipulación directa es WIMP (Windows, Icons, Menus and Pointers): Ventanas, Iconos, Menús y Apuntadores [9].



Figura 4. Entorno WIMP, característicos de sistemas como Mac OS X (en la figura), X Window System y Microsoft Windows.

Tabla III. Ventajas y desventajas de la Manipulación directa.

Ventajas	Desventajas
<ul style="list-style-type: none">■ Los usuarios noveles pueden aprender funcionalidades básicas rápidamente.■ Los usuarios experimentados pueden realizar una amplia gama de tareas■ Los usuarios pueden ver inmediatamente si sus acciones responden a sus objetivos.■ Los usuarios pueden memorizar los conceptos operacionales.■ Los usuarios tienen confianza porque al iniciar una acción, controlan y predicen la respuesta.■ Los usuarios tienen menos ansiedad porque el sistema es comprensible y las acciones son reversibles.	<ul style="list-style-type: none">■ No todas las tareas pueden ser descritas por objetos concretos■ No todas las acciones se pueden realizar directamente

Interacción asistida

El estilo de interacción dominante, el de manipulación directa, requiere que el usuario explicita todas las tareas y controle todos los eventos. Este estilo ha de cambiar si se quiere que los nuevos usuarios hagan un buen uso de los ordenadores y redes del futuro.

En tal sentido, actualmente comienza a utilizarse un nuevo estilo de interacción que se ha dado en llamar interacción asistida, que utiliza la metáfora del asistente personal o agente que colabora con el usuario en el mismo ambiente de trabajo **[10]**.



Figura 5. La imagen muestra al típico asistente humanoide de las películas de ciencia ficción.

Tabla IV. Ventajas y desventajas de la Interacción asistida.

Ventajas	Desventajas
<ul style="list-style-type: none">■ Los usuarios pueden aprender funcionalidades básicas rápidamente.■ Permite reducir el esfuerzo necesario para realizar tareas rutinarias.	<ul style="list-style-type: none">■ El usuario no tiene control sobre los procesos asistidos que ocurren en su ordenador.

1.2.2 Interfaz Gráfica de Usuario

La Interfaz Gráfica de Usuario o Graphical User Interface (GUI) es en el contexto de interacción persona-ordenador, según definición de la Wikipedia [11], el artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático. Es un tipo de interfaz de usuario que utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz, que surge como evolución de la interfaz por línea de comandos de los primeros sistemas operativos y es pieza fundamental en un entorno gráfico, como el “escritorio” del sistema operativo Windows, el sistema de ventanas X o Aqua de Mac OSX.

1.2.3 Metáforas en la interacción persona-ordenador

Aunque es un término propio de las ciencias sociales, empleado fundamentalmente por lingüistas, en informática se usa de manera muy similar para comunicar un concepto nuevo de una manera familiar y accesible a través de otro u otros conceptos, que se conocen mejor[12].

En el diseño de las interfaces gráficas actuales, las metáforas son esenciales, pues su uso permite conseguir maneras más eficientes y efectivas de desarrollar programas que puedan ser usados por comunidades de usuarios más diversas [13]. En tal sentido puede decirse que las metáforas establecen un esquema para las interfaces gráficas, determinándolas.

Metáforas visuales

Según se define en el artículo “Metáforas, estilos y paradigmas”[14], de los autores citados con anterioridad:

“La metáfora visual normalmente es una imagen, que permite representar alguna cosa y que el usuario puede reconocer lo que representa y por extensión puede comprender el significado de la funcionalidad que recubre” .

Su importancia está dada por el hecho de que, al decir de Graciela Alonso y Silvana Asteggiante [15]:

“Es mediante “proyecciones metafóricas de los esquemas de las imágenes” que comprendemos experiencias peor conocidas por no estar claramente delineadas (emociones, ideas, el tiempo) en función de otros conceptos que entendemos con mayor claridad por estar más cercanos a nuestra experiencia corporal y en consecuencia, categorizarlos más adecuadamente mediante nuestros esquemas corporeizados (orientaciones espaciales, objetos, etc.).”

La idea expuesta refiere que los esquemas mentales del mundo que perciben los humanos se traducen en esquemas de imágenes, los cuales están más cercanos a las experiencias corporales de estos como resultado de ser, algo que no expresan las citadas autoras, pero que prueban estudios recientes sobre el tema, seres mayormente visuales.

Metáfora Global y Metáfora Compuesta

La metáfora global es una metáfora que proporciona el marco para las demás metáforas del sistema, aunque muchas veces para extender su funcionalidad se recurre a la combinación de metáforas no asociadas con esta, dando lugar al término de metáfora compuesta. Ejemplo de metáfora global es el escritorio sin metáforas adicionales como la barra de desplazamiento o la de cortar y pegar, que en la actualidad lo convierten en una metáfora compuesta [16].

Evolución de las metáforas visuales en el proceso de desarrollo tecnológico de la computación.

La generación de metáforas visuales en el marco de los estudios de las interfaces persona-ordenador, acompaña las etapas históricas por las que ha transitado el desarrollo tecnológico en el campo de la computación y ha sufrido una evolución marcada por las distintas posibilidades técnicas a las que el hombre ha llegado en la constante búsqueda por mejorar su interacción con los ordenadores.

Primera generación (1950-1960)

La interacción persona-computador era casi inexistente, pues los ordenadores trabajaban en procesamiento por lotes cuyas entradas se realizaban a través de tarjetas perforadas con salidas impresas.[17]

Segunda generación (1960-1980)

Surgida dentro del contexto de la era del “tiempo compartido” llevado a cabo sobre mainframes y minicomputadores, establecía una interacción persona-ordenador a través de la confección semántica y sintáctica de comandos con parámetros.

La metáfora que subyace en este tipo de interacción es la del “diálogo escrito”, por la cual se establece una conversación en la que el hombre y la computadora se preguntan y se responden a través de la sintaxis de comandos. No es por esto, una interfaz amigable, sobre todo para los usuarios inexpertos o novatos.[18]

Tercera generación (1970-)

Esta etapa de desarrollo del ambiente gráfico enriqueció notablemente la interacción hombre-computador en tanto pasó de la metáfora del “diálogo escrito” basado en la pura lingüística semántico-sintáctica, a un modelo de “diálogo no lingüístico” basado en la potencialidad de los símbolos gráficos.

De todos los ejemplos de metáforas que habría para abordar, tiene especial significación por su trascendencia, impacto, y popularidad, la interfaz gráfica construida por la Xerox

Star, sobre la base de la metáfora de un escritorio u oficina con todas las funciones de administración de los documentos que allí se pueden efectuar.

La interfaz gráfica también introdujo la posibilidad de manipular objetos en forma visual, a partir de la simulación de objetos reales con la integración al mismo tiempo de texto, imágenes y sonido, etc., dando paso a lo que hoy se conoce como multimedia [19].

Cuarta generación (1990-)

Coexistente con la etapa anterior, la cuarta generación apunta a no usar menús, ni barras de herramientas, comenzando a trabajar sobre el hipertexto y la multimedia, la realidad virtual, las interfaces post-WIMP, las interfaces adaptativas y la manipulación indirecta a través de agentes.

Esta etapa responde a la búsqueda de una interacción, que procura posibilitar o concretar la metáfora de la “computadora humana”. La proyección metafórica apunta a un diálogo directo, sin interfaz, donde a través del reconocimiento de gestos o de la voz, y a través de agentes inteligentes, el hombre pueda directamente conversar e interactuar con el computador como la haría con cualquier otra persona [20].

Metáforas visuales actuales

El escritorio

La metáfora del escritorio fue la primera metáfora global y a los días de hoy continúa siendo utilizada prácticamente en todas las interfaces gráficas.

En sus inicios el uso de los ordenadores estaba dirigido a los oficinistas, por tanto esta metáfora correspondería exactamente al entorno en el que la mayor parte de los usuarios trabajaban. En la actualidad, este modelo ha ido perdiendo vigencia, toda vez que los ordenadores constituyen un elemento cotidiano, tan familiar para las personas como lo fueron en su tiempo dispositivos como el radio o el grabador de sonido, por lo que necesita ser sustituido por otro más eficiente, que se ajuste a las características propias de los usuarios.

La ventana

Otra de las metáforas visuales que se utiliza hoy de manera generalizada es la ventana, la cual fue introducida inicialmente por Xerox, antes incluso de que lo hiciera con el concepto de escritorio. La ventana forma parte de la mayoría, por no decir de todos, los sistemas gráficos conocidos en el presente, a los que con razón se les denomina sistemas de ventanas o windowing systems. Un análisis de esta metáfora permite hacer las siguientes acotaciones sobre su funcionamiento, señalándose algunas de sus evidentes deficiencias:

En el mundo real las ventanas son aberturas en las paredes a través de las cuales puede apreciarse el medio ambiente detrás de ellas, difícilmente permiten acceder a lo que se observa, pues existe una pared de por medio, y cuando se hace, se corre el riesgo de tener que abandonarla. En la práctica esto no funciona de la misma manera en los escritorios, donde lo que se denomina "ventana" en realidad es una especie de contenedor de elementos visuales que puede tratarse de manera independiente de los demás, tal cual si se tratara de cualquier objeto de la cotidianidad y no un espacio a través del cual acceder a solo mirar. Por otro lado lo que se observa a través de una ventana no se mueve con ella si es que las ventanas pueden moverse.

La conclusión más obvia es que en verdad lo que se tiene en el escritorio no son ventanas, sino objetos que con los nuevos efectos visuales pueden tener un índice de plasticidad o refracción determinados, moverse, rotarse, cambiar de forma y apariencia, etc, es decir objetos que pueden manipularse. Esta idea reveladora, hace por tanto que el término ventana parezca inadecuado para caracterizar nuestra manera de sentir lo que nos rodea, no es por tanto una buena metáfora, en cambio el concepto abstracto de objeto se ajusta más al significado que concretamente se percibe.

La idea fundamental detrás de la ventana está dada por el hecho de que estas recortan la realidad detrás de ellas, como los marcos en la web, que recortan su contenido a fin de ajustarse a un espacio variable determinado, pero esto no justifica el que puedan moverse o se deformen, se realicen efectos que modifican su comportamiento. La razón del

problema es el resultado de la definición de la metáfora en un momento en el que los gráficos por ordenador eran solo bidimensionales, la tecnología del microprocesador recién se iniciaba, nada se sabía de tarjetas gráficas o de la potencia con que hoy cuentan estos dispositivos.

1.3 Multimedia

Multimedia, según define la Wikipedia [21] es un término que en informática se aplica a cualquier objeto que usa simultáneamente diferentes formas de contenido informativo como texto, sonido, imágenes, animación y vídeo. Es una tecnología digital de comunicación, constituida por la suma de hardware y software, que tiene como fin humanizar la máquina.

Existen varios tipos de información multimedia, como:

- Texto: sin formatear, formateado, lineal e hipertexto.
- Gráficos: utilizados para representar esquemas, planos, dibujos, etc.
- Imágenes: son documentos formados por píxeles, tienden a ser ficheros muy voluminosos.
- Animación: presentación de un número de gráficos por segundo que genera en el observador la sensación de movimiento.
- Vídeo: presentación de un número de imágenes por segundo, que crean en el observador la sensación de movimiento. Pueden ser sintetizadas o captadas.
- Sonido: puede ser habla, música u otros.

En el contexto de la interacción persona-ordenador, la multimedia abarca conceptualmente casi todos los objetivos de la disciplina de las IPO, por cuanto en ella convergen e interactúan diferentes tipos de información provenientes de los medios que intervienen en el diálogo del hombre con la máquina.

La multimedia puede ser clasificada de acuerdo a la linealidad de la interacción que

establece con los usuarios en multimedia interactiva y multimedia no interactiva.

Multimedia no interactiva

Por multimedia no interactiva se entiende aquella, que no permite la manipulación por el usuario de los componentes que contiene, como el vídeo por ejemplo, donde a pesar de integrar varios tipos de información simultáneamente como texto, animación, gráficos y sonido, estos no pueden ser accedidos de manera independiente [22].

Multimedia interactiva

Se habla de multimedia interactiva cuando el usuario tiene cierto control sobre la presentación del contenido. La hipermedia podría considerarse como una forma especial de multimedia interactiva que emplea estructuras de navegación más complejas a fin de aumentar el control del usuario sobre el flujo de la información [23]. La realidad virtual es quizás la expresión máxima de la multimedia interactiva, por cuanto se auxilia de múltiples dispositivos que cubren prácticamente todos los sentidos a fin de lograr niveles de interactividad superior.

La multimedia y la comprensión humana

Cuando un programa de computador, un documento o una presentación combina adecuadamente los medios, se mejora notablemente la atención, la comprensión y el aprendizaje, pues se acercará más a la manera habitual en la que los seres humanos se comunican, cuando se emplean varios sentidos para comprender un mismo objeto o concepto [24].

La multimedia encuentra su uso prácticamente en todas las áreas del saber y el actuar humanos, como arte, educación, entretenimiento, ingeniería, medicina, matemáticas, negocio y la investigación científica, principalmente la relacionada con la simulación y los ambientes de realidad virtual.

La experiencia cubana al respecto avala la calidad de esta tecnología, difundida masivamente en centros educacionales, académicos y científicos.

1.4 Entornos Gráficos

Aún cuando es un concepto ampliamente utilizado en medios como la propia Wikipedia, foros de discusión y debate sobre el tema, entre otros, no existe una definición del mismo, salvo de manera indirecta por medio de conceptos que constituyen una clasificación del mismo, como lo son los entornos de escritorio. Es así como basados en su experiencia en el tema, los autores del presente trabajo proponen la siguiente definición para entorno gráfico:

Un entorno gráfico es una solución completa de interfaz gráfica de usuario GUI, que permite la interacción persona-ordenador mediante el uso de metáforas visuales, las cuales abstraen el comportamiento interno de la computadora: estas metáforas son las responsables de definir conceptualmente el entorno para el usuario, siendo el escritorio la más extendida de todas. Actualmente los entornos mejor conocidos son los basados en las interfaces gráficas de Microsoft Windows, Mac OSX y X Window System.

