

Universidad de las Ciencias Informáticas

Facultad 10



**Título: Estudio sobre Herramientas de Diseño en
Software Libre.**

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

Autor: Vladimir Molina Pérez

Tutora: Lic. Yenisleydi Cariaga Cristo

Ciudad de la Habana, Junio 2007

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 exclusivo.

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

Vladimir Molina Pérez

Lic. Yenisleydi Cariaga Cristo

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

R. M. Stallman

AGRADECIMIENTOS

A la Revolución, por darme la posibilidad de estar aquí.

A mi tutora, por la confianza y el optimismo.

A Nurys, por su apoyo y cariño.

DEDICATORIA

A mis padres, por el amor, el sacrificio y la dedicación.

A mi hermana, crece sobre mis hombros.

A mis amigos, dondequiera que estén, por los buenos y malos consejos.

A la Comunidad de Software Libre.

RESUMEN

Actualmente la mayoría del software que se utiliza en nuestro país es privativo, existiendo un desconocimiento generalizado acerca de las herramientas libres. Esta investigación se centra particularmente en el área del diseño, motivada por la falta de guías para los usuarios enfocadas a una eventual migración de Software Privativo a Software Libre. Para el trabajo se escogieron Blender y Gimp por el desarrollo y calidad de estas aplicaciones. Se realiza una investigación sobre cada herramienta por separado, exponiendo sus características, y en cada caso como se pueden extender sus funcionalidades mediante lenguajes y técnicas de programación. Se ofrece, además, una comparación entre las herramientas seleccionadas con algunas de sus contrapartes del Software Privativo, exponiendo de forma crítica sus ventajas y desventajas. Como resultado del trabajo se obtuvo un estudio sobre Blender y Gimp, ofreciendo una caracterización de estos programas que puede utilizarse como guía en el momento de la migración de plataforma.

Palabras Claves:

Blender, Diseño, Gimp, Migración, Programación, Software Libre, Software Privativo.

AGRADECIMIENTOS	I
DEDICATORIA.....	II
RESUMEN.....	III
INTRODUCCIÓN.....	1
CAPÍTULO 1: CARACTERÍSTICAS GENERALES DE LAS HERRAMIENTAS.	7
1.1 FUNDAMENTACIÓN DEL TEMA	7
1.2 MÉTODO CIENTÍFICO DE INVESTIGACIÓN	8
1.3 BLENDER. CARACTERÍSTICAS GENERALES.....	9
<i>¿Qué es Blender?.....</i>	<i>9</i>
<i>Características.....</i>	<i>10</i>
<i>Historia.....</i>	<i>12</i>
<i>Comparación entre Blender y 3D Max.....</i>	<i>13</i>
1.4 GIMP. CARACTERÍSTICAS GENERALES	18
<i>¿Qué es GIMP?.....</i>	<i>18</i>
<i>Características.....</i>	<i>19</i>
<i>Historia.....</i>	<i>20</i>
<i>Gimp como alternativa a Adobe Photoshop</i>	<i>22</i>
CAPÍTULO 2: LA PROGRAMACIÓN.....	25
2.1 LA PROGRAMACIÓN APLICADA A HERRAMIENTAS LIBRES	25
2.2 SCRIPTS Y PLUGINS	26
2.3 PYTHON	27
2.4 EXTENDIENDO BLENDER	29
2.5 EXTENDIENDO GIMP	33
2.6 VENTAJAS DEL USO DE SCRIPTS Y PLUGINS.....	38
2.7 EQUIPOS DE TRABAJO	39
CONCLUSIONES.....	40
RECOMENDACIONES.....	41
REFERENCIAS BIBLIOGRÁFICAS.....	42
BIBLIOGRAFÍA.....	44
ANEXOS	48

INTRODUCCIÓN

Desde el triunfo de la Revolución Cubana nuestro gobierno ha trabajado por lograr una sociedad mejor, en la cual todos tengan las mismas oportunidades, derechos y deberes, una sociedad que sea culta, que se encuentre en la vanguardia científico-técnica a nivel mundial. La erradicación del analfabetismo, los avances en la salud, el impulso del movimiento deportivo cubano son algunos ejemplos de esta política estatal. Desde el punto de vista científico se destaca el desarrollo en el área de la medicina y los innegables logros en la biotecnología. En estos días resulta común hablar de Cultura General Integral o de la Informatización de la Sociedad

Cuando el país se alista para un cambio tecnológico tan importante como es la informatización de la sociedad y llevar las nuevas tecnologías de la informática y las telecomunicaciones a todos los espacios del ámbito nacional es necesario pensar en aspectos técnicos como los requerimientos y posibilidades económicas desde el punto de vista del hardware y las opciones en cuanto al software a utilizar.

Actualmente en nuestro país se utiliza generalmente como sistema operativo alguna versión de Microsoft Windows, mientras que la mayoría de las aplicaciones en la educación, la salud o la producción de bienes y servicios son de tipo privativo. Por cuestiones del bloqueo Cuba no paga gran parte de las Licencias de estos programas, lo que trae como consecuencia que muchos estén desactualizados o que las empresas nacionales no puedan exportar sus productos.

La piratería de software afecta directamente al software libre ya que la mayoría de los usuarios que adquieren software pirata no se sienten atraídos por los beneficios del software libre. Estos usuarios tienen motivos netamente utilitarios para el software y no les interesa tener acceso al código, ni les preocupa la legalidad de sus acciones. [1]

La respuesta a esta situación es el uso de Software Libre. La idea de la migración en nuestro país no es nueva, ya desde el año 2004 se expuso la necesidad del cambio y se manejaron fechas - tal vez más entusiastas que reales - para la migración a la plataforma GNU/Linux, pero todavía hasta la fecha no se vislumbra el cambio de plataforma. Es lógico que algo nuevo -y con la fama que lo precede- como

GNU/Linux provoque dudas y rechazos, por eso la nueva tarea es luchar contra el desconocimiento de las posibilidades y herramientas que alberga el Software Libre. Es necesario demostrar el constante avance y desarrollo del proyecto GNU/Linux y de las distintas aplicaciones que existen en esta plataforma.

El desarrollo tecnológico ha dado paso a una gran evolución en el diseño gráfico, facilitando herramientas para la creación de diseños innovadores y de gran originalidad. Al respecto, cabe destacar el uso de programas como Photoshop y Coreldraw, excelentes para el tratamiento de imágenes, ambos ampliamente extendidos y populares. También son productos de reconocida calidad el 3DStudio Max y Maya Alias Wavefront, ambas herramientas para la realización de gráficos y animaciones en 3D.