Microsoft Windows

Aún cuando constituyen las interfaces gráficas más familiares para los usuarios de ordenador en el mundo, resultado de las inmensas campañas de marketing que organiza la compañía homónima, el hecho de ser software privativo ha propiciado la migración de Cuba hacia tecnologías libres, a fin de garantizar su desarrollo. A lo anterior se une la falta de iniciativa y novedad en el diseño de dichas interfaces, que ha llevado a Microsoft a copiar ideas de otros sistemas como Compiz en X Window System o Aqua en Mac OSX (ver en los **Anexos, Figura 1**).

Mac OSX

Para muchos uno de los sistemas más potentes y estables cuya interfaz gráfica goza del más alto reconocimiento de especialistas de todo el planeta, salvo por el hecho de también ser software privativo cuyos costos en el mercado son inaccesibles para la mayoría de los usuarios. A Mac OSX se le debe particularmente el desarrollo de la

tecnología de los administradores de composición y efectos, que hoy lleva a los entornos de escritorio de la mano del 3D (ver en los **Anexos, Figura 2**).

Sistema de Ventanas X

El sistema de ventanas X, o más comúnmente X Window System, es el más extendido entre los usuarios de sistemas Unix por los beneficios que reporta su estructura basada en el modelo cliente servidor, y la flexibilidad que ofrece para el desarrollo de entornos de escritorio, los cuales se pueden implementar sobre diversas GUIs. Debido a que X es un sistema libre, constituye para Cuba una alternativa a Microsoft, frente a las prerrogativas del bloqueo y a la crisis del capital que actualmente sacude el orbe. Tecnologías como Compiz Fusion y entornos de escritorio como KDE, Gnome y Xfce, ampliamente conocidos dentro y fuera del ámbito del software libre, son resultado de este potente sistema (ver en los **Anexos, Figura 3**), que no obstante presenta algunas grietas como más adelante se verá .

Tendencias de los entornos gráficos

Para nadie es un secreto de que la industria de los gráficos por ordenador ha evolucionado en estos años hacia el espacio 3D, en consonancia con las nuevas exigencias de la industria cinematográfica y otras ramas asociadas a la realidad virtual. Por todos es sabido de que los entornos gráficos se han movido también en esta línea motivados por mejorar la usabilidad de las metáforas existentes para el diseño de interfaces gráficas de usuario, que con el tiempo habían ido perdiendo en consistencia, elegancia y funcionalidad. Apple fue la primera compañía en darse cuenta de ello, y aprovechó la oportunidad para lanzar su conocido entorno Aqua y con él iniciar la era de la composición en los escritorios, apoyados principalmente en OpenGL, API ampliamente soportado por los fabricantes de tarjetas de vídeo.

Debido a las limitaciones de su arquitectura inicial no enfocada a OpenGL, en el sistema de ventanas X se implementaron alternativas más costosas que permitieran su uso por las aplicaciones, naciendo tecnologías como Mesa3D ,DRI, GLX y Xgl que darían paso a la creación de Compiz, Beryl y recientemente Compiz Fusion. En la actualidad (2008) la

composición 3D mediante AIGLX en el servidor X es una realidad.



Figura 6. El cubo, uno de los efectos implementados en Compiz Fusion

Actualmente proyectos como KDE investigan seriamente la rama del 3D, en tanto que Gnome y Xfce se han quedado a la saga, lo cual quizás marque la diferencia en un futuro entre estos entornos de escritorio.

En el caso de Microsoft Windows, su interfaz Aero, aunque mejor que las anteriores exhibidas por el sistema al incorporar efectos como la transparencia o las ventanas 3D sin antialiasing mediante la composición, continúa hasta hoy siendo deficiente en el manejo de los recursos de los ordenadores de bajas prestaciones.

No obstante los nuevos horizontes que se abren con la llegada a los ordenadores domésticos de la tecnología tridimensional, lo cierto es que pocos se han proyectado todavía por cambiar de manera seria la metáfora vigente del escritorio, incluso es poco probable de que alguno de los sistemas señalados lo haga en un futuro a mediano plazo existiendo un número limitado de instituciones que en el mundo se dedican a investigar sobre este tema de vital importancia para la usabilidad del software, sobre todo si se tiene en cuenta que alrededor del 45% del código de un programa está dedicado a su interfaz gráfica.

Multimedia y entornos gráficos

Debido a que la multimedia integra animaciones, imágenes y texto, entre otros, hablar de entornos gráficos es tratar un tema íntimamente ligado a su concepto y lo que ella representa. A saber más, que los entornos gráficos constituyen el escenario donde convergen los diferentes contenidos multimedia, aunque la verdadera integración ocurre en aplicaciones especialmente diseñadas con este fin como Director, propiedad de la compañía Adobe, entre otros, no existiendo en software libre todavía una herramienta que permita realizar esta tarea.

¿Por qué no hacer entonces un entorno gráfico basado en el modelo de los software especializados en la integración de contenido multimedia? ¿Cuáles son los problemas tecnológicos asociados a esta idea?.

En los entornos gráficos de hoy la integración de los contenidos multimedia ocurre de manera incipiente, con animaciones que distan mucho de las posibilidades reales, mientras que el uso del sonido, se reserva solo para eventos importantes como el inicio o la terminación de sesión, sin contar la precariedad de las metáforas.

Tecnologías recientemente incorporadas como el 3D, constituyen uno de los esfuerzos por hacer del escritorio algo más que un ente estático, se proyectan por la manipulación de los componentes que lo integran, añadiendo efectos que lo dinamizan, pero no van más allá de eso.

A saber más, que el principal problema de los entornos gráficos de hoy es su arquitectura, que arrastra todavía el lastre de los escenarios bidimensionales, mientras la tecnología 3D se emplea de manera ocasional para realizar efectos que de otro modo no podrían lograrse.

Un análisis del sistema de ventanas X, permite arrojar más elementos sobre lo expresado anteriormente, así como comprender mejor el funcionamiento de los entornos gráficos de hoy.

1.5 El sistema de ventanas X

El sistema de ventanas X es un sistema de mapa de bits basado en el protocolo de red X11, que fue concebido para dotar de interfaz gráfica a los sistemas operativos Unix, y permite a los usuarios controlar todas las sesiones desde una sola pantalla, con aplicaciones ejecutándose bien en una ventana o en terminales virtuales separadas. El sistema, basado en la arquitectura Cliente-Servidor, y en un modelo de capas que posibilita su mantenimiento y le proporciona robustez, ofrece la capacidad de gestionar tanto ventanas remotas mediante TCP/IP, como locales, a través de zócalos de BSD (Berkeley Software Distribution), haciendo posible además el concepto de transparencia de la red.

1.5.1 Elementos que conforman el sistema de ventanas X

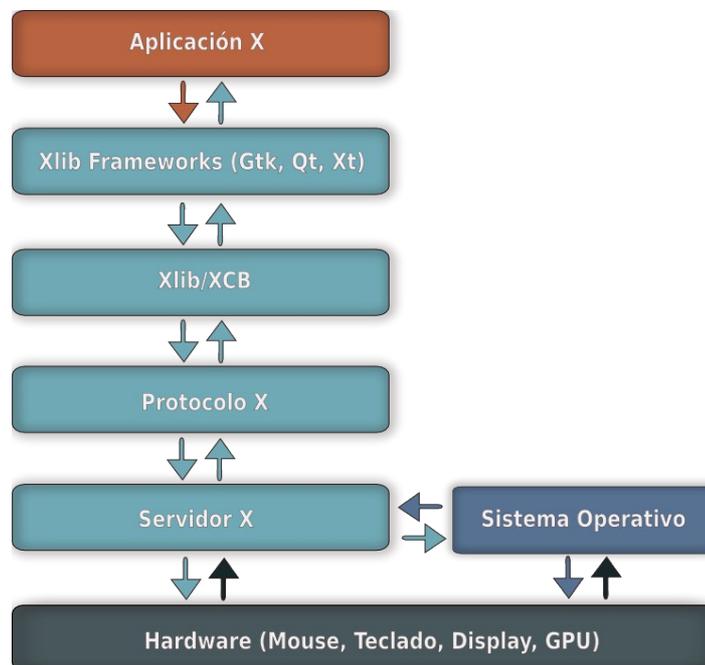


Figura 7. Arquitectura del sistema de ventanas X

Protocolo X

El Protocolo X es la entidad encargada de establecer el formato de los mensajes intercambiados entre los clientes y el servidor, y está orientado a conexión mediante TCP/IP. Utiliza un bloque simple de protocolo sobre la capa de transporte formateados en PostScript, en consonancia con la tecnología DPS (Display PostScript) que emplea X para la visualización en pantalla, la cual hace que la información gráfica sea solo bidimensional. La comunicación entre el cliente y el servidor se realiza usando cuatro clases de mensajes: request, reply, event y error. Estos mensajes son procesados por llamadas a Xlib, por lo que el programador no necesita usar el Protocolo X directamente.

Request

Un request es un mensaje de petición de servicio del cliente al servidor. Si necesita una respuesta, el cliente espera la contestación del servidor, si no continúa su tarea. La mayoría de las llamadas de la Xlib envían este tipo de mensajes. Las peticiones se procesan en el orden en que se envían, almacenándose normalmente en el buffer de la Xlib. El buffer se vacía y las peticiones se envían al servidor sólo cuando se satisfacen ciertas condiciones, por ejemplo, que el buffer esté lleno o que se requiera un reply (respuesta) del servidor.

Reply

Un reply lo envía el servidor al cliente como respuesta a una petición de este. El cliente permanece a la espera hasta que llegue el reply. El reply es el único mensaje síncrono de todo el protocolo. Algunas de las peticiones de la Xlib requieren información del servidor, como por ejemplo el tamaño y posición de una ventana o de una fuente, etc., en estos casos, la Xlib vacía el buffer, envía la petición al servidor y se queda esperando hasta que reciba la respuesta.

Event

Un event es un mensaje asíncrono que envía el servidor al cliente cuando ocurre un suceso en la estación de trabajo, que el servidor gestiona. Existen muchos eventos que el servidor detecta, por ejemplo, que se pulsa o se suelta el botón del mouse, o que se pulsa

una tecla. El cliente no puede detectar estos eventos directamente, solo el servidor puede hacerlo. El servidor envía mensajes event a los clientes para notificar los eventos. Los eventos no solo pueden ser generados por hardware, sino también por software, como el mover una ventana o cambiar su tamaño. El servidor no envía todos los eventos a los clientes, sino solo aquellos que los clientes especifican previamente; sin esa restricción, los clientes estarían demasiado ocupados procesando los eventos.

Error

Un error lo envía el servidor al cliente cuando detecta una situación de este tipo.

Servidor X

El Servidor X o comúnmente display, es un programa que tiene como propósito ofrecer servicios de display en una terminal gráfica a pedido de los clientes, controlando la pantalla, el mouse y el teclado para uno o varios de ellos, lo cual lo hace responsable, por ejemplo del mapeado de colores, fuentes, así como del teclado, entre otras funciones asociadas a este y otros dispositivos.

Normalmente los servidores X se ejecutan en terminales gráficas de alto rendimiento diseñadas específicamente para realizar dicha tarea, y pueden estar ubicados en cualquier lugar de la red, o incluso en el mismo ordenador de los clientes que solicitan sus servicios.

Como todo servidor, el Servidor X dispone de una serie de recursos determinados por un identificador, a los cuales acceden los clientes, de manera que estos no necesitan almacenarlos, ni enviarlos o recibirlos de aquel. A saber dichos recursos son: ventanas, contextos gráficos(GC), pixmaps, cursores y fuentes de textos.

Ventana

Una ventana es un área rectangular que se muestra en la pantalla, aunque no directamente, pues necesita ser mapeada antes. Se puede superponer una ventana sobre otra o cambiar el orden de apilamiento, dibujar gráficos sobre ellas(mapeado), obtener entradas del teclado o eventos del mouse.

Pixmap

Los pixmaps son recursos de almacenamiento de gráficos, de manera que no se visualizan directamente en la pantalla hasta después de ser mapeados. Se pueden mapear pixmaps total o parcialmente sobre las ventanas, incluso repetirlos como patrones en polígonos y otras figuras planas.

Contexto Gráfico (GC)

Un contexto gráfico es un área donde se almacenan los atributos de los gráficos, como por ejemplo el ancho y estilo de una línea o su color. Cuando se dibujan gráficos no es necesario enviar sus atributos cada vez, solo se necesita especificar qué contexto gráfico de los que se encuentran en el servidor se quiere usar.

Cursor

Un cursor es un símbolo que muestra la posición del puntero; es en el contexto del Servidor X un recurso donde se almacenan las propiedades de estos, tales como tamaño, forma y color.

Fuente

Una fuente es un recurso que almacena la información de cada fuente de texto o tipo de letra, y a la que comúnmente se identifica como fuente del Sistema.

Pantalla

La pantalla (en inglés screen), es el hardware donde los datos gráficos se vuelcan. No es sinónimo de display. De hecho, un mismo servidor puede controlar múltiples pantallas.

Cliente X

Se denomina Cliente X a cada una de las aplicaciones que se relacionan directamente con los usuarios, permitiéndole a estos comunicarse con el ordenador mediante el empleo de una interfaz gráfica como su salida en el display.

Para acceder al dispositivo de salida gráfico, los clientes X utilizan librerías como Xlib o wrappers de esta última como Gtk, Qt o Xt, entre otras, dotadas de un conjunto de widgets, que posibilitan su diseño.

Xlib

Xlib es una librería que contiene una interfaz de programación de aplicaciones (API) basada en una colección de subrutinas primitivas de C embebidas en todos los clientes X. Esta constituye el nivel más bajo de software que controla directamente el protocolo X, y sobre la cual se desarrollan las aplicaciones de los usuarios, generalmente a través de alguna extensión de la librería que permita funciones más complejas.

Los procedimientos en Xlib traducen las peticiones de los clientes a solicitudes del protocolo X y analizan los mensajes que llegan (eventos, respuestas, errores) de los servidores X, además de suministrar diversas utilidades adicionales como la realización de operaciones independientes del sistema operativo (gestión de almacenamiento, etc.).

Según la bibliografía consultada, varios son los problemas técnicos asociados actualmente a Xlib, entre los cuales pueden citarse los siguientes:

- Presenta un código extenso que complejiza el procesamiento de la información.
- Si bien soporta programación multihilos, no es menos cierto que el API hace compleja y propensa a errores la tarea.
- Las solicitudes requieren una respuesta efectiva síncrona, bloqueándose mientras la respuesta llega, aún cuando el resultado se necesite de inmediato.
- Hace una caché con las peticiones de los clientes, entre otras optimizaciones similares, que hacen difícil la gestión de los procesos de solicitudes y respuestas específicas.

XCB

Con el propósito de resolver las dificultades de Xlib anteriormente relatadas, surge en el año 2001 XCB (X C Binding). XCB es un binding de C para el sistema de ventanas X, que tiene como propósito reemplazar a Xlib y cuya primera versión fue publicada en el año

2007. Es una librería de desarrollo que posee compatibilidad a nivel de interfaz binaria para Xlib, pues usa su misma capa de protocolo en tanto sustituye la capa de transporte por una propia, dándole soporte a los wrappers que la recubren.

Los objetivos de XCB son:

- Reducir el tamaño y la complejidad de la librería Xlib.
- Permitir el acceso directo al protocolo X.
- Replantear las descripciones del protocolo X.
- Documentar el protocolo X.
- Permitir bindings para otros lenguajes.
- Hacer una interfaz C asíncrona, que facilite la programación multihilo, así como la implementación de extensiones por medio de descripciones XML del protocolo X.

Lo interesante de esta nueva librería, actualmente en uso es su proyección hacia estándares internacionales para el intercambio de información como XML y XSLT, que proporciona una mayor flexibilidad del sistema de ventanas X y del protocolo X.

Administrador de Ventanas

El Administrador de Ventanas, es un cliente X localizado en la estación de trabajo donde se ejecuta el servidor. Dado que muchos clientes X compiten por los servicios de un Servidor X, el administrador de ventanas -entidad separada y diferenciada-, resuelve los conflictos producidos por esta competencia, siendo esta su función fundamental; el administrador no es necesario para crear las ventanas, aunque permite redimensionarlas, moverlas y modificarlas cuando sea solicitado por los clientes, proporcionando una vista gráfica global en la salida del display.

Diferentes administradores de ventanas pueden tener políticas distintas, de ahí la diversidad de administradores existentes, algunos de los cuales se han ganado un lugar entre los usuarios por su usabilidad y estética, sin descartar el hecho de que forman parte de proyectos de entornos de escritorio reconocidos como XFCE, GNOME o KDE.

Existen varios tipos de administradores de ventanas: Administradores de Ventanas de Mosaico, Administradores de Ventanas de Pila (los tradicionales) y Administradores de Composición de Ventanas.

Administrador de Ventanas de Mosaico

Los administradores de ventanas de mosaico o tiling window managers posicionan todas las ventanas sin solaparse. Un ejemplo de esto lo podemos ver en Blender, aplicación de diseño donde las ventanas aparecen una a continuación de la otra, afectando sus dimensiones, lo cual es su principal desventaja.

Administrador de Ventanas de Pila

Un administrador de ventanas de pila o stacked window manager, es aquel en el que las ventanas aparecen superpuestas, pero no permite ver a través de las ventanas visibles las ventanas ocultas por ellas, limitando el diseño de interfaces gráficas a mapeados con índice de refracción nulo, lo que impide efectos de opacidad y distorsión propios de muchos objetos del mundo real como las fibras de vidrio, por ejemplo. Estos son los administradores más comunes, una lista resumida de los cuales aparece en la **Tabla VII** en los Anexos.

Administrador de Composición de Ventanas

Un administrador de composición de ventanas es un administrador de ventanas y un administrador de composición al mismo tiempo, que a diferencia de un administrador de pila no dibuja directamente sobre el display, sino que procesa la salida antes en un buffer invisible y luego proyecta el resultado en aquel.

Administrador de Composición

El administrador de composición es el software que se encarga de componer las ventanas y realizar efectos sobre ellas antes de ser mostradas por el administrador de ventanas en la pantalla.

Existen dos tipos esenciales de administradores de composición: administradores de

composición 2D y administradores de composición 3D, los cuales se caracterizan por operar sobre el contenido gráfico propio de estas dimensiones.

La técnica de composición se vale de la idea del doble buffer(DBE), que implementa OpenGL para la aceleración gráfica. El empleo del doble buffer permite redireccionar la información de los píxeles 2D/3D de las ventanas a la memoria del host o a mapas de píxeles fuera del buffer frontal, y con ello copiar dicha información a este cuando se necesite para actualizar lo que ve el usuario como su escritorio. A tales efectos el administrador de composición es el encargado de controlar cuándo se redireccionan las ventanas y cómo se copia la información de los píxeles en el buffer de la pantalla (render).

La composición es una tecnología que permite efectos visuales avanzados, como la opacidad (ver **Figura 8**), desvanecimiento, escalamiento, duplicación, deformación, y redirección de aplicaciones. La incorporación de una tercera dimensión virtual permite nuevas características como sombreados reales detrás de las ventanas, la apariencia de distancia y profundidad, versiones miniaturas de las ventanas en tiempo real y animaciones complejas.

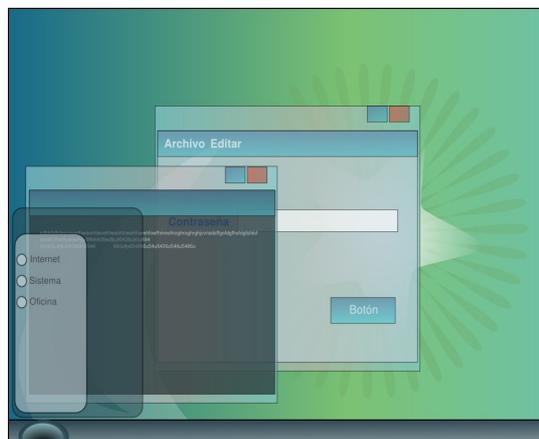


Figura 8. La opacidad, uno de los efectos que pueden conseguirse con un administrador de composición.

Problemas asociados a la no composición

Un análisis como el que se muestra a continuación, permite apreciar las desventajas para los usuarios, que se derivan de trabajar directamente sobre el buffer frontal. De manera general son las siguientes:

- Las aplicaciones constantemente tienen que estar redibujando las ventanas aunque no hayan sufrido cambios.
- Si la aplicación no responde, lo único que se verá será un rectángulo vacío.
- Si la aplicación se demora en redibujar, se verá un retraso en el despliegue.
- Una ventana que quiera utilizar áreas no rectangulares o transparentes, no tendrá control de lo que está dibujado detrás de ella, de manera que no se mapeará correctamente.

Al problema del renderizado directo sobre el buffer frontal se une la naturaleza estática de los estados de las ventanas, y el hecho de que estos estados son o bien instantáneos, o bien tienen animaciones de transición primitivas. Un ejemplo de lo anterior puede apreciarse al maximizar y minimizar ventanas, donde estas aparecen y desaparecen instantáneamente, o se muestran sencillas siluetas que se utilizan en una animación donde la ventana se reduce has el icono.

Composición 2D en el Sistema de Ventanas X

En el sistema de ventanas X, la composición 2D fue la primera en implementarse mediante la extensión composite, debido a que su arquitectura base solo manejaba, y aún solo maneja información bidimensional.

La composición 2D, a diferencia de la composición 3D, permite efectos menos realistas, razón por la que su desarrollo sería relegado a un segundo plano, aún cuando consume menos recursos hardware que esta. Administradores de ventanas como Metacity, Xfwm y Kwin fueron los primeros, y todavía son los únicos en contar con un administrador de composición 2D.

Composición 3D en el Sistema de Ventanas X

La composición 3D en el sistema de ventanas X tomó varias alternativas en función de aprovechar las potencialidades del estándar OpenGL. La primera de estas variantes fue implementar un servidor X nuevo denominado Xgl, mediante el uso del protocolo GLX, DRI y la librerías Mesa3D y glitz. El segundo camino a seguir estuvo enfocado en adaptar las especificaciones de OpenGL a X mediante el desarrollo de AIGLX (Aceleración Indirecta del protocolo GLX), siendo además el que actualmente se encuentra en desarrollo como parte del proyecto X.Org Server.

1.5.2 Concepto funcional

Existen dos partes en el Sistema de Ventanas X, que se comunican entre sí:

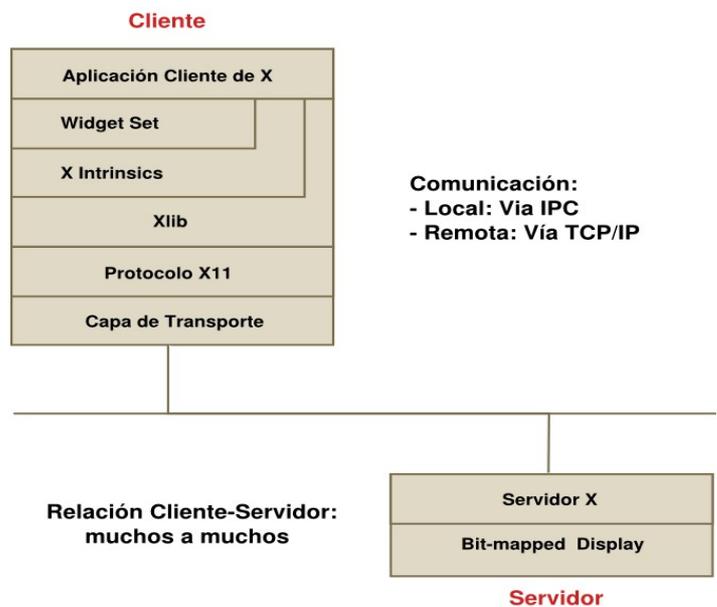


Figura 9. Concepto del Sistema de Ventanas X

1. La aplicación o cliente X, que recibe entradas del usuario, ejecuta código y envía su salida al display, empleando Xlib para transmitir y recibir datos de/a la terminal de usuario.

2. La terminal de usuario, que ejecuta la interfaz con el display y envía datos al servidor o los recibe de este.

1.5.3 Descripción del funcionamiento

1. Los servidores y clientes X pueden o no estar en el mismo host, comunicándose o bien mediante IPC (comunicación entre procesos) a través de zócalos, o usando TCP/IP por medio de la red respectivamente.
2. Sólo hay un servidor X por terminal con el que se pueden comunicar varios clientes X. El deber del servidor X es mostrar las ventanas de aplicación y enviar al cliente X correspondiente las entradas del usuario.
3. Depende del cliente X el mantener las ventanas que ha creado. Los cambios efectuados en el display por otros clientes son notificados por eventos del servidor X. Sin embargo, los clientes no tienen que preocuparse de qué parte sus ventanas son visibles y cuáles no, o de cuándo se deben dibujar o redibujar estas.
4. El servidor X contabiliza las ventanas visibles y no visibles a través de pilas. Una pila contiene los "hijos de primera generación" de una ventana padre. Una ventana hijo puede ser padre si tiene a su vez una o más ventanas hijas, que de nuevo se almacenarán en una subpila. La pila primaria es la que guarda todas las ventanas localizadas directamente bajo la raíz (Ver **Figura 9**). Las subventanas sólo pueden ser visibles del todo cuando su padre está en la cima de su respectiva pila y se halla mapeado en el display.
5. El servidor X en sí carece de funciones de gestión, sólo realiza el recorte de las ventanas sobre el puerto de visión según sus pilas (Ver **Figura 10** y **Figura 11**). Cada cliente es responsable de sus propias ventanas. Existe un administrador de ventanas que manipula las ventanas de la cima de todos los clientes. El administrador no forma parte del servidor X, sino que es en sí un cliente más. En

cuanto el administrador cambia algo en la pantalla (por ejemplo, redimensiona una ventana), hace que el servidor X envíe un evento a los demás clientes.

- Los clientes envían mensajes de solicitud al servidor, que contesta con mensajes de respuesta o mensajes de error. El Servidor X también puede enviar mensajes de eventos a las aplicaciones. Los mensajes de eventos indican cambios en las ventanas (y su visibilidad), y en la entrada de usuario (mouse y teclado).

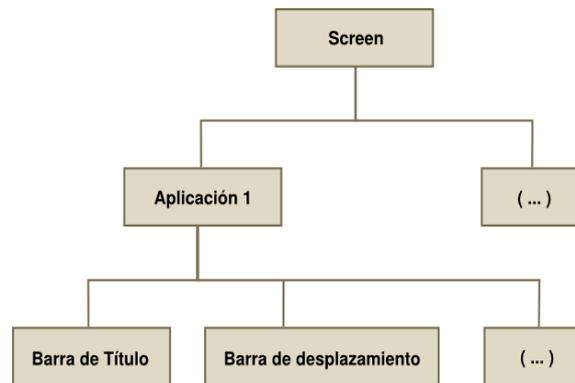


Figura 10. Estructura de ventanas del Sistema de Ventanas X: en la figura cada ventana es hija de otra ventana, siendo la raíz el display.

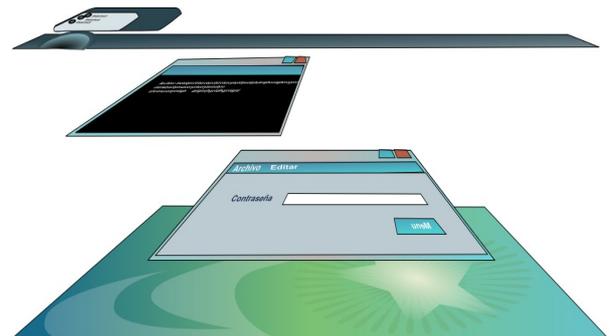
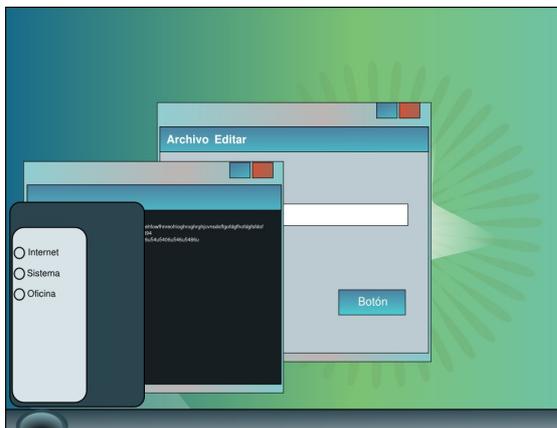


Figura 11. Estructura de ventanas del sistema de ventanas X: la figura muestra la manera en que el orden de apilamiento se refleja visualmente en el display mediante el recorte de las ventanas.