Aunque todas estas son excelentes opciones para trabajos de diseño cabe señalar que son privativas, y que por eso tienen algunas desventajas propias de este tipo de productos, como son: el elevado coste de las licencias. Las compañías que las desarrollan tienen el derecho exclusivo de innovación, es ilegal realizar copias sin una licencia para ello, grandes requerimientos de hardware y en ocasiones estos productos no se ajustan a las necesidades de la Empresa, o es necesario más de un software para completar un trabajo.

Actualmente se está desarrollando un fuerte movimiento a nivel mundial pro-Software Libre, siendo adoptado por administraciones estatales, regionales, educativas y empresariales de todas partes del mundo. GNU/Linux comienza a ganar terreno en todos los campos que pueden incluir el uso de computadoras y por ende en el diseño digital. Aunque GNU/Linux cuenta con innegables ventajas como su libre uso, su adaptabilidad, los bajos requerimientos de hardware de la mayoría de las aplicaciones y su bajo costo también tiene desventajas: su elevada curva de aprendizaje y el hecho que la mayoría de la configuración del hardware no sea intuitiva son algunas de las más importantes.

Entre las ventajas mencionadas está la adaptabilidad de GNU/Linux y de las aplicaciones que se desarrollan para esta plataforma. El hecho es que al disponer del código se pueden realizar los cambios que sean necesarios para que la herramienta se ajuste mejor a las necesidades personales o de la empresa, y con ello mejorar la herramienta en si. El uso de scripts es un recurso ampliamente utilizado, pues permite incluir funcionalidades de acuerdo a nuestras necesidades que no estaban en el producto

original. De acuerdo a la filosofía que rige al Software Libre estos scripts se pueden descargar gratuitamente desde la red, donde la comunidad los publica para el uso común de todos.

Algunas de las alternativas que brinda el Software Libre para las tareas relacionados con el Diseño Gráfico son: Blender, Gimp, Inkscape y Cinelerra. En este trabajo se van a tratar dos herramientas en particular, el Blender, para diseño y animación 3D y el Gimp, programa de manipulación de imágenes de GNU. Se han escogido estas dos herramientas porque actualmente son las que más se han destacado en el mundo del diseño dentro del Software Libre. Han demostrando como se pueden desarrollar aplicaciones bajo este modelo que puedan ofrecer todas las prestaciones que necesita un usuario final y tienen una comunidad de usuarios que exhiben resultados que demuestran la calidad y posibilidades de estos programas.

La importancia del estudio radica en la posibilidad de explorar el mundo del Diseño en la plataforma GNU/Linux, que encarna no solo un cambio tecnológico, sino que además constituye un profundo cambio social, pues se basa en una filosofía de intercambio y solidaridad, donde el conocimiento y los adelantos científicos están al alcance de todos.

Es el momento de comenzar a pensar y trabajar en el tema de la migración seriamente, antes de que el país tenga que pagar sumas millonarias por las licencias de los productos de software que se utilizan. Por otra parte, migrar a Software Libre de forma abrupta tampoco sería lo mejor. Esta última solución traería más pérdidas económicas en un periodo a corto plazo que todos los gastos de una migración planificada, al tiempo que sería mucho más difícil el proceso de adaptación de los factores implicados.

Desde el punto de vista económico el Software Libre también puede traer independencia tecnológica. Este trabajo podría contribuir con la creación de una comunidad que acogiera a todos los grupos y personas que trabajan hoy con el software libre en el país.

Una comunidad de Linux sería una fuente segura de desarrollo tecnológico, pues el modelo de trabajo del Software Libre es compartir la información de modo cooperativo, y es de esta forma que la humanidad ha innovado y avanzado. El hecho de que varias personas de edad, sexo, preferencias y localización geográfica diferente compartan el trabajo necesariamente debe conducir a un producto más refinado y con

más calidad, al tiempo que contribuye a la informatización de la sociedad. Este desarrollo comunitario posibilitaría la creación de nuevos puestos de trabajo, pues donde antes solo se requería una persona con conocimientos para operar un determinado software se necesitaría todo un equipo de diseñadores, programadores, ingenieros de software, etc.

Por esto, atendiendo a la necesidad de dar a conocer las potencialidades de las herramientas de diseño y las posibilidad que brinda el modelo de desarrollo del Software Libre de modificarlas, hacerlas más amigables y añadirle funcionales, se presenta el siguiente problema a resolver.

Problema: no existen guías de estudio para orientar a diseñadores, programadores y entidades que trabajen el diseño gráfico y digital en la migración de herramientas propietarias a herramientas libres. También existe desconocimiento sobre como aplicar técnicas de programación a dichas herramientas para extender sus funcionalidades.

Idea a defender: Atendiendo a la inminente migración tecnológica a Software Libre se hace más factible el trabajo con herramientas de diseño y edición desarrolladas según esta filosofía.

Objeto de Estudio: Las Herramientas Informáticas de Diseño sobre la plataforma libre.

Objetivo de la Investigación: Conocer sobre las herramientas del software libre para los trabajos de diseño y su relación con la programación.

Objetivos Específicos:

- Investigar sobre el uso de las herramientas de diseño sobre Software Libre en la actualidad.
- Caracterizar Blender como herramienta de diseño
- Caracterizar GIMP como herramienta de diseño.
- Estudiar de la programación aplicada a estas herramientas.
- Redactar una posible guía de estudio sobre las herramientas.

Campo de Acción: Las Herramientas de Diseño de Software Libre que se vinculan con el Diseño Gráfico.

Tareas de la Investigación:

- Realizar una búsqueda bibliográfica sobre el tema del Diseño Gráfico en el Software Libre.
- Investigar acerca de Blender.
- Investigar acerca del Gimp.
- Investigar como vincular cada herramienta con la programación.
- Proporcionar una guía para la posible migración a herramientas libres y sobre el uso de la programación aplicada a estas.

Estructura del Trabajo

Capítulo 1: Fundamentación del tema. Características generales de cada una de las herramientas. Se exponen las características de cada programa por separado y se realiza una comparación con sus contrapartes de software privativo.

Capítulo 2: La programación aplicada a las herramientas de diseño. Python, Plug-ins y Scripts. Se muestra como extender las funcionalidades de las herramientas por separado, resaltando la facilidad y la utilidad del uso y creación de añadidos.