1.5.4 Presente y futuro del sistema de ventanas X

Es harto conocido que las grandes compañías y consorcios no solo de la informática, sino de cualquier rama, tienen por costumbre sentarse a esperar debajo del árbol que otros sembraron para recoger sus frutos; para nadie es un secreto la manera en la que Microsoft copia de sus competidores, o como Apple aprovecha una idea genial de terceros y la convierte en un producto sin precedentes.

En el caso del sistema de ventanas X ha ocurrido algo similar a lo largo de su historia, pues aún con las deficiencias del caso, X continúa siendo un software de calidad. Ya una vez la idea fue tomada impunemente y mejorada por Mac OSX con su tecnología Quartz, actualmente bajo patente, en tanto parece que Red Hat se enfila a realizar lo mismo con AIGLX y Wayland. Compiz, única tecnología que en el sistema está pensada para funcionar como un administrador de composición de ventanas, fue desarrollada por Novell y a los días de hoy continúa bajo su tutela. Pareciera que todas las grandes compañías se han puesto de acuerdo para tomar una tajada del gigante del Open Group, sin que puede determinarse cuánto durará su cuerpo de árbol, no obstante podría vaticinarse un futuro movido por las divisiones, pues aún cuando una absorción es posible, el poder de las comunidades que una vez sostuvieron el proyecto podría hacerse cargo otra vez de continuarlo para los que como nosotros el software libre es la única alternativa posible.

Estas ideas demuestran la importancia que tiene para Cuba el construir sus propias herramientas, sobre todo si se tiene en cuenta las restricciones del bloqueo, extensibles también a las TICs.

1.5.5 Más allá del sistema de ventanas X

Aún cuando el sistema de ventanas X significó un paso de avance notable para la época en que surge, lo cierto es que en la actualidad presenta una serie de inconvenientes que motivan a pensar en una renovación de su arquitectura con miras al futuro que se avecina.

La base del código que conforma el actual sistema X.Org Server es antiguo y en algunos lugares innecesario. El servidor de X.Org sigue evolucionando y ha recibido una serie de importantes adiciones en los últimos tiempos, que no obstante constituyen recubrimientos a sus evidentes deficiencias. Circunstancias como estas son las que han llevado al desarrollo, a lo largo de toda su vida, de alternativas como Xgl y más recientemente Wayland.

Uno de los argumentos que avalan la afirmación anterior es el hecho de que el uso de OpenGL en el sistema de ventanas X ocurre de manera indirecta por medio de GLX y no del protocolo X propiamente, lo cual evidencia que este no soporta OpenGL de manera nativa todavía, pues la base del protocolo X está determinada por sintaxis de PostScript, un lenguaje orientado a gráficos bidimensionales que ha llevado a la migración reciente de una parte importante del sistema hacia XML a fin de aligerar su estructura con el desarrollo de XCB como sustituto de Xlib.

El uso de estándares como XML para describir el formato del protocolo X demuestra la importancia de este lenguaje y los dialectos derivados de él como SVG y X3D para la transmisión de información gráfica por la red, especialmente para gráficos tridimensionales como los proporcionados por OpenGL.

Las ventajas de OpenGL en el desarrollo de un entorno gráfico enfocado a la multimedia, que además permita diferentes tipos de metáforas, son evidentes, pues esta API:

- Implementa la técnica de doble buffer, haciendo posible la composición.
- Posee una mayor cantidad de primitivas geométricas, sin que estas definan una metáfora visual en particular.
- Permite gráficos en tercera dimensión.
- Es soportada por el hardware que producen numerosas compañías como NVIDIA o ATI, y la tendencia es al incremento.

Todas estas ideas hacen pensar en la necesidad de un sistema un tanto diferente al

sistema de ventanas X actual, que integre las modificaciones arquitectónicas anteriormente relatadas, pero que sin embargo conserve las que a juicio de los autores son las principales ventajas de X Window System, dadas por el empleo del modelo cliente-servidor que implementa:

- Las aplicaciones no tienen que conocer las características hardware de la terminal.
- Las aplicaciones no tienen que estar en el mismo ordenador que la terminal.
- Los programas escritos en Xlib/XCB son portables.
- Los programadores no tienen que ocuparse de las comunicaciones, sólo han de escribir aplicaciones gráficas para Xlib/XCB, con independencia de si los usuarios son locales o remotos.

El entorno además debe estructurarse de tal forma que no defina por sí mismo desde el servidor la metáfora a emplear, a fin de que el cliente sea quien decida la metáfora que mostrará como interfaz gráfica al usuario.

CAPÍTULO II. ARQUITECTURA PARA ENTORNO GRÁFICO ENFOCADO A MULTIMEDIA

Hacer posible un entorno gráfico con características multimedia, que implemente diversas metáforas visuales requiere antes que nada diseñar su estructura y establecer sus principios de funcionamiento, es decir su arquitectura. La arquitectura es la pieza angular de cualquier sistema informático por cuanto constituye una guía para la implementación del software en términos de lo que ha de hacer, y cómo, explicitando las interacciones entre sus partes.

Es objetivo de este capítulo diseñar a grandes rasgos, por cuanto son varias las piezas de software que lo componen, la arquitectura de un futuro entorno gráfico con las características anteriormente mencionadas, exponiendo además los requerimientos y herramientas para su desarrollo. La arquitectura serviría para definir un sistema mayor orientado completamente a la multimedia que permita integrar información sonora, así como el trabajo con el sistema operativo subyacente, esto es el manejo del sistema de ficheros y algunas funcionalidades, que necesitan ser tratadas para su integración con el gráfico.

2.1 Principios generales de la arquitectura

Optimizar el código

Es un criterio generalizado por estos días el adquirir tecnologías “fascinantes” que en la práctica no son funcionales, sencillamente porque están hechas para explotar capacidades con las cuales no se cuenta. La mayor parte de estas tecnologías son el resultado de la reutilización de código, las más de las veces redundante, que nunca se revisa ni se optimiza, pues van dirigidas a usuarios de países económicamente desarrollados con un poder adquisitivo que dista mucho del Tercer Mundo; otras son recubrimientos de tecnologías subyacentes ya obsoletas o en decadencia, que meritan ser cambiadas.

En el mundo de los gráficos por ordenador, debido a los complejos cálculos asociados al procesamiento de la información de los píxeles, la eficiencia del código resulta de vital importancia, de manera que todo sistema gráfico que se implemente debe empeñarse en hacer un uso eficiente de las potencialidades del hardware, sin afectar en lo posible sus prestaciones, máxime para un país como Cuba cuyos recursos son limitados y las necesidades asociadas a los numerosos programas de la Revolución enormes.

No imponer una metáfora visual

Todas las personas ven el mundo de manera diferente, aún cuando pueden distinguirse ciertos patrones repetitivos en su mapa conceptual, resultado de la influencia del grupo o grupos sociales a los que pertenecen, sus costumbres, credo e idiosincrasia. Imponer por tanto una metáfora visual para todos los usuarios, solo proporciona problemas de comprensión que laceran la usabilidad del software. Un sistema gráfico debe constituirse sobre la base de estos principios, proporcionando una plataforma que permita la implementación de cualquier tipo de metáfora visual con un gasto mínimo de esfuerzo. En ese sentido es importante que el sistema, funcione en el nivel más bajo de manera abstracta sobre el concepto de un sistema de objetos, como los presentados por OpenGL, y no un sistema de ventanas como es lo habitual.

Utilizar el modelo de los sistemas distribuidos basados en la arquitectura cliente-servidor

Uno de los principales avances de estos tiempos en materia informática lo son los sistemas distribuidos, de lo cuales el sistema de ventanas X, aún con las deficiencias del caso, es un ejemplo, que se ha ganado el reconocimiento y respeto de los usuarios dentro y fuera del ámbito del software libre.

El modelo implementa una arquitectura cliente-servidor, cuyos beneficios en la Era de Internet son apreciables, pues permiten una mayor eficiencia y control de los recursos, al descentralizar el procesamiento de estos, particularmente de los servicios y la visualización de la interfaz gráfica de usuario.

Encargarle al núcleo del sistema operativo el control del hardware

El sistema de ventanas X, utilizado como referencia para este trabajo, opera como una máquina virtual de procesos que maneja el hardware, principalmente el asociado a los periféricos como el mouse, el teclado y el monitor, así como la tarjeta gráfica. Esta característica es la que hace posible que X funcione en todos los sistemas operativos, es decir que sea multiplataforma, pero al mismo tiempo provoca que el procesamiento no sea el más eficiente, como resultado de realizar actividades que solo le competen al núcleo.

El desarrollo de un entorno gráfico, que aproveche las ventajas de procesamiento del núcleo permite, que la carga de trabajo sobre el sistema sea menor como resultado de la disminución de la complejidad de este, lo cual hace patente una vez más la frase romana: "Al César lo que es del César", que llevada a un lenguaje informático, podría interpretarse como: "Al kernel lo que es del kernel".

Posibilitar la composición

Hacer que un entorno gráfico sea verdaderamente interactivo, incluye la posibilidad de efectos que mantengan al usuario en constante atención. La composición es una técnica que permite realizar estos efectos, los cuales pueden ser aprovechados en ambientes multimedia.

2.2 Herramientas, tecnologías y metodologías para el desarrollo

2.2.1 Lenguajes de programación

Lenguaje de programación C

C es un lenguaje de programación creado en 1969 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell. Es un lenguaje orientado a la implementación de sistemas operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce, especialmente para el procesamiento de algoritmos complejos, como los relacionados con los gráficos por computador.

Entre las características más importantes de C **[25]** destacan las siguientes:

- Posee un núcleo simple, con una amplia gama de funcionalidades proporcionadas por bibliotecas.
- Es un lenguaje muy flexible que permite programar con múltiples estilos.
- Un sistema de tipos que impide operaciones sin sentido.
- Usa un lenguaje de preprocesado para tareas como definir macros e incluir múltiples ficheros de código fuente.
- Acceso a memoria de bajo nivel mediante el uso de punteros.
- Un conjunto reducido de palabras clave.
- 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.
- Tipos de datos agregados que permiten que datos relacionados se combinen y se manipulen como un todo.

XML

XML(Extensible Markup Language) es un lenguaje para estructurar cualquier tipo de datos, haciendo más fácil para el ordenador el proceso de generar y controlar estos, lo cual lo ha convertido en un estándar para la comunicación entre procesos y plataformas, y especialmente para el desarrollo de frameworks o APIs como Gtk, Qt o Motif, que tienen como objetivo definir las interfaces gráficas de usuario.

Contiene características que aseguran su validez durante mucho tiempo, es ampliable, independiente de plataforma, no pertenece a ninguna firma concreta de software, soporta internacionalización y es un estándar de la W3C.

SVG

SVG(**S**calable **V**ector **G**raphics) es un lenguaje para describir gráficos vectoriales bidimensionales, tanto estáticos como animados, que se convirtió en una recomendación

del W3C en septiembre de 2001, fecha desde la cual rivaliza con Macromedia Flash en términos de potencia y poder, con la diferencia de que es un estándar abierto.

El dibujado en los SVG puede ser dinámico e interactivo. Un juego amplio de manejadores de eventos pueden ser asignados a cualquier objeto SVG.



Figura 12. Diferencias de calidad entre el dibujado vectorial y el de mapa de bit

Debido a que SVG es una especificación de XML para gráficos bidimensionales, puede emplearse como un estándar para el proceso de render de los componentes visuales de un entorno gráfico con estas características.

X3D

X3D(Extensible 3D), estándar ISO para gráficos vectoriales en tercera dimensión, que puede emplear tanto una sintaxis similar a la de XML como una de tipo VRML(Virtual Reality Modelling Language), amplía VRML con extensiones de diseño y brinda la posibilidad de emplear XML para modelar escenas completas en tiempo real. X3D es ideal para el renderizado de componentes visuales de entornos gráficos tridimensionales.



Figura 13. Ejemplo de dibujo vectorial tridimensional con X3D

2.2.2 Tecnologías

OpenGL

OpenGL es una especificación de un API multilenguaje y multiplataforma usado por la mayoría de los entornos gráficos de hoy para añadirle características 3D a la metáfora que estos implementan, aunque encuentra su uso principalmente en el desarrollo de videojuegos, realidad virtual, representación científica, simulación de vuelo y en programas de infografía debido entre otras cosas a que es un estándar.

El funcionamiento básico de OpenGL consiste en aceptar primitivas tales como puntos, líneas y polígonos, y convertirlas en píxeles, algo muy similar a lo que realiza X Window System, pero con la diferencia de que soporta un mayor número de primitivas y funciones de transformación, iluminación, y la posibilidad de programar algoritmos complejos en un espacio que este no domina. Este proceso es realizado por una pipeline gráfica conocida como la Máquina de Estados de OpenGL.

OpenGL ha influido notablemente en el desarrollo de las tarjetas gráficas, promocionando un nivel básico de funcionalidad que actualmente es común en el hardware comercial.

Otra API similar a OpenGL, pero con menos prestaciones solo existe en software no libre y es propiedad de Microsoft.

Bash

Bash (GNU Project) es un intérprete de comandos de tipo Unix (shell) escrito para el proyecto GNU. Su nombre es un acrónimo de Bourne-Again Shell. Es el shell por defecto en la mayoría de las distribuciones de GNU/Linux de hoy en día. Se encarga de interpretar las órdenes para su proceso por el kernel. Al escribir un comando, es Bash el responsable de interpretarlo y ordenar que se ejecute.