Conclusiones

Recomendaciones

Referencias Bibliográficas

Bibliografía

Anexos

CAPÍTULO 1: CARACTERÍSTICAS GENERALES DE LAS HERRAMIENTAS.

1.1 Fundamentación del tema

En el universo digital existen muchas variantes y posibilidades, sin embargo la mayor parte del software hoy en día responde a intereses económicos, políticos, sociales y culturales de unas pocas empresas y países. Los Estados Unidos controlan el 50 por ciento de los satélites de comunicación y el 75 por ciento de Internet. Producen el 60 por ciento del software de uso mundial y una sola compañía, Microsoft, domina con Windows, el sistema operativo instalado en más del 90 por ciento de las computadoras personales. Actualmente el 40 por ciento de los usuarios se concentran en este país, y el 80 por ciento de los contenidos difundidos en las páginas web están en inglés. Igualmente dominan el 85 por ciento del comercio electrónico. [2]

A causa del Bloqueo nuestro país no puede acceder legalmente a programas informáticos de la más diversa índole, y ante la necesidad de desarrollar la infraestructura económico-social de la nación nuestro gobierno se encontró forzado a informatizar nuestra sociedad con un producto muchas veces pirata o que no es el óptimo, ajeno a nuestra cultura y nuestras necesidades, sin posibilidades de recibir servicios técnicos ni disponer de actualizaciones.

¿Qué es Diseño?

El diseño se aplica, en todos los ámbitos y se encuentra todas partes. Dentro del mundo digital, en la red, Internet. Se encuentra diseño en el mobiliario, una silla, una puerta, dentro del mundo publicitario y comercial, revistas, periódicos, libros, manuales. En Internet, en las multimedias, televisión, videos, en el cine y sus efectos especiales.

Es posible definir como diseño todo pensamiento o idea, que es posible plasmar y llevar a cabo físicamente, a través de una metodología.

Se puede definir el Diseño Gráfico como el proceso o actividad que tiene como objetivo proyectar, coordinar, seleccionar y organizar un conjunto de elementos para crear y producir objetos visuales destinados a comunicar mensajes específicos a grupos determinados. [3]

El término Diseño Digital se integra a la evolución de la expresión gráfica contemporánea, a partir del gran avance de los medios tecnológicos, que genera la digitalización de la imagen. El Diseño Digital propone profundizar en el conocimiento de los sistemas visuales o los "entornos virtuales" que abarquen, no sólo los aspectos tradicionales de composición de la imagen y sus respectivos parámetros de percepción, sino también las múltiples posibilidades de la interactividad.

1.2 Método Científico de Investigación

Método Sistémico: Se estudia cada aplicación independientemente y la forma en que la programación se aplica a dicha herramienta.

Métodos Empíricos

Entrevista: es posible utilizar más de un tipo de entrevista, en dependencia de las necesidades propias de la investigación. En el trabajo se empleará la entrevista informal.

Entrevista informal: consiste en una conversación que sostiene el investigador con uno o varios de los miembros del grupo investigado u otra persona conocedora de la materia. Esta técnica tiene como ventaja principal, el hecho de que la conversación se realiza en un ambiente de amistad de no como una conversación de trabajo, lo que posibilitará conocer más a fondo los criterios de las personas acerca del tema.

Se utiliza para conocer el estado de opinión sobre el Software Libre en General, las herramientas del estudio y su relación con la programación. Se aplicará en grupos de discusión con y sin conocimientos o experiencias previas con el Software Libre.

Análisis Documental. Se realiza una investigación en Internet para determinar el nivel de aceptación de las herramientas así como las características y resultados de las mismas. Se presta atención a la confiabilidad de las fuentes y datos consultados.

1.3 Blender. Características generales

¿Qué es Blender?

Blender (Ver Figura 1 Anexos) es un programa de código abierto que integra una serie de herramientas para el modelado 3D, animación, renderizado, post producción y creación de videojuegos. Disponible para la mayoría de los sistemas operativos bajo los términos de la GNU Public License. El programa se mantiene gracias a la colaboración desinteresada de muchos programadores y voluntarios de distintas áreas.

La interfaz gráfica de usuario del Blender es muy peculiar, diferente a la de otros programas, como 3D Max o Ligthwave. Aunque se critica como poco intuitiva, pues no se basa en el sistema clásico de ventanas, tiene a su vez ventajas importantes sobre estas, como la configuración personalizada de la distribución de los menús y vistas de cámara.

Blender también es un buen editor de vídeo. Soporta los formatos clásicos AVI y AVIJPG además de permitir agregar imágenes estáticas TGA o JPG. Una característica muy útil que fue añadida al poco tiempo que el código fue "abierto" es la de añadir, editar y sincronizar pistas de audio con las animaciones.

Esta herramienta es todavía nueva y, a pesar de que avanza con muy buen paso no se ha usado en grandes producciones para generar secuencias CGI (Imagen generada por ordenador), aunque existen proyectos actuales que han empezado a usarlo profesionalmente:

- Películas tales como "Spiderman 2" que lo ha usado para hacer una pre-visualización de escenas (Screen-Board Test), han usado de manera incipiente las capacidades del Blender.

- Algunas propuestas más llevadas a la producción e integración con gráficos mediante Motion Track tales como “Friday or another day”.
- Otros proyectos hechos como “Elephants Dream” (el primer cortometraje realizado en software libre) son experimentos de sus capacidades, extendidas gracias a la posibilidad de poder editar su código fuente.
- En estos momentos se trabaja en el primer largometraje, donde solo se utilizarán herramientas libres, principalmente Blender. Plumíferos es el nombre de la producción y se anuncia su estreno para finales del 2007.

Blender aún es una aplicación poco conocida, sin embargo tiene las condiciones para un desarrollo sostenido y, en un futuro no muy lejano, su inclusión dentro de los ámbitos profesionales más reacios a abandonar el uso de las herramientas tradicionales donde ya comienza a ganar terreno. Blender está llamado a convertirse en otra de las aplicaciones de referencia dentro del mundo del Software Libre.

Características

Blender es multiplataforma, libre, gratuito y con un tamaño de origen realmente pequeño comparado con otros paquetes de 3D, dependiendo del sistema operativo en el que se ejecuta.

Interfaz

- Interfaz flexible y configurable.
- Idéntica en todas las plataformas.
- Internacionalización parcial.
- Dividida en ventanas.
- Editor de scripts integrado.

Modelado

- Variedad de objetos 3D: Mallas poligonales, superficies NURBS, curvas Bezier y B-Spline, metaballs, fuentes vectoriales (TrueType, PostScript, OpenType), SubSurf.
- Funciones de edición: extrude, spin, subdivide, noise, smooth y funciones booleanas para mallas.
- Scripting Python > Extensibilidad.

Animación

- Armatures con cinemática directa e inversa, auto skinning y weight paint.
- Animación no-lineal y walkcycles sobre paths.
- Sistema de constraints.
- Sincronización de sonido.
- Scripting Python.

Render

- Render scanline rápido y radiosidad integrados
- Oversampling, motion blur, efectos post-produccion, fields, mapas de entorno, halos, lens.
- Scripts python para varios motores de render externos: Renderman, VirtualLight, PovRay, Yafray (actualmente integrado).

Creación de Video Juegos

- Editor lógico para definir el comportamiento interactivo sin necesidad de programar.
- Detección de colisiones y simulaciones dinámicas soportadas por *Bullet Physics Library*. Bullet una librería de código abierto para la detección de colisiones y dinámicas de cuerpos rígidos desarrollada para Play Station 3.
- Soporte completo para dinámicas de vehículos.
- API de python para un control sofisticado e IA, lógica de juego avanzada completamente definida.

Plataformas soportadas por Blender

- Windows (98/ME/2000/XP/Vista)
- Linux (i386)
- Mac OS X (PPC, Intel)
- Linux (PPC)
- Sun Solaris 2.8 (Sparc)
- FreeBSD 5.4 (i386)
- SGI IRIX 6.5

Historia

En un principio, el programa fue desarrollado como una aplicación propia por el estudio de animación holandés NeoGeo. Ton Roosendaal, el principal autor del software fundó la empresa "Not a Number Technologies" (NaN) en junio de 1998 para desarrollar y distribuir el programa. En la base de NaN, estaba el deseo de crear y distribuir gratuitamente una suite de creación 3D compacta y multiplataforma. En ese momento, esto fue un concepto revolucionario ya que la mayoría de los programas comerciales de modelado costaban miles de dólares. NaN esperaba conseguir una herramienta de modelado y animación de un nivel profesional al alcance del público en general. El modelo de negocio de NaN consistía en proporcionar productos comerciales y servicios alrededor de Blender. Desafortunadamente, las ambiciones y oportunidades de NaN no coincidieron con las capacidades de la compañía ni con la realidad del mercado de la época. Debido a las decepcionantes ventas y al continuo clima de dificultades económicas, los inversores decidieron dar por terminadas las actividades de NaN. Esto también incluía parar el desarrollo de Blender. En este contexto, negándose a permitir que Blender quedara en el olvido y ante la realidad de que relanzar una nueva compañía con un equipo suficientemente grande de desarrolladores no era factible, en marzo de 2002, Ton Roosendaal fundó la organización no lucrativa *Blender Foundation* (Fundación Blender). El primer objetivo de la Fundación Blender fue encontrar una manera de continuar el desarrollo y la promoción de Blender como un proyecto de código abierto basado

en la comunidad de usuarios. En julio de 2002, Ton logró obtener de los inversores de NaN la aprobación para que la Fundación Blender llevara a cabo su plan de que Blender fuera código abierto. Eran necesarios 100.000 EUR para que la Fundación pudiese comprar los derechos del código fuente y los de propiedad intelectual de Blender a los inversores de NaN y, posteriormente, liberar Blender a la comunidad de código abierto. Con un entusiasta grupo de voluntarios -entre los que se encontraban varios ex-empleados de NaN- fue lanzada la campaña de "Liberad a Blender", que alcanzó el objetivo en tan sólo 7 semanas. El domingo 13 de octubre de 2002, Blender fue liberado al mundo bajo los términos de la Licencia Pública General de GNU (GPL). [4]

Comparación entre Blender y 3D Max.

Requerimientos de Hardware

Aspectos	Blender [5]		3D Max [6]	
	Mínimo	Óptimo	Mínimo	Óptimo
CPU	300 MHz	2 GHz	Intel® Pentium® IV, AMD o superior.	Intel® Pentium® IV, AMD o superior.
RAM	128 MB	2 GB	512 MB	1 GB
Espacio en Disco	20 MB	-	500 MB	2 GB
Mouse	De 3 botones	De 3 botones	Intellimouse o similar.	Intellimouse o similar.
Tarjeta Gráfica	16 MB	128 o 256 MB	64 MB	256 MB

Detalles Técnicos

Herramienta	Plataforma	Soporte de Video	Lenguajes de Extensión	Precio con Funcionalidades Básicas
3D Max	Windows 32/64 Bit	Direct X, Software Rendering, OpenGL, Heidi	MAXScript	\$ 3,495.00
Blender	Windows, OSX, Linux, FreeBSD, Irix, Solaris	Software Rendering, OpenGL	C, Python	SWL (GPLv2) \$ 0.00

Herramientas de Modelado

Herramienta	NURBS	Patch	SubD	Polygon	Smooths 3gons	Smooths ngons
3D Max	Sí	Sí	Sí	Sí	Sí	Sí
Blender	Sí	No	Sí	Sí	Sí	No

Texturizado y Pintura

Herramienta	Texturizado Basado en Nodos	Pintura 2D	Pintura 3D	Pintura de Vértices
3D Max	Sí	Plugin	Plugin	Sí
Blender	Sí	Sí	Sí	Sí

Renderizado

Herramienta	Motor de Render Primario	Otros Motores de Render	Render en Red	# de Nodos de Red
3D Max	Scanline, Mental Ray, Light Tracer, Radisosity Engine (Lightscape)	Brazil r/s, VRay, finalRender, finalToon, Illustrate!	Sí	Unlimited
Blender	Blender	YafRay, Mental Ray, Sunflow, Kerkythea, Indigo	Via Plugin	Unlimited

Formatos de Archivo soportados.