Se propone esta herramienta para la interacción de bajo nivel con el ordenador, debido a que el desarrollo del entorno no requiere de ninguna interfaz gráfica de por medio, al menos en sus inicios.

GCC

GNU Compiler Collection -Colección de Compiladores GNU-, es un conjunto de compiladores creados por el proyecto GNU. GCC es software libre y lo distribuye la Free Software Foundation bajo la licencia GPL. Estos compiladores se consideran estándar para los sistemas operativos derivados de UNIX, de código abierto o también propietarios, como Mac OS X.

El desarrollo de GCC usa un entorno de desarrollo abierto y soporta muchas otras plataformas con el fin de fomentar el uso de un compilador-optimizador de clase global.

Vim

Vim (del inglés Vi IMproved) es una versión mejorada del editor de texto vi, presente en todos los sistemas UNIX. La principal característica de Vim consiste en que dispone de diferentes modos entre los que se alterna para realizar ciertas operaciones, lo que le diferencia de la mayoría de editores comunes.

Sus principales funcionalidades son:

- Corrector ortográfico integrado
- Autocompletado de texto
- Navegación por pestañas

- Ventanas múltiples, que dividen el área de edición horizontal o verticalmente.
- Resaltado de sintaxis dependiente del lenguaje de programación o de etiquetas utilizado
- Comandos deshacer y rehacer
- Comprensión de más de 200 sintaxis diferentes
- Lenguaje de scripting para programar extensiones
- Completado de comandos, palabras y nombres de ficheros
- Compresión y descompresión de ficheros, que posibilita editar ficheros comprimidos
- Reconocimiento de formatos de fichero y conversión entre los mismos.
- Historial de comandos ejecutados
- Grabación y reproducción de macros
- Guardado de la configuración entre sesiones
- Plegado automático y manual de código
- Interfaz gráfico opcional
- Altamente configurable y personalizable

GForge

GForge es un software que ofrece alojamiento de proyectos, control de versiones (CVS y Subversion), seguimiento de fallos, y mensajería. Es software libre licenciado bajo la GPL (GNU General Public License), aunque no su última versión GForge Advance Server. En la actualidad GForge es la plataforma implementada para los proyectos de la facultad 10 de la Universidad de las Ciencias Informáticas.

2.2.3 Metodología de desarrollo

Por la envergadura del proyecto a realizar se requiere de una metodología de desarrollo rápida y flexible, que describa un proceso sencillo y bien documentado de los elementos imprescindibles tanto para los clientes como para los desarrolladores, proponiéndose SXP, la cual reúne muchas de estas características, y fue desarrollada en la Universidad de las Ciencias Informáticas para proyectos de software libre.

SXP

SXP es una metodología compuesta por las metodologías SCRUM y XP, que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles, que permite actualizar los procesos de software para el mejoramiento de la actividad productiva, propicia el desarrollo de la creatividad, eleva el nivel de preocupación y responsabilidad de los miembros del equipo y ayuda al líder del proyecto a tener un mejor control de este.

SCRUM es una metodología para gestionar un equipo de manera que trabaje eficientemente, proporcionando información sobre los progresos alcanzados.

XP es una metodología encaminada al proceso de desarrollo, que se centra en el principio de una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, lo cual es requisito indispensable para el éxito del proyecto.

Consta de 4 fases principales:

- **Planificación - Definición:** donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- **Desarrollo:** es donde se realiza la implementación del sistema hasta que este listo para ser entregado.
- **Entrega:** es la puesta en marcha.
- **Mantenimiento:** es la fase donde se realiza el soporte para el cliente.

Por 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 para documentar todo el proceso.

Las entregas son frecuentes, y existe una refactorización continua, lo que nos permite mejorar el diseño cada vez que se le añade una nueva funcionalidad. SXP está especialmente orientada a 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.

Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo bien delimitado, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes puedan ver día a día cómo progresa el trabajo. **[26]**

2.3 Descripción

Siguiendo la lógica de los sistemas distribuidos, la arquitectura que se propone se fundamenta, como se ha mencionado antes, en el esquema cliente-servidor, y en la especialización de tres capas o niveles de este que implementa el sistema de ventanas X , con varios cambios estructurales que combinados definen un nuevo paradigma.

2.3.1 Arquitectura cliente-servidor

La arquitectura cliente-servidor consiste en un programa cliente y un programa servidor que se comunican mediante mensajes a través de un protocolo. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta obligatoriamente sobre una sola máquina ni es necesariamente un sólo programa.

Estas y otras características permiten:

- Descentralizar el procesamiento y recursos, sobre todo, de cada uno de los servicios y de la visualización de la Interfaz Gráfica de Usuario.
- Que los servidores estén dedicados solo a una aplicación determinada, y por tanto puedan ejecutarla de manera eficiente.
- Que una determinada aplicación pueda ejecutarse sobre PCs distintas.

El servidor

En el caso de la arquitectura que se propone el servidor no maneja directamente el hardware como ocurre en el sistema de ventanas X, sino que se sirve del sistema operativo subyacente para hacerlo por medio de una Capa de Abstracción denominada CASO (Capa de Abstracción de Sistema Operativo). Actividades como almacenar recursos gráficos, permitir el renderizado de las aplicaciones, gestionar con los clientes y el sistema operativo los eventos provocados por el mouse y el teclado y comunicarse con otros servidores son responsabilidad también del servidor. La tecnología empleada en el servidor es OpenGL cuyos beneficios fueron relatados en una sección anterior.

Entre las **ventajas** que este modelo proporciona están:

- Permite gestionar el proceso de renderizado en varias PCs mediante la comunicación con otro servidor, lo cual resulta muy útil para aplicaciones gráficas de diseño 3D.
- Permite crear una capa de abstracción del sistema operativo para las aplicaciones, posibilitando su funcionamiento sobre varias plataformas.

Desventajas

- La gestión básica del hardware la realiza el sistema operativo, dependiendo de este la calidad y eficiencia de su funcionamiento.

El cliente

Conceptualmente para la arquitectura los clientes son las aplicaciones gráficas, el framework empleado para su implementación y el API básico sobre el cual se sustenta, encargado este último de hacer las peticiones de recursos al servidor y gestionar directamente los eventos que envía. El cliente se encarga además de conformar la interfaz gráfica de usuario y enviar la información de esta al servidor para su renderizado.

El protocolo

El protocolo de red es quien permite la comunicación entre el servidor y el cliente, o entre servidores para la ejecución remota de una o varias aplicaciones: un servidor se comunica con otro servidor para transmitirle la información de las aplicaciones que gestiona. El formato que se propone para los mensajes se basa en el empleo de alguna especificación XML, por su flexibilidad y sencillez de uso.

2.3.2 Arquitectura en tres capas

La arquitectura en tres capas tiene como objetivo fundamental el separar la lógica de negocios de la lógica de diseño, confiando una función simple a cada nivel, lo cual permite:

- Distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.
- Permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).

De forma general las capas que define el patrón son: capa de presentación, capa lógica y la capa de acceso a datos. Para la arquitectura del entorno que se propone la definición del funcionamiento de estas capas sería:

Capa de presentación

La capa de presentación es la encargada de definir la interfaz gráfica de usuario por medio de herramientas que implementan alguna o varias metáforas visuales sobre la base de alguna especificación XML, de manera que es en esta capa donde se conceptualiza el entorno gráfico para el usuario.

Capa de aplicación

La capa de aplicación es una subcapa dentro de la capa de presentación, en la cual se localizan las aplicaciones gráficas, con las que interactúa el usuario. De manera general estas pueden clasificarse, de acuerdo a la función que realizan en:

1. Aplicaciones de entorno gráfico: son aquellas aplicaciones enfocadas a la interacción del usuario con el entorno gráfico o que permiten su personalización.
2. Aplicaciones de usuario: son aplicaciones dedicadas a realizar tareas propias de los usuarios, las cuales no involucran al sistema, ni modifican su estado, tales como editar textos, realizar gráficos, programar, jugar o visualizar una película o vídeo.

Capa de herramientas GUI o Framework GUI

En esta subcapa se implementan los componentes de las metáforas tales como botones, menús, barras de desplazamiento, entre otros objetos conocidos comúnmente como gadgets, los cuales son utilizados por los programadores de aplicaciones gráficas en sus desarrollos.

Capa lógica

Esta capa se enfoca en los procesos de intercambio entre el servidor y la librería básica del cliente, explicados con anterioridad.

Capa de acceso a datos

La capa de acceso a datos tiene como objetivo permitir la manipulación de los datos que gestiona el servidor, los cuales se localizan en una Base de Datos de Recursos Gráficos

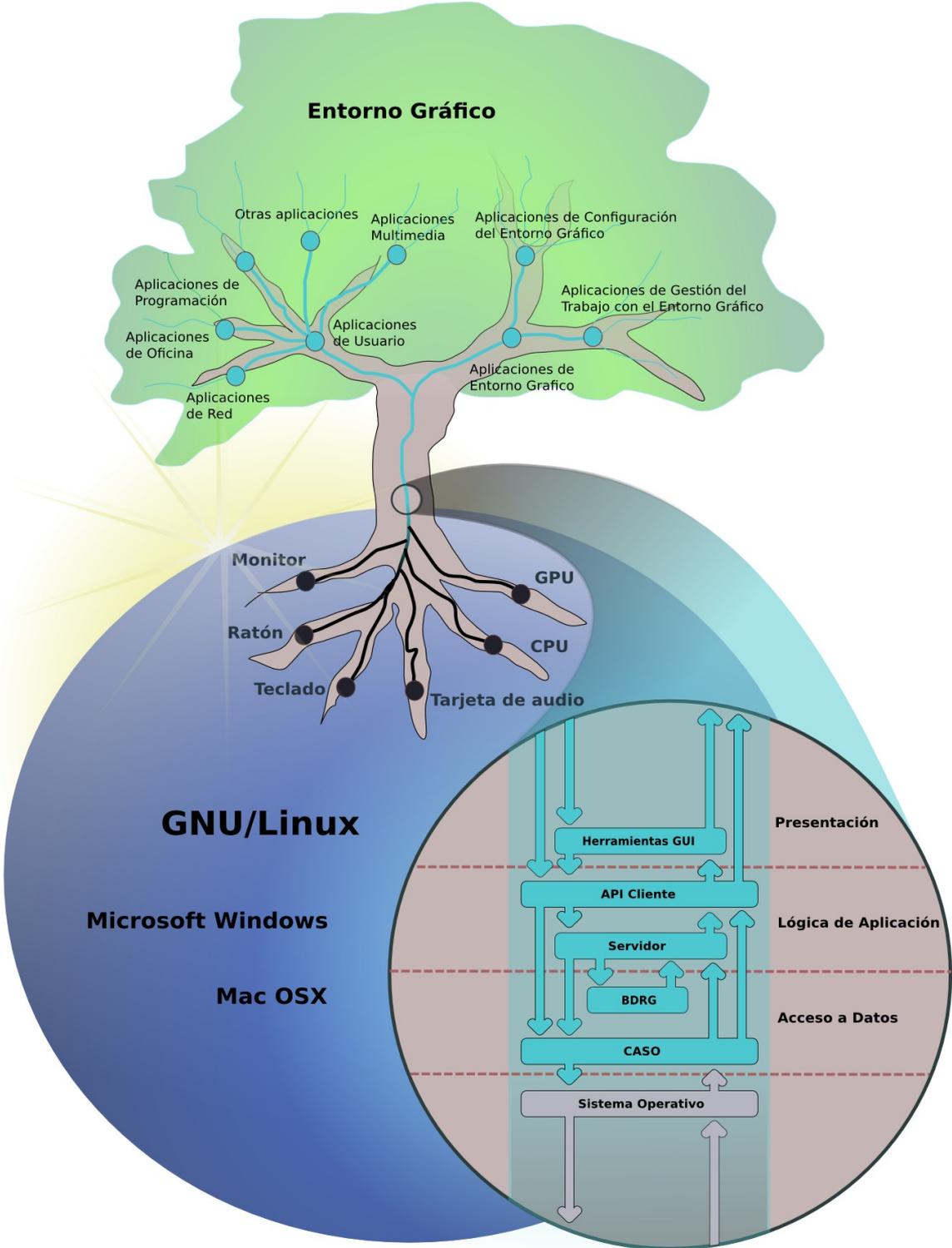
(BDRG). Posibilita además la comunicación con el sistema operativo subyacente por medio de la Capa de Abstracción de Sistema Operativo (CASO).

2.4 Modelos arquitectónicos

2.4.1 Modelo físico



2.4.2 Modelo lógico



2.5 Sobre la arquitectura

Debido a que la arquitectura propone un nuevo paradigma sin precedentes para los entornos gráficos, no es compatible con las interfaces gráficas de usuario existentes, como tampoco las actuales son compatibles entre ellas. No obstante esta realidad, los programas de otras plataformas pueden correr sin problemas sobre el sistema propuesto, solo que necesitarían de una implementación de su API para este, que traduzca el API nativo en términos de XML, lo cual de hacerse, sería en la capa de presentación, manteniéndose el resto de las capas inalterables.

La arquitectura en sí misma, permite que el entorno sea multiplataforma, es decir que pueda correr sobre sistemas operativos como Windows o Mac OSX, por medio de la CASO (Capa de Abstracción de Sistema Operativo), sin que haya que reimplementar otras capas.

El uso de estándares como SVG y X3D permiten traducir en términos de un mismo lenguaje los diferentes dialectos XML que implementarían los frameworks o APIs para GUIs, facilitando el proceso de render de estos por medio de una estructura común que permanecería inalterable.

La implementación de una metáfora concreta ocurriría a nivel de la capa de presentación y no de la lógica del negocio, por lo que su programación no afectaría la base del sistema.

Una de las desventajas de esta arquitectura es su dependencia del sistema operativo en cuanto al control de hardware, lo que hace difícil el control de los parámetros de eficiencia funcional de esta.