Importar

Herramienta	OBJ	3DS	DXF	FBX	LWO	Q3D	VRML	MB	MA	x
3D Max	Sí	Sí	Sí	Sí	?	?	Sí	No	No	Plugin
Blender	Sí	Sí	Sí	No	Sí	No	Sí	No	No	Sí

Importar (continuación)

Herramienta	Collada1.4	JPEG	TIFF	PSD	AEP	BMP	GIF	PNG	EXR	HDR
3D Max	Sí	Sí	Sí	Sí	?	Sí	Sí	Sí	Sí	Sí
Blender	Sí	Sí	Sí	Sí	No	Sí	Sí	Sí	Sí	Sí

Exportar

Herramienta	OBJ	3DS	DXF	FBX	LWO	Q3D	VRML	MB	MA	x
3D Max	Sí	Sí	Sí	Sí	Plugin	?	Sí	No	No	Plugin
Blender	Sí	Sí	Sí	Sí	Sí	No	Sí	No	No	Sí

Exportar (continuación)

Herramienta	Collada 1.4	JPEG	TIFF	PSD	AEP	BMP	GIF	PNG	EXR	HDR
3D Max	Sí	Sí	Sí	Sí	?	Sí	Sí	Sí	Sí	Sí
Blender	Sí	Sí	Sí	Sí	No	Sí	Sí	Sí	Sí	Sí

Como puede apreciarse Blender tiene requerimientos de hardware menos exigentes que el 3D Max, sin embargo esto significa que sea una herramienta menos potente. Las funcionalidades de Blender están al nivel de cualquier otra aplicación comercial. Se puede importar trabajos desde otras aplicaciones y salvarlos en formatos propios de otras herramientas, facilitando así la usabilidad y el trabajo con diferentes programas. Se pueden realizar los mismos trabajos que con otras herramientas de diseño, al tiempo que brinda otras ventajas propias del software libre.

La comunidad de Blender se desarrolla rápidamente y es muy activa. En la Web se puede localizar y descargar documentación de cualquier tipo sin costo alguno, teniendo además los foros donde también se puede encontrar ayuda. Otra ventaja es que no hay que pagar una licencia, que en aplicaciones equivalentes como el 3D Max resultan muy costosas.

Según la revisión del tema en Internet la opinión general es que Blender puede ser utilizado seriamente para trabajos 3D, refiriéndose con énfasis a su motor de juegos y su fortaleza en el modelado. Los grupos de usuarios se dividen a la hora de proponer una herramienta basándose principalmente en la experiencia y conocimientos que poseen sobre determinada aplicación, mostrándose en ocasiones verdaderas “batallas” entre los usuarios de los distintos programas. Al ser Blender una aplicación relativamente nueva

se observa menos su presencia en el mundo del diseño, sin embargo está ganado terreno en este campo día a día.

Uno de los problemas de Blender es que los diseñadores que trabajan el tema del diseño 3D generalmente están acostumbrados a 3D Max o algún otro programa, y los primeros encuentros con la peculiar interfaz de Blender suelen hacerlos desistir de aprender a utilizarlo. El diseñador la mayoría de las veces va a preferir continuar con su forma de trabajar antes que asimilar un nuevo modo de hacer las cosas, y va a criticar el programa sin siquiera conocerlo.

Esta característica de Blender también ofrece ventajas, una de ellas es que resulta más ligero que aplicaciones similares, lo mismo en tamaño del instalador que a la hora de correr la aplicación. La interfaz, al ser la misma para todas las plataformas soportadas por Blender brinda la posibilidad de poder comenzar a utilizarlo en la educación y en la producción sin necesidad de esperar a migrar a GNU/Linux. Esto trae resultados totalmente deseables; se comienza a fomentar una cultura de uso de software libre contra software privativo (pirata en muchos casos), se preparan los usuarios para el futuro cambio de plataforma y una vez realizado este no se apreciará disminución en la producción porque serían las mismas herramientas.

Se debe señalar que se recomienda el uso de Software Libre tanto o más por razones sociales que económicas. Si bien es cierto que trae innegables beneficios a la economía nacional por cuestiones de ahorro de hardware y licencias el Software Libre encarna un cambio más profundo. Está desarrollado según el modelo de compartir y socializar la información, trayendo consigo una generalización del conocimiento y una competencia leal por alcanzar nuevas metas. Lograr que el uso del Software Libre se convierta en lo cotidiano será un nuevo reto y un paso de avance en el desarrollo de una sociedad mejor.

1.4 GIMP. Características generales

¿Qué es GIMP?

GIMP (GNU Image Manipulation Program, Programa de manipulación de imágenes GNU) (Ver Figura 2 Anexos). Es un programa de distribución gratuita para el retoque de fotografías, creación y edición de imágenes. Disponible para varios sistemas operativos e idiomas.

Gimp es un programa totalmente libre bajo licencia GPL del proyecto GNU. Esta disponible para múltiples plataformas, siendo su interfaz la misma para cada una de ellas. El Gimp ocupa poco espacio en el disco duro y realiza un buen manejo de la memoria física de la máquina. Es un programa muy completo con muchísimas opciones y que puede ser fácilmente extendido por medio de añadidos.

Puede utilizarse como un simple programa de dibujo, un programa experto de retoque fotográfico, o como un convertidor de formatos de imagen. La potencia de GIMP es similar a la de Photoshop, con la diferencia de que es Software Libre. Hoy en día GIMP es considerado una herramienta de alta calidad que logra competir con cualquier programa comercial de su género.

Una de las ventajas realmente importante del GIMP es la posibilidad de poder usar añadidos (plug-ins), los cuales permiten crear al usuario sus propias características y efectos especiales. La comunidad de GIMP ha escrito cientos de estos añadidos, y como es una herramienta “libre” los usuarios pueden conseguir estos plug-ins con facilidad del Web o escribir los propios.

La interfaz modular de GIMP puede confundir un poco al usuario novel. En vez de usar una ventana de aplicación que abarca toda la pantalla, GIMP visualiza cada elemento del interfaz como un componente por separado. Incluso la ventana de la imagen es independiente junto con las ventanas de las herramientas.

Una de las desventajas del GIMP es, la falta de soporte nativo para PANTONE y CMYK, que son los modelos de colores utilizados por las imprentas. *Pantone Inc.* es una compañía creadora de un sistema de control de colores para las artes gráficas, denominado PANTONE. Este sistema es el más utilizado en la

industria y se basa en una muestra o gama de colores. Así es siempre es posible obtener un color determinado por mezclas de tintas predeterminadas. A cada color le corresponde un único número. Según la compañía estas listas de números son propiedad intelectual, es por esta razón que no pueden ser utilizadas por GIMP (u otros programas de software libre).

Por otro lado GIMP aún no cuenta con soporte para CMYK de forma nativa, aunque posibilita salvar los trabajos en formato TIFF. Este es un formato muy versátil, basado en ficheros con etiquetas. Entre sus ventajas están que permite varios modos de color (entre ellos CMYK), varios algoritmos de compresión y varios documentos de imagen en un solo archivo.