CAPÍTULO III. IMPLEMENTACIÓN DEL GESTOR DE CONFIGURACIÓN DE INTERFACES TOCORORO

En el presente capítulo se explica el proceso de implementación del Gestor de Configuración de Interfaces Tocatoro, en lo adelante GCIT, en forma de historias de usuarios, prototipos de interfaz de usuario y otros modelos auxiliares.

Se realiza además una valoración crítica, partiendo de las dificultades técnicas encontradas durante el proceso de implementación, del diseño propuesto por el analista a fin de corroborar de manera indirecta la validez de la arquitectura que se propone en este trabajo.

3.1 Valoración crítica del diseño propuesto por el analista

Haciendo un análisis del diseño propuesto por el analista para el GCIT (Gestor de Configuración de Interfaces Tocatoro), puede decirse que conceptualmente es bueno, debido a su arquitectura basada en módulos independientes, tal cual ocurre con las interfaces que dichos módulos configuran. Por otro lado la interfaz propuesta para la aplicación es fácil de comprender, simple y aceptable para los estándares vigentes, si bien no es menos cierto que presenta dificultades en su implementación, como el hecho de que no sea posible sino es mediante un administrador de composición, propiciar la opacidad en la barra de título y los bordes de las ventanas, su decoración con gráficos no homogéneos o su forma no rectangular con antialiasing.

A lo anteriormente planteado, y después de una revisión exhaustiva de los elementos expuestos, de la visión inicial del proyecto, su concepción y diseño, unido a los resultados de las investigaciones plasmadas en este trabajo, se hace necesario añadir la necesidad de corregir, a juicio de los autores algunos puntos importantes.

Sobre el concepto

El GCIT no es un gestor de paquetes, de manera que no tiene como función la de instalar ningún tipo de software, pues existen aplicaciones como Synaptic o Adept que ya tienen esta responsabilidad.

Sobre la misión

La tarea del GCIT es gestionar la configuración de las interfaces gráficas de los sistemas GNU/Linux solamente , lo cual está determinado por las siguientes operaciones:

- Agregar a la lista de temas un nuevo tema de interfaz.
- Activar un tema de interfaz seleccionado.
- Mostrar datos del tema de interfaz.
- Diseñar un tema de interfaz.

Nota: Los temas de interfaz disponibles, o al menos los que gestiona la aplicación son los que aparecen en la lista de temas de cada uno de los módulos.

Sobre el alcance

El alcance del GCIT se extiende solo a GNOME, de manera que ideas como TocoPan o GnuS formarían parte de un nuevo entorno gráfico.

Sobre los aportes tecnológicos

Los aportes tecnológicos del GCIT estarían enmarcados en la integración que conceptualmente hace de los procesos de configuración de las interfaces gráficas del sistema, y la importancia para los usuarios de configurar mediante solo algunos clic, las interfaces de su entorno GNOME.

Sobre la visión

El desarrollo del GCIT, ha servido para constatar la necesidad de un entorno gráfico diferente a los actuales, donde la idea de un gestor de configuración de interfaces con iguales o mejores prestaciones cobra igual sentido e importancia.

Sobre las historias de usuario

Las funciones descritas en las historias de usuario no se corresponden en su mayoría con la lógica actual del proceso, en tanto que otras obvian funcionalidades implícitas que se presentan sin embargo en esta versión del producto. Debido a los anteriores problemas se ha tenido que modificar las historias de usuario originales a fin de ajustarlas a las nuevas condiciones, incluyéndose además nuevas tareas de ingeniería para cada una de ellas, las cuales se presentan en el epígrafe correspondiente.

Sobre los lenguajes, herramientas y tecnologías empleadas

Las tecnologías, lenguajes y herramientas utilizadas en el desarrollo del GCIT son las propuestas por el analista, salvo que no se hará uso de los lenguajes de programación XML y C/C++ , y el IDE Anjuta se ha sustituido por el editor de textos SciTE. Python con Gtk se mantienen como lenguaje y API respectivamente para la implementación tanto de la interfaz como de las funcionalidades que esta recubre.

SciTE

SCIntilla based Text Editor o SciTE es un editor de textos multiplataforma escrito por Neil Hodgson usando el componente de edición Scintilla.

Ligero y hecho para ser veloz, está diseñado principalmente para edición de código fuente, y resaltado de sintaxis, y tiene referencia de función en línea para muchos lenguajes de computación. SciTE comparte algunas características con Notepad++ que también está basado en el componente de edición Scintilla.

3.2 Planificación del proyecto por roles

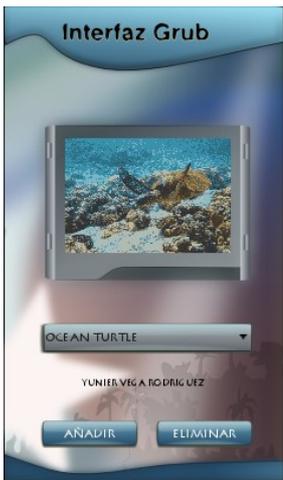
Tabla V. Roles de los Miembros del Proyecto.

Rol	Miembro
Gerente/Management	Eiger Mora Moredo
Cliente/Customer	Nova GNU/Linux
Desarrollador/Developer	Javier Menéndez Rizo Abel Ernesto Cuadrado Sospedra Eiger Mora Moredo
Analista/Analyst	Yunier Vega Rodríguez Eiger Mora Moredo
Diseñador/Designer	Yunier Vega Rodríguez
Encargado de Prueba/Tester	Eiger Mora Moredo
Arquitecto/Architect	Yunier Vega Rodríguez

3.3 Historias de usuario y prototipos de interfaz de usuario

A continuación se relacionan las historias de usuario y los prototipos de interfaz de usuarios. Estas definen los requisitos funcionales que deben ser implementados en la etapa de elaboración, sin embargo, constituyen una planificación inicial del proceso, que en su evolución debe ser evaluado y adecuado a las nuevas necesidades y propuestas.

Historia de Usuario	
Número: U-TD-01	Nombre Historia de Usuario: Administrar módulos de configuración de las interfaces gráficas de usuario
Modificación de la Historia de Usuario:	
Usuario: Abel Ernesto Cuadrado Sospedra	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Asignados: 3
Riesgo en Desarrollo: Alto	Puntos Reales: 4
Descripción: Comienza cuando el usuario ejecuta la aplicación y se muestra su interfaz principal, con los módulos de configuración existentes.	
<p>Acceder al módulo de configuración de una interfaz</p> <ul style="list-style-type: none"> El usuario selecciona mediante clic la opción correspondiente a la interfaz, el sistema le muestra información visual y escrita de la misma, así como activa el módulo para su configuración. <p>Mostrar información de una interfaz</p> <ul style="list-style-type: none"> El usuario posiciona el mouse sobre la opción correspondiente a la interfaz y el sistema le muestra información visual y escrita de la misma. 	
Observaciones:	
Prototipo de interfaz:	
	

Historia de Usuario	
Número: U-TD-02	Nombre Historia de Usuario: Gestionar la configuración del Grub splash
Modificación de la Historia de Usuario:	
Usuario: Eiger Mora Moredo	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Asignados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 2
<p>Descripción: Comienza una vez activado el módulo de configuración del Grub splash desde la interfaz principal.</p> <p>Seleccionar y Activar un tema de Interfaz</p> <ul style="list-style-type: none"> El usuario selecciona primero un tema de la lista de temas existentes y entonces el sistema aplica la configuración elegida por el usuario al Grub splash. <p>Añadir un nuevo tema de interfaz</p> <ul style="list-style-type: none"> El usuario selecciona la opción Añadir y el sistema le muestra una ventana de búsqueda. El usuario busca el tema en el sistema de ficheros, que desea añadir y acepta, luego el sistema actualiza la lista de temas existentes. <p>Eliminar un nuevo tema de interfaz</p> <ul style="list-style-type: none"> El usuario selecciona la opción Eliminar y la aplicación procede a eliminar el tema activo. 	
Observaciones:	
<p>Prototipo de interfaz:</p> 	

Historia de Usuario	
Número: U-TD-03	Nombre Historia de Usuario: Gestionar la configuración del Bootsplash
Modificación de la Historia de Usuario:	
Usuario: Eiger Mora Moredo	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Asignados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 2
<p>Descripción: Comienza una vez activado el módulo de configuración del Bootsplash desde la interfaz principal.</p> <p>Seleccionar y Activar un tema de Interfaz</p> <ul style="list-style-type: none"> El usuario selecciona primero un tema de la lista de temas existentes y entonces el sistema aplica la configuración elegida por el usuario al Bootsplash. <p>Añadir un nuevo tema de interfaz</p> <ul style="list-style-type: none"> El usuario selecciona la opción Añadir y el sistema le muestra una ventana de búsqueda. El usuario busca el tema en el sistema de ficheros, que desea añadir y acepta, luego el sistema actualiza la lista de temas existentes. <p>Eliminar un nuevo tema de interfaz</p> <ul style="list-style-type: none"> El usuario selecciona la opción Eliminar y la aplicación procede a eliminar el tema activo. 	
Observaciones:	
<p>Prototipo de interfaz:</p> 	

Historia de Usuario	
Número: U-TD-04	Nombre Historia de Usuario: Gestionar la configuración del administrador de sesión del entorno de escritorio Gnome(GDM)
Modificación de la Historia de Usuario:	
Usuario: Eiger Mora Moredo	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Asignados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 2
<p>Descripción: Comienza una vez activado el módulo de configuración del GDM desde la interfaz principal.</p> <p>Seleccionar y Activar un tema de Interfaz</p> <ul style="list-style-type: none"> El usuario selecciona primero un tema de la lista de temas existentes y entonces el sistema aplica la configuración elegida por el usuario al GDM. <p>Añadir un nuevo tema de interfaz</p> <ul style="list-style-type: none"> El usuario selecciona la opción Añadir y el sistema le muestra una ventana de búsqueda. El usuario busca el tema en el sistema de ficheros, que desea añadir y acepta, luego el sistema actualiza la lista de temas existentes. <p>Eliminar un nuevo tema de interfaz</p> <ul style="list-style-type: none"> El usuario selecciona la opción Eliminar y la aplicación procede a eliminar el tema activo. 	
Observaciones:	
<p>Prototipo de interfaz:</p> 	

3.4 Tareas de Ingeniería

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

Tarea de Ingeniería	
Número de la tarea: 1	Número Historia de Usuario: U-TD-01
Nombre de la Tarea: Diseñar prototipo de la interfaz principal	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha de Inicio: 03/11/2008	Fecha de Fin: 16/11/2008
Desarrollador Responsable: Yunier Vega Rodríguez	
Descripción: La interfaz principal debe permitirle al usuario seleccionar un módulo de configuración determinado, así como mostrar información sobre la misma.	

Tarea de Ingeniería	
Número de la tarea: 2	Número Historia de Usuario: U-TD-01
Nombre de la Tarea: Mostrar información de una interfaz	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha de Inicio: 17/11/2008	Fecha de Fin: 23/11/2008
Desarrollador Responsable: Javier Menéndez Rizo	
Descripción: Al hacer prelight sobre el botón de la interfaz correspondiente se cambia la imagen mapeada sobre este por la relativa al evento realizado si el botón no se encuentra activo.	

Tarea de Ingeniería	
Número de la tarea: 3	Número Historia de Usuario: U-TD-01
Nombre de la Tarea: Acceder al módulo de configuración de una interfaz	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha de Inicio: 24/11/2008	Fecha de Fin: 30/11/2008
Desarrollador Responsable: Abel Ernesto Cuadrado Sospedra	
Descripción: Al hacer clic sobre el botón de la interfaz correspondiente se cambia la imagen mapeada sobre este por la relativa al evento realizado, además se muestra información sobre la interfaz tanto visual como escrita. Seguidamente se carga su interfaz. Cuando se hace clic sobre un botón de algún módulo de configuración se requiere desactivar los demás botones activos, cambiando la imagen mapeada sobre estos por la relativa al estado desactivado.	

Tarea de Ingeniería	
Número de la tarea: 4	Número Historia de Usuario: U-TD-02
Nombre de la Tarea: Diseñar prototipo de interfaz del módulo de configuración del Grubsplash	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de Inicio: 01/12/2008	Fecha de Fin: 03/12/2008
Desarrollador Responsable: Yunier Vega Rodríguez	
Descripción: La interfaz deberá proveer una lista con los temas disponibles, así como mostrar una vista previa del tema activo y las opciones de añadir y eliminar un tema de interfaz.	

Tarea de Ingeniería	
Número de la tarea: 5	Número Historia de Usuario: U-TD-02
Nombre de la Tarea: Seleccionar y Activar un tema de interfaz Grubsplash	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de Inicio: 04/12/2008	Fecha de Fin: 07/12/2008
Desarrollador Responsable: Eiger Mora Moredo	
Descripción: Al seleccionar la opción correspondiente se activa el tema de interfaz y se muestra el progreso del procedimiento.	

Tarea de Ingeniería	
Número de la tarea: 6	Número Historia de Usuario: U-TD-02
Nombre de la Tarea: Añadir un tema de interfaz Grubsplash	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de Inicio: 08/12/2008	Fecha de Fin: 11/12/2008
Desarrollador Responsable: Javier Menéndez Rizo	
Descripción: Una vez seleccionada la opción Añadir, se muestra una ventana de búsqueda donde el usuario referencia el lugar donde se encuentra el tema que desea añadir, el usuario acepta y se carga el tema buscado en la carpeta de temas de la interfaz, actualizándose además la lista de temas disponibles.	

Tarea de Ingeniería	
Número de la tarea: 7	Número Historia de Usuario: U-TD-02
Nombre de la Tarea: Eliminar un tema de interfaz Grub splash	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de Inicio: 12/12/2008	Fecha de Fin: 14/12/2008
Desarrollador Responsable: Eiger Mora Moredo	
Descripción: Una vez seleccionada la opción Eliminar, la aplicación procede a eliminar el tema activo.	