Características

Pintura

- Completo conjunto de herramientas para dibujar, incluyendo pinceles, lápiz, aerógrafo, reproducción.
- Muestreo de sub-píxel para todas las herramientas de dibujo para antialiasing de alta calidad.

Sistema

- Gestión especial de memoria cuando el tamaño de la imagen está sólo limitado por el espacio disponible en disco.
- Virtualmente no existen límites en cuanto al número de archivos abiertos.

Manipulación Avanzada

- Soporte completo para canal Alfa.
- Capas y canales.
- Múltiple hacer/deshacer (sólo limitado por el espacio en disco).

- Herramientas de transformación incluyendo rotación, escala.
- Herramientas de selección, incluyendo rectángulo, elipse, libre, difuso, curvas bezier.

Extensible

- Base de datos procedural para llamadas a funciones internas GIMP desde programas externos, como Script-Fu.
- Capacidades de scripting avanzadas. (Scheme, Python, Perl)
- Complementos que permiten añadir fácilmente nuevos formatos de archivos y nuevos filtros de efectos.

Manejo de Archivos

- Formatos de archivos soportados: bmp, gif, jpeg, mng, pcx, pdf, png, ps, psd, svg, tiff, tga, xpm y otros.
- Carga, visualización, conversión y salvaguarda de diferentes formatos de archivos.

Historia

La historia de GIMP comienza con un proyecto de una asignatura de un par de alumnos de la universidad de Berkley, Spencer Kimball y Peter Mattis. Se propusieron hacer algo diferente a las prácticas habituales y, aunque no tenían experiencia en el mundo del diseño gráfico basado en computador, se decidieron a intentar hacer un programa de manipulación de imágenes al que llamaron GIMP. Eras las siglas de "General Image Manipulation Program" que más tarde pasarían a ser "GNU Image Manipulation Program", el Programa GNU para la Manipulación de Imágenes. Tuvieron la suerte de tener un profesor que les animó e incluso les dio buenas ideas para su desarrollo y el proyecto cogió fuerza en los ánimos de estos dos desarrolladores.

Durante casi un año se dedicaron ellos dos en solitario al desarrollo de la aplicación y en enero de 1996 publicaron una primera versión, la 0.54, que ya incluía el sistema de ampliación de GIMP por plug-ins, uno de sus grandes éxitos al permitir que desde el principio, la comunidad de desarrolladores pudiera ampliar de forma sencilla la funcionalidad. La aparición de plug-ins de todo tipo (filtros, efectos especiales, creación de imágenes, nuevas herramientas...) no se hizo esperar y, gracias a la lista de distribución que se creó para la comunidad interesada en GIMP, ésta tuvo una excelente vía de comunicación con lo que se aceleró aún más el crecimiento de GIMP.

El impulso definitivo dentro de la comunidad de GNU/Linux se lo dio a GIMP Larry Ewing con la creación de la famosa mascota Tux utilizando GIMP. Fue un trabajo que gustó mucho a la gente y era una clara muestra de lo que se podía obtener con GIMP.

Uno de los momentos críticos del proyecto fue la versión 0.60. En ella se hicieron fuertes cambios en el núcleo de GIMP, con lo que muchos plug-ins dejaron de funcionar. Hubo muchas reticencias a evolucionar a la nueva versión pero se logró. En esta versión ya se habían introducido grandes mejoras a la hora de la gestión de memoria con la "tiled memory" que permiten dividir la memoria en regiones y con ello, gestionarla mejor.

Es importante tener en cuenta el gran consumo de memoria de los programas de manipulación gráfica. Una buena gestión de memoria por parte del programa era vital para que GIMP pudiera seguir creciendo.

Ya en esta versión Peter comenzó a trabajar en el nuevo toolkit que debería de sustituir a Motif en GIMP, GDK/GTK. En principio no se quería que fuera un toolkit para uso general de construcción de interfaces gráficas, pero se creó justo en el momento que el software libre más lo necesitaba, y recibió tal apoyo de la comunidad que poco a poco se fue convirtiendo en el toolkit de referencia para los GUI de los sistemas libres y, cuando GNOME decidió adoptarlo, le dio el empujón definitivo para que junto a Qt, sean las dos grandes alternativas libres que en nada tienen que envidiar a los toolkits comerciales como Motif.

Entre los meses de junio y octubre de 1997 ocurrieron varios e importantes cambios dentro del proyecto GIMP. Peter y Spencer publican la última versión controlada por ellos, la 0.99.10, luego se gradúan y abandonan el liderazgo del proyecto. Entonces Federico Mena Quintero, destacado miembro de la

comunidad GNOME, se encarga de organizar las cosas para una versión estable, luego de lo cual se aparta un poco del proyecto. Pero ya GIMP disponía de su canal IRC y comenzó a crecer siguiendo el modelo de bazar.

En un principio se desarrolló para GNU/Linux pero actualmente existen versiones totalmente funcionales para Microsoft Windows y Mac OS X. Cabe señalar que la biblioteca de controles gráficos GTK, que fuera desarrollada para GIMP dio origen a al entorno gráfico Gnome. GIMP es quizás la primera gran aplicación libre para usuarios finales, también es considerado como una prueba de es posible crear aplicaciones para usuarios no avanzados que puedan ser aprovechadas productivamente, usando el proceso de desarrollo del software libre. [7]

Gimp como alternativa a Adobe Photoshop

¿Puede GIMP sustituir a Photoshop? Depende del escenario en el que vaya a ser usado. GIMP tiene la mayoría de las utilidades del producto estrella de la compañía Adobe y puede satisfacer las necesidades comunes para las que un usuario promedio utiliza un editor de imágenes. El retoque de fotografías, quitar el efecto de ojos rojos, escalar o convertir a otro formato una imagen son algunas tareas que se pueden realizar fácilmente con GIMP.

Al usuario típico Gimp le puede servir para resolver sus necesidades, esta es una herramienta gratuita, se puede descargar en un par de minutos de internet y ocupa muy poco en el disco. Gimp será suficiente para el retoque de imágenes o la creación de un banner para un sitio web.