Tarea de Ingeniería	
Número de la tarea: 8	Número Historia de Usuario: U-TD-03
Nombre de la Tarea: Diseñar prototipo de interfaz del módulo de configuración del Boot splash	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de Inicio: 12/1/2009	Fecha de Fin: 14/1/2009
Desarrollador Responsable: Yunier Vega Rodríguez	
Descripción: La interfaz deberá proveer una lista con los temas disponibles, así como mostrar una vista previa del tema activo y las opciones de añadir y eliminar un tema de interfaz.	

Tarea de Ingeniería	
Número de la tarea: 9	Número Historia de Usuario: U-TD-03
Nombre de la Tarea: Seleccionar y Activar un tema de interfaz Boot splash	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de Inicio: 15/1/2009	Fecha de Fin: 18/1/2009
Desarrollador Responsable: Eiger Mora Moredo	
Descripción: Al seleccionar la opción correspondiente se activa el tema de interfaz y se muestra el progreso del procedimiento.	

Tarea de Ingeniería	
Número de la tarea: 10	Número Historia de Usuario: U-TD-03
Nombre de la Tarea: Añadir un tema de interfaz Bootsplash	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de Inicio: 19/1/2009	Fecha de Fin: 22/1/2009
Desarrollador Responsable: Abel Ernesto Cuadrado Sospedra	
Descripción: Una vez seleccionada la opción Añadir, se muestra una ventana de búsqueda donde el usuario referencia el lugar donde se encuentra el tema que desea añadir, el usuario acepta y se carga el tema buscado en la carpeta de temas de la interfaz, actualizándose además la lista de temas disponibles.	

Tarea de Ingeniería	
Número de la tarea: 11	Número Historia de Usuario: U-TD-03
Nombre de la Tarea: Eliminar un tema de interfaz Bootsplash	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de Inicio: 23/1/2009	Fecha de Fin: 25/1/2009
Desarrollador Responsable: Eiger Mora Moredo	
Descripción: Una vez seleccionada la opción Eliminar, la aplicación procede a eliminar el tema activo.	

Tarea de Ingeniería	
Número de la tarea: 12	Número Historia de Usuario: U-TD-04
Nombre de la Tarea: Diseñar prototipo de interfaz del módulo de configuración del GDM	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de Inicio: 26/1/2009	Fecha de Fin: 28/1/2009
Desarrollador Responsable: Yunier Vega Rodríguez	
Descripción: La interfaz deberá proveer una lista con los temas disponibles, así como mostrar una vista previa del tema activo y las opciones de añadir y eliminar un tema de interfaz.	

Tarea de Ingeniería	
Número de la tarea: 13	Número Historia de Usuario: U-TD-04
Nombre de la Tarea: Seleccionar y Activar un tema de interfaz GDM	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de Inicio: 29/1/2009	Fecha de Fin: 1/2/2009
Desarrollador Responsable: Eiger Mora Moredo	
Descripción: Al seleccionar la opción correspondiente se activa el tema de interfaz y se muestra el progreso del procedimiento.	

Tarea de Ingeniería	
Número de la tarea: 14	Número Historia de Usuario: U-TD-04
Nombre de la Tarea: Añadir un tema de interfaz GDM	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de Inicio: 2/2/2009	Fecha de Fin: 5/2/2009
Desarrollador Responsable: Javier Menéndez Rizo	
Descripción: Una vez seleccionada la opción Añadir, se muestra una ventana de búsqueda donde el usuario referencia el lugar donde se encuentra el tema que desea añadir, el usuario acepta y se carga el tema buscado en la carpeta de temas de la interfaz, actualizándose además la lista de temas disponibles.	

Tarea de Ingeniería	
Número de la tarea: 15	Número Historia de Usuario: U-TD-04
Nombre de la Tarea: Eliminar un tema de interfaz GDM	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha de Inicio: 6/2/2009	Fecha de Fin: 8/2/2009
Desarrollador Responsable: Eiger Mora Moredo	
Descripción: Una vez seleccionada la opción Eliminar, la aplicación procede a eliminar el tema activo.	

3.5 Plan de Versiones

En este epígrafe se define el plan de versiones para realizar las entregas intermedias y la entrega final. Tiene como entrada la relación de Historias de Usuario definidas previamente. Para colocar una historia en cada iteración se tiene en cuenta la prioridad que definió el cliente para dicha Historia.

Tabla VI. Historias de Usuario por Versión

Versión	Historias de Usuario	Tiempo estimado
1	1,2	6 semanas
2	3,4	4 semanas

3.6 Sobre el desarrollo del GCIT

El conjunto de pruebas fallidas realizadas a la interfaz con el propósito de lograr los efectos de opacidad, decoración no rectangular y textura no homogénea en los bordes de las ventanas, demuestra la necesidad e importancia de un administrador de composición dentro de un entorno gráfico orientado a la multimedia para el que las metáforas pueden ser objetos con estas características.

Con relación al diseño propuesto por el analista, el mismo tuvo que ser modificado para su implementación y orientado a las nuevas condiciones, que incluyen funcionalidades no concebidas en el plan inicial.

CONCLUSIONES GENERALES

La investigación realizada ha permitido llegar a la conclusión de que, desarrollar un entorno gráfico enfocado a multimedia, que permita implementar diferentes metáforas visuales es posible mediante la arquitectura propuesta, pues esta, a diferencia de sus predecesoras tiene las siguientes ventajas:

- Implementa un modelo basado en estándares como XML y OpenGL, tanto para la visualización como para la comunicación entre los clientes y el servidor, lo cual permite el diseño de interfaces en cualesquiera de las dimensiones conocidas, sentando las bases para la materialización de la idea de la diversidad de metáforas.
- No define una metáfora visual concreta en la capa base del entorno, sino en la capa de presentación, lo cual posibilita que la determinación de una metáfora en particular, sea posible sin necesidad de conocer detalles del funcionamiento interno de este.
- Simplifica la estructura de los entornos gráficos, al delegar en el núcleo del sistema operativo el manejo del hardware.

RECOMENDACIONES

Por la importancia del tema, se recomienda:

- Sistematizar en los procesos de renderizado OpenGL, a fin de elaborar un motor de render para el servidor.
- Sistematizar en las metáforas visuales, que permita proponer un modelo genérico para su descripción, el cual serviría de base para la elaboración de un protocolo OpenGL.
- Sistematizar en el funcionamiento de las redes de telecomunicaciones, específicamente en lo concerniente a los protocolos de comunicación.
- Implementar la arquitectura.
- Sobre la base del entorno implementado a partir de la arquitectura, desarrollar un sistema completamente multimedia, que integre información sonora.

REFERENCIAS BIBLIOGRÁFICAS

[1] Graciela Alonso, Silvana Asteggiante. *“Las metáforas en la interacción hombre-computador”*. 1999.

[Disponible en: <http://www.crnti.edu.uy/05trabajos/interface/hcisil.doc>]

[2] Wikipedia. *“Interacción Persona-ordenador”*. [Disponible en:

http://es.wikipedia.org/wiki/Interacci%C3%B3n_del_hombre_con_la_computadora]

[3] Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. & Carey, T., *“Human-Computer Interaction”*. 1994.

[4] Jesús Lorés, Joan Manuel Gimeno. Ferran Pérdríx. *“Metáforas, estilos y paradigmas”*.

[Disponible en: <http://griho.udl.es/ipo/ipo/doc/03Metafo.doc>]

[5], [6], [7], [8], [9], [10] Idem **[4]**

[11] Wikipedia. *“Interfaz Gráfica de Usuario”*.

[Disponible en: <http://es.wikipedia.org/wiki/GUI>]

[12] Graciela Alonso, Silvana Asteggiante. *“Las metáforas en la interacción hombre-computador”*. 1999.

[Disponible en: <http://www.crnti.edu.uy/05trabajos/interface/hcisil.doc>]

[13] Jesús Lorés, Joan Manuel Gimeno. Ferran Pérdríx. *“Metáforas, estilos y paradigmas”*.

[Disponible en: <http://griho.udl.es/ipo/ipo/doc/03Metafo.doc>]

[14] Idem **[13]**

[15] Graciela Alonso, Silvana Asteggiante. *“Las metáforas en la interacción hombre-computador”*. 1999.

[Disponible en: <http://www.crnti.edu.uy/05trabajos/interface/hcisil.doc>]

[16] Jesús Lorés, Joan Manuel Gimeno. Ferran Pérdríx. *“Metáforas, estilos y paradigmas”*.

[Disponible en: <http://griho.udl.es/ipo/ipo/doc/03Metafo.doc>]

[17], [18], [19], [20] Idem **[16]**

[21] Wikipedia. *“Multimedia”*.

[Disponible en: <http://es.wikipedia.org/wiki/Multimedia>]

[22], [23], [24] Idem **[21]**

[25] Wikipedia. *“Lenguaje de programación C”*.

[Disponible en: [http://es.wikipedia.org/wiki/C_\(lenguaje_de_programaci%C3%B3n\)](http://es.wikipedia.org/wiki/C_(lenguaje_de_programaci%C3%B3n))]

[26] Gladys Marsi, P. R., 2008, *“MA-GMPUR2 Metodología ágil para proyectos de software libre”*, Universidad de las Ciencias Informáticas, Facultad 10. [Disponible en: http://bibliodoc.uci.cu/TD/TD_0693_07.pdf]

BIBLIOGRAFÍA

Fuentes electrónicas

"El sistema X Window". [Disponible en:
<http://ditec.um.es/laso/docs/tut-tcpip/3376c49.html>]

James Gettys, Keith Packard. "*The(Re)Architecture of the X Window System*". 2004
[Disponible en: http://keithp.com/~keithp/talks/xarch_ols2004/]

Amador Durán Toro. "*Programación en X Window*". 1994
[Disponible en: <http://www.lsi.us.es/cursos/xlib.html>]

Fernando Pérez Costoya. "*Curso del sistema de ventanas X*". [Disponible en:
<http://laurel.datsi.fi.upm.es/~fperez/cursoX/indice.html>]

Daniel Marrique. "*X Window System Architecture Overview HOWTO*". 2001 [Disponible en:
<http://www.faqs.org/docs/Linux-HOWTO/XWindow-Overview-HOWTO.html>]

Matthias Hopf. "*Un escritorio OpenGL acelerado con Xgl y Compiz más allá de X11*".
[Disponible en: www.linux-magazine.es/issue/21/X11.pdf]

Wikipedia. "*Display PostScript*". [Disponible en:
http://es.wikipedia.org/wiki/Display_PostScript]

The X.Org Foundation. "*Documentation for the X Window System Version 11 Release 6.8.1 (X11R6.8.1)*". [Disponible en: <http://www.x.org/archive/X11R6.8.1/doc/>]

Freedesktop. "*Mesa 3D and Direct Rendering Infrastructure wiki*". [Disponible en:
<http://dri.freedesktop.org/wiki/>]

Wikipedia. “*Lenguaje de descripción de páginas*”. [Disponible en: http://es.wikipedia.org/wiki/Lenguaje_de_Descripci%C3%B3n_de_P%C3%A1gina]

Wikipedia. “*PostScript encapsulado*”. [Disponible en: http://es.wikipedia.org/wiki/PostScript_encapsulado]

Freedesktop. “*XCB*”. [Disponible en: <http://xcb.freedesktop.org/>]

Wikipedia. “*XCB*”. [Disponible en: <http://es.wikipedia.org/wiki/XCB>]

Wikipedia. “*X Window System*”. [Disponible en: http://es.wikipedia.org/wiki/X_Window_System]

Carlos Corrales Díaz. “*La Tecnologías Multimedia: Una nueva Tecnología de Comunicación e Información. Características, concepciones y aplicaciones*”.1994. [Disponible en: <http://iteso.mx/~carlosc/pagina/documentos/multidef.htm>]

Wikipedia. “*Reproductor Multimedia*”.
[Disponible en: http://es.wikipedia.org/wiki/Reproductor_multimedia]

Monografías. “*Multimedia*”. [Disponible en: <http://www.monografias.com/trabajos10/mmedia/mmedia.shtml>]

Wikipedia. “*Xnest*”. [Disponible en: <http://es.wikipedia.org/wiki/Xnest>]

Freedesktop. “*Wayland*”.
[Disponible en: <http://cgit.freedesktop.org/~krh/wayland/tree/NOTES>]

Phoronix. “*Wayland: A New X Server For Linux*”. [Disponible en: http://www.phoronix.com/scan.php?page=article&item=xorg_wayland&num=1]

Wikipedia. “*Xlib*”. [Disponible en: <http://es.wikipedia.org/wiki/Xlib>]

Wikipedia. “*Protocolo del sistema de ventanas X*”.

[Disponible en: http://es.wikipedia.org/wiki/Protocolo_del_sistema_de_ventanas_X]

David Santo Orcero. “*Tecnologías de Terminales*”. [Disponible en:

<http://www.revistaesalud.com/index.php/revistaesalud/article/viewArticle/237/553>]

Wikipedia. “*Comunicación entre procesos*”.

[Disponible en: http://es.wikipedia.org/wiki/Comunicaci%C3%B3n_entre_procesos]

Freedesktop. “*HAL-Hardware Abstraction Layer*”.

[Disponible en: <http://freedesktop.org/wiki/Software/hal>]

The X.Org Foundation. “*Documentation for the X Window System Version 11 Release 6.9 and 7.0 (X11R6.9/X11R7.0)*”.

[Disponible en: <http://www.x.org/archive/X11R7.0/doc/html/index.html>]

The X.Org Foundation. “*Releases/7.5*”.

[Disponible en: <http://www.x.org/wiki/Releases/7.5?>]

Wikipedia. “*PostScript*”. [Disponible en: <http://es.wikipedia.org/wiki/PostScript>]

Wikipedia. “*SMIL*”. [Disponible en: <http://es.wikipedia.org/wiki/SMIL>]

Wikipedia. “*XMMS*”. [Disponible en: <http://es.wikipedia.org/wiki/XMMS>]

Eric Anholt. “*High Performance X Servers in the Kdrive Architecture*”. [Disponible en:

http://www.usenix.org/events/usenix04/tech/freenix/full_papers/anholt/anholt_html/]

Wikipedia. “*XDMCP*”. [Disponible en: <http://es.wikipedia.org/wiki/XDMCP>]

“*Interfaces Gráficas de Usuario*”. [Disponible en: <http://sabia.tic.udc.es/gc/Contenidos20adicionales/trabajos/Interfaces/enlightment/index.html>]

Graciela Alonso, Silvana Asteggiante. “*Las metáforas en la interacción hombre-computador*”. 1999.

[Disponible en: <http://www.crnti.edu.uy/05trabajos/interface/hcisil.doc>]

Jesús Lorés, Joan Manuel Gimeno. Ferran Pérdrich. “*Metáforas, estilos y paradigmas*”.

[Disponible en: <http://griho.udl.es/ipo/ipo/doc/03Metafo.doc>]

Wikipedia. “*Multimedia*”. [Disponible en: <http://es.wikipedia.org/wiki/Multimedia>]

Wikipedia. “*HAL (software)*”. [Disponible en: [http://es.wikipedia.org/wiki/HAL_\(software\)](http://es.wikipedia.org/wiki/HAL_(software))]

Wikipedia. “*Mesa 3D*”. [Disponible en: http://es.wikipedia.org/wiki/Mesa_3D]

Wikipedia. “*Scalable Vector Graphics*”. [Disponible en: <http://es.wikipedia.org/wiki/.svg>]

Wikipedia. “*X3D*”. [Disponible en: <http://es.wikipedia.org/wiki/X3D>]

Wikipedia. “*XML*”. [Disponible en: <http://es.wikipedia.org/wiki/XML>]

Web3D Consortium. “*X3D International Standards*”.

[Disponible en: <http://www.web3d.org/x3d/specifications/>]

Web3D Consortium. “*Public X3D Wiki*”.

[Disponible en: http://www.web3d.org/x3d/wiki/index.php/Main_Page]

World Wide Web Consortium (W3C). “*Scalable Vector Graphics (SVG)*”. [Disponible en:

<http://www.w3.org/Graphics/SVG/>]

Wikipedia. “Vim”. [Disponible en: <http://es.wikipedia.org/wiki/Vim>]

Kevin E. Martin. “Horizonte Gráfico: AIGLX y el surgimiento del escritorio compuesto”.
[Disponible en: www.linux-magazine.es/issue/21/018-020_AiglxLM21.crop.pdf]

AIPO. “Asociación Interacción Persona Ordenador”. [Disponible en: <http://www.aipo.es/>]

Wikipedia. “Infografía”. [Disponible en: http://es.wikipedia.org/wiki/Dise%C3%B1o_Infogr%C3%A1fico]

ANEXOS

Tablas

Tabla VII. Administradores de ventanas para X Window System más populares

Nombre	Descripción
AfterStep	Basado en FVWM y de apariencia similar a NeXTSTEP
AmiWM	(Amiga Window Manager)
Blackbox	
CTWM.	
Enlightenment	(También llamado 'E'), basado originalmente en fvwm2.
Fluxbox	Derivado de la versión 0.61.1 de Blackbox
FVWM	
FVWM95	Versión modificada de fvwm2.x para que tome el aspecto de Windows 95.
IceWM	
Ion WM	
Kwin,	Administrador de ventanas de KDE.
Metacity	El administrador de ventanas ligero de algunas versiones de GNOME 2.
Metisse	Administrador de ventanas en 3D basado en otro gestor de ventanas, FVWM),
MWM	(Motif Window Manager).
OLWM/OLVWM	(OpenLook Window Manager / OpenLook Virtual Window Manager).
quartz-wm	Gestor de ventanas de Apple, de aspecto similar a Aqua, para el sistema X Window (X11) en Mac OS X.
Sawfish	originalmente conocido como Sawmill.
SCWM	
TWM/ VTWM	(Tab Window Manager, también llamado Tom's Window Manager / Virtual TWM)
WindowMaker	Emula la interfáz de NeXT, como AfterStep
wm2/wmx	

Figuras

Figura 1: La figura muestra una imagen del entorno de escritorio Windows Vista con su interfaz Aero.

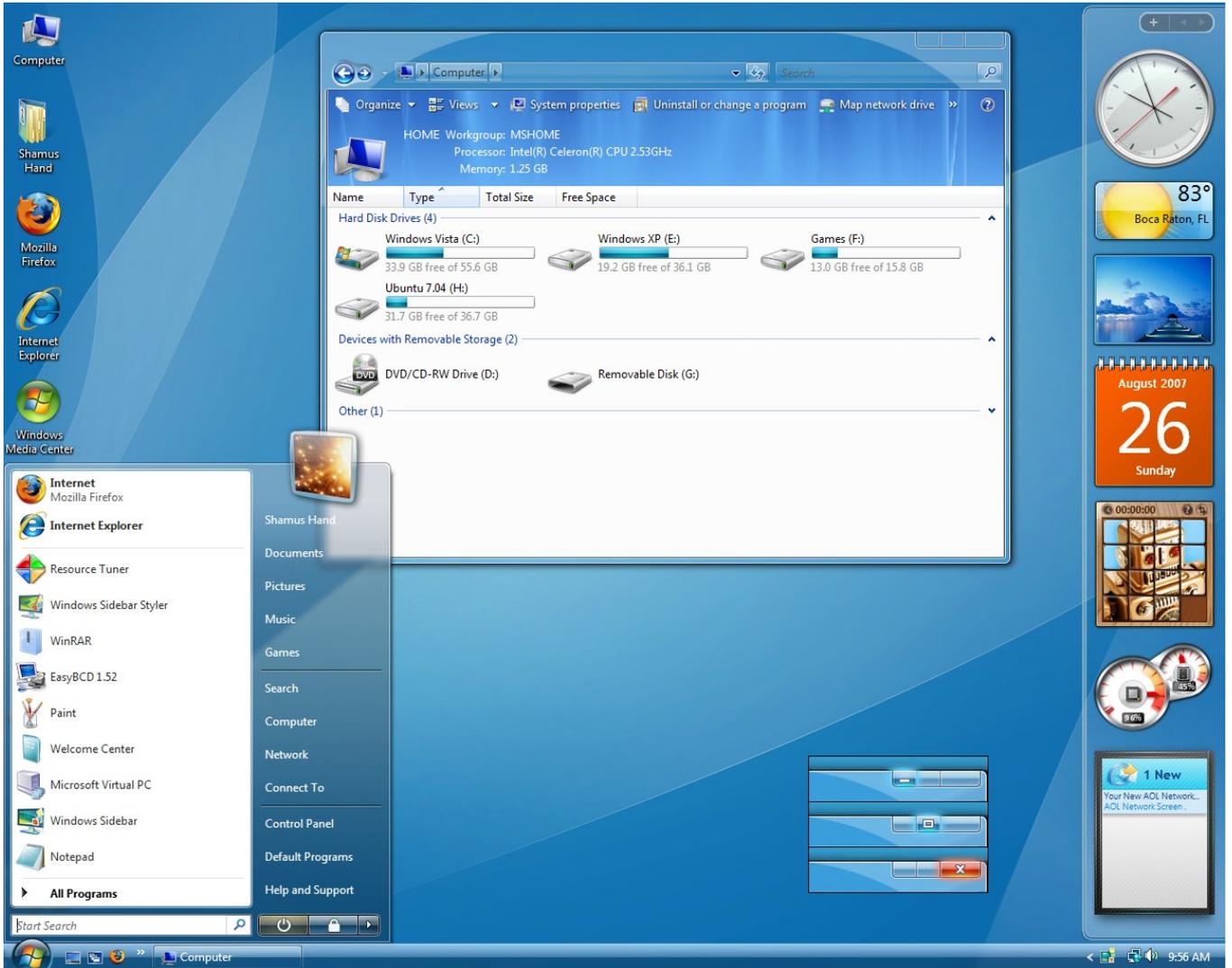


Figura 2: En la figura, imagen que muestra el entorno de escritorio del sistema Mac OS X con su interfaz Aqua.



Figura 3: La figura muestra una imagen del sistema X Window System con su entorno de escritorio KDE.



GLOSARIO DE TÉRMINOS

Personalización

Adaptación de un producto, servicio o contenido a una persona o usuario, en función de sus características, preferencias personales o información previa que proporciona.

Xerox Star

Primer sistema comercial en incorporar varias tecnologías que han llegado a ser hoy en día corrientes en computadores personales, incluyendo la pantalla con bitmaps en lugar de solo texto, una interface gráfica de usuario basada en ventanas, íconos, carpetas, ratón, red Ethernet, servidores de archivos, servidores de impresoras y e-mail.

Macintosh

Nombre con el que actualmente se refiere a cualquier computadora personal diseñada, desarrollada, construida y comercializada por Apple Inc.

Realidad Virtual

Sistema o interfaz informático que genera entornos sintéticos en tiempo real, representación de las cosas a través de medios electrónicos o representaciones de la realidad sin soporte objetivo, ya que existe sólo dentro del ordenador.

Interfaz adaptativa

Interfaces que se adaptan a diferencias o cambios que existen o pueden tener lugar en la población de usuarios de un sistema informático .

Microprocesador

Circuito integrado que contiene algunos o todos los elementos necesarios para conformar una (o más) "unidad central de procesamiento" UCP, también conocido como CPU (por sus siglas en inglés: Central Process Unit).

Compiz

Uno de los primeros gestores de composición de ventanas para el sistema de ventanas X Window, que fue capaz de aprovechar la aceleración OpenGL. También presenta algunas de las características que tienen otros competidores, como Exposé en Mac OS X de Apple.

Compiz Fusion

Proyecto relacionado con Compiz que resulta de la unión entre el paquete "Compiz Extras" de Compiz y los plugins del proyecto Beryl, que era independiente del núcleo del gestor de ventanas.

Beryl

Nombre de un proyecto nuevo para la rama Quininstorm de Compiz, anunciado el 19 de septiembre de 2006, en el foro de Compiz.net, después de que Quininstorm y el equipo de desarrollo decidieron que la bifurcación se había alejado demasiado de Compiz-vanilla, la rama de Compiz iniciada por Novell.

Aqua

Nombre comercial de la apariencia de la interfaz gráfica de usuario del sistema operativo Mac OS X de Apple Computer.

Quartz

Conjunto de tecnologías asociadas a la interfaz gráfica de Mac OSX.

Windows Aero

Nueva interfaz gráfica incluida en Windows Vista que sustituye a la Interfaz gráfica Luna utilizada en Windows XP.

Mesa3D

Biblioteca gráfica de código abierto, que proporciona una puesta en práctica genérica de OpenGL para renderizar gráficos tridimensionales en plataformas múltiples.

DRI

(**D**irect **R**endering **I**nfrastructure) Es un interfaz usada en el sistema de ventanas X para que las aplicaciones de usuario puedan acceder de manera segura al hardware de vídeo sin tener que pasar los datos por el servidor X.

GLX

Extensión del protocolo X que permite a los programas que desean utilizar OpenGL hacerlo dentro de una ventana proporcionada por X Window System.

XGL

Arquitectura X server para OpenGL que tiene como propósito aprovechar las ventajas de las modernas tarjetas gráficas.

KDE

(**K** **D**esktop **E**nvironment) Entorno de escritorio e infraestructura de desarrollo para sistemas Unix/Linux implementado para parecerse a Windows.

GNOME

(**G**NU **N**etwork **O**bject **M**odel **E**nvironment) Entorno de escritorio e infraestructura de desarrollo para sistemas operativos Unix/Linux, compuesto enteramente de software libre.

Xfce

Entorno de escritorio ligero para sistemas tipo Unix como Linux, BSD, Solaris y derivados.

Antialiasing

Algoritmo que permite evitar el aliasing gráfico.

Aliasing

Efecto que causa que señales continuas distintas se tornen indistinguibles cuando se les muestrea digitalmente. Cuando esto sucede, la señal original no puede ser reconstruida de forma unívoca a partir de la señal digital.

Adobe Director

Aplicación desarrollada por Adobe, que permite generar presentaciones multimedia (en archivos ejecutables, por ejemplo).

Framework

Estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

API

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.

Terminal virtual

Es un servicio de aplicación que permite utilizar varios programas simultáneamente .

Unix

Sistema operativo portable, multitarea y multiusuario desarrollado por un grupo de empleados de los laboratorios Bell de AT&T.

TCP/IP

Familia de protocolos que sirven para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN) y área extensa (WAN).

BSD

Son las iniciales de Berkeley Software Distribution (en español, Distribución de Software

Berkeley) y se utiliza para identificar un sistema operativo derivado del sistema Unix nacido a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley.

PostScript

Lenguaje de Descripción de Página (en inglés PDL, Page Description Language), utilizado en muchas impresoras y, de manera usual, como formato de transporte de archivos gráficos en talleres de impresión profesional.

Buffer

Es una ubicación de la memoria en una computadora o en un instrumento digital reservada para el almacenamiento temporal de información digital mientras espera ser procesada.

Síncrono

Que tiene un intervalo de tiempo constante entre cada evento. Son procesos síncronos los que dependen de un acontecimiento externo que los dispara.

Asíncrono

Que no tiene un intervalo de tiempo constante entre cada evento.

Binding

Adaptación de una biblioteca para ser usada en un lenguaje de programación distinto de aquel en el que ha sido escrita.

Opacidad

Es una propiedad óptica de la materia resultado de no dejar pasar la luz en proporción apreciable.

Open Group

Consorcio de la industria del software que provee estándares abiertos para la

infraestructura de la informática como el X Window System o sistema de ventanas X.

TocoPan

Es una idea de panel o comúnmente barra de tareas, capas de adoptar diferentes formas visuales. El nombre es una contracción de los términos **Tocororo** y **Panel**, que evoca a la mascota cubana de los juegos panamericanos del año 1991.

Gnus

Es la aplicación que permite diseñar un tema de panel TocoPan, e incluye la definición de su geometría. Su nombre es una evocación del término GNU en plural y significa marcar la diferencia.

XSLT

Estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML.

Infografía

Representación más visual que la propia de los textos, en la que intervienen descripciones, narraciones o interpretaciones, presentadas de manera gráfica normalmente figurativa, que pueden o no coincidir con grafismos abstractos y/o sonidos.