Si optase por Photoshop las cosas ya se complicarían, primero porque tendría que escoger entre utilizar una versión shareware de 15 días (que ocupa unos cuantos cientos de megas), o bien optar entre comprar el software o “piratear” una versión de Photoshop, en este caso, tendría que buscar una fuente en el emule o bittorrent y bajarse los más de 700 Mb que ocupa Photoshop y además es probable que no utilizaría ni el 80% de las capacidades de este último.

Sería lógico preguntar si es necesario utilizar un software que consume muchos más recursos de la máquina, del que no se utilizarán todas las funcionalidades y que además es “pirata” teniendo disponible otro de gran calidad y con el que se pueden realizar las mismas tareas.

El punto débil de GIMP es la impresión. Al no brindar soporte para PANTONE o CMYK los trabajos de comerciales de imprenta pueden resultar más factibles de realizar con Photoshop que con GIMP, sin embargo es necesario señalar que actualmente se trabaja sobre este tema y si bien la inclusión de soporte para PANTONE no se vislumbra como posible por cuestiones de Licencias se trabaja en la inclusión de soporte nativo para CMYK. Por ahora existe un plugin llamado Separate que incluye las opciones de CMYK para GIMP.

El otro aspecto que afecta a GIMP es la interfaz. Los usuarios acostumbrados a Photoshop se sienten incómodos con las ventanas del GIMP, pero esto de cierta forma está pensado para que resulte así. Los desarrolladores de GIMP no intentan crear un “clon” de Photoshop, sino una herramienta de diseño nueva y potente, por esta razón no se debe esperar un cambio radical en la interfaz de GIMP en aras de hacerla semejante a la de Photoshop (aunque existe un proyecto denominado Gimpshop, que modifica los menús de Gimp para hacerlos más parecidos a los del programa de Adobe).

Esto causa que diseñadores con experiencia en el trabajo con Photoshop se sientan indispuestos a cambiar de herramienta, pues aprender a utilizar Gimp, les llevaría tiempo que no están dispuestos a emplear en familiarizarse con la herramienta. Una vez que aprendieron una vía para realizar su trabajo sencillamente se niegan a estudiar formas alternativas, por cuestiones de rendimiento o de comodidad, les resulta más difícil el cambio de herramientas.

Por esto se recomienda el uso de Gimp desde niveles básicos de enseñanza, al tiempo que sería acertado incluirlo como parte de los planes de estudio de carreras afines al diseño gráfico. Destacar que al tener una versión totalmente funcional para Windows no es necesario esperar a la migración de plataforma para comenzar la capacitación en esta herramienta. Esto tiene la ventaja de que llegada la hora de la migración los usuarios ya estarán familiarizados con la aplicación y no se producirán

Desde el punto de vista del costo, actualizar a Adobe Photoshop CS3 (una de las últimas versiones de la herramienta) cuesta \$199.00 US y comprar esa misma versión completa \$649.00 US [8]. Los requerimientos de hardware también son mayores [9]:

- Intel Pentium 4, Intel Centrino, Intel Xeon o Intel Core Duo™ (o compatible).
- Microsoft Windows XP con Service Pack 2 o Windows Vista™ Home Premium, Business, Ultimate o Enterprise (certificado para ediciones de 32 bits).
- 512 Mb de RAM
- 64 Mb de RAM de video.
- 1GB de espacio disponible en el disco duro (se requiere espacio libre adicional durante la instalación).
- Unidad de DVD-ROM.
- Es necesaria una conexión a Internet o telefónica para activar el producto.

Por otro lado GIMP puede ejecutarse como mínimo sobre un Pentium II, sobre Windows 98/ME/2000/XP y sobre la mayoría de los sistemas Unix que corren X11. Requiere 128 Mb de RAM mínimo y 256 Mb o más como recomendado. El espacio en disco esta determinado por el tipo de imágenes a procesar. No necesita conexión a Internet, aunque se puede descargar con facilidad desde la web puede ser instalado localmente si se copian los paquetes necesarios desde un CD u otro dispositivo externo. Como puede apreciarse actualmente la infraestructura de nuestro país se presta más a la utilización de GIMP que Photoshop, siendo este un factor importante a considerar.

Por otro lado es necesario recordar que el uso de software libre es también una filosofía, que sería importante implantar en nuestro país, pues se basa en el trabajo comunitario y en compartir la información y el conocimiento.

CAPÍTULO 2: LA PROGRAMACIÓN

2.1 La programación aplicada a Herramientas Libres

Las aplicaciones “libres”, desarrolladas según el modelo de “bazar” pueden ser fácilmente adaptadas o mejoradas por la comunidad de usuarios. Los desarrolladores liberan sus aplicaciones bajo los términos de licencias como *la GNU Public License* (GPL) o *Creative Commons* (CC) con el objetivo de garantizar el libre uso del software. Publican el código, brindan documentación y soporte del mismo para cualquiera que este interesado en incorporarse al proyecto. Debe tenerse en cuenta que esto no atenta contra la calidad del software; todo lo contrario, las aplicaciones desarrolladas siguiendo este modelo han demostrado ser más seguras, y es más fácil encontrar y corregir errores en ellas que en los productos propietarios.

El tema de la programación se puede abordar desde dos perspectivas: la primera es el desarrollo y mantenimiento de las aplicaciones, para lo cual generalmente se crean proyectos más formales y se requieren conocimientos de programación más profundos. Algunos de los lenguajes de programación utilizados son C/C++, Perl y Python. El otro enfoque se basa en la creación de scripts o plugins para añadir nuevas funcionalidades a los programas. Estos pueden ser desarrollados por usuarios con menos conocimientos de programación y muchos son creados por usuarios individuales que no tiene relación directa con los desarrolladores oficiales del proyecto. Se utiliza C así como Python, Perl y otros lenguajes script.

En la investigación se abordará sobre el segundo de estos aspectos, la creación y la utilidad de los scripts y plugins para Gimp y Blender.

Aunque las aplicaciones tienen múltiples funcionalidades en ocasiones los usuarios intentan ampliar las características del programa original, pero sin tocar nada de él, solo añadiéndole nuevas utilidades desde ficheros externos. Se debe tener en cuenta que no todos los programas admiten estas extensiones, ya que requiere tiempo preparar el modo de interactuar entre el programa original y las extensiones. Sin embargo tanto Blender como GIMP admiten, e incorporan de forma natural el uso de scripts y plugins.

2.2 Scripts y Plugins

Cuando las extensiones son sencillas, como puede ser simplemente una secuencia de acciones ya aportadas por el programa original que se desea se repitan automáticamente, se denominan scripts.

Sin embargo a veces se desea añadir una funcionalidad que el programa original no admite, por ejemplo, un tratamiento novedoso a una imagen. Será un proceso más largo que el anterior y hay que tener más conocimientos de programación y de la propia aplicación. Entonces se le llama plugin.

Definición de Script

En informática la palabra script (sustantivo anglosajón, literalmente “guión”) se refiere a un conjunto de instrucciones, que permite automatizar tareas sencillas y repetitivas creando pequeñas utilidades. Son interpretadas en tiempo de ejecución, y generalmente son archivos de texto.

Definición de Plugin

Un plugin o plug-in (en inglés “enchufar”) es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal. Se utilizan como una forma de expandir programas de forma modular, de manera que se puedan añadir nuevas funcionalidades sin afectar a las ya existentes ni complicar el desarrollo del programa principal.

2.3 Python

Python es un lenguaje de programación fácil de aprender y potente. Es un lenguaje interpretado, multiplataforma, orientado a objetos de propósito general. La elegante sintaxis de Python, su gestión de tipos dinámica y su naturaleza interpretada hacen de él el lenguaje ideal para guiones (scripts) y desarrollo rápido de aplicaciones en muchas áreas y en la mayoría de las plataformas. Dispone de eficaces estructuras de datos de alto nivel y una solución de programación orientada a objetos simple pero eficaz.

Python permite mantener de forma sencilla interacción con el sistema operativo, y resulta muy adecuado para manipular archivos de texto. Esta característica hace que muchas distribuciones de GNU/Linux utilicen Python para sus herramientas de configuración, aunque puede ser usado en casi cualquier sistema operativo. También es ampliamente utilizado en la Web.

Ventajas

- *Desarrollo más rápido:* Puedes escribir un programa, salvarlo y ejecutarlo. En un lenguaje compilado tienes que pasar por los pasos de compilar y ligar el software, lo cual puede ser un proceso lento.
- *Multiplataforma:* El mismo código funciona en cualquier arquitectura, la única condición es que disponga del intérprete del lenguaje. No es necesario compilar el código una vez para cada arquitectura.

Desventajas

- *Lentitud:* Los programas interpretados son más lentos que los compilados. Sin embargo los programas interpretados suelen ser cortos (como los scripts), en los que la diferencia es inapreciable.

Instalar Python

Para muchas distribuciones de GNU/Linux python viene instalado por defecto, ejemplo de estas son Gentoo, Debian, Ubuntu. Para Windows los usuarios deberán instalarlo como cualquier otra aplicación.

Estructura de un script

Un script en Python básicamente tiene 3 partes importantes. La primera sería donde se importan los módulos necesarios que luego serán utilizados en el código, la segunda donde se declaran las variables globales y las funciones, y la tercera el programa en sí.

Los módulos son código que ofrece Python para poder hacer un trabajo concreto. Se puede importar estos módulos y utilizarlos para algunas de las tareas del script. Cuando se importa un módulo no es necesario saber como hace las cosas, sino simplemente como se debe utilizar en el trabajo. Un módulo en Python se importa de esta manera:

```
>>> import nombre_modulo
```

Por otra parte, las funciones en Python (y en cualquier otro lenguaje) sirven para no tener que repetir una y otra vez un trozo de código. Las funciones tienen que ser creadas por el usuario y servirán para hacer algún trabajo concreto. Las funciones en Python se declaran de la siguiente forma:

```
>>> def nombre_funcion(lista_parametros): # Se puede declarar una función sin parámetros. >>>
    sentencias
```

La palabra reservada "**def**" indica que se está creando una función, luego viene el nombre de la misma, lo siguiente es colocar entre paréntesis los parámetros que se van a utilizar separados por comas y para terminar ":". A continuación se escribe el código que tiene que estar indentado (esto es muy importante), porque de esta manera Python sabrá que esas instrucciones pertenecen a esa función, en el momento que una línea no tenga esa separación, Python no lo reconocerá como parte de la función.

El código que ejecuta Python al arrancar un script debe de encontrarse, normalmente al final del script después de las funciones, y luego de poner:

```
>>> if __name__ == "__main__": []
```

2.4 Extendiendo Blender

Scripts

Blender tiene una característica muy poderosa: un intérprete de Python totalmente funcional (Ver Figura 3 Anexos). Esto le permite a cualquier usuario añadir funcionalidades a Blender escribiendo un simple script. Los script de Python tenían unas funcionalidades limitadas en Blender 2.25, la última de las versiones que liberó NaN, sin embargo cuando Blender se hizo código abierto muchos de los desarrolladores que se movían alrededor de la Fundación Blender lo eligieron para trabajar y, junto con el cambio de UI, el API de Python es probablemente la parte de Blender que ha tenido un desarrollo más grande.

Blender tiene una ventana texto junto con los otros tipos de ventana la cual es accesible vía el botón del menú Tipos de Ventana o a través de SHIFT-F11. Una ventana de texto recién abierta es gris y está vacía, y tiene una barra de herramientas muy sencilla (*Barra de herramientas de texto*).

Barra de herramientas de texto.

El botón Menú permite seleccionar que buffer de texto tiene que ser mostrado, y también le permite crear un nuevo buffer o cargar un fichero de texto. Si elige cargar un fichero en la ventana texto temporalmente se crea una ventana de selección de ficheros, con las funciones usuales. Una vez que el buffer de texto está en la ventana texto, esta se convierte en un editor de texto muy simple. Escribir en el teclado provoca que aparezca texto en el buffer de texto. Como siempre con presionar clic izquierdo arrastrar y liberar se selecciona el texto. Los siguientes comandos del teclado son aplicables:

CTRL-C - Copia el texto marcado al portapapeles.

CTRL-X - Corta el texto marcado al portapapeles.

CTRL-V - Pega el texto marcado desde portapapeles en la posición del cursor en la Ventana Texto.

ALT-S - Guarda el texto como un fichero de texto, una Ventana de Selección de Ficheros aparecerá.

ALT-O - Carga un texto, una Ventana de Selección de Ficheros aparecerá.

ALT-F - Emergerá la herramienta buscar.

SHIFT-ALT-F o RMB - Emergerá el Menú Fichero para la Ventana de Texto.

ALT-J - Emergerá un panel donde puede especificar un número de línea para situar el cursor.

ALT-P - Ejecuta el texto como un script de Python.

ALT-U – Deshacer.

CTRL-R - Reabrir (recargar) el buffer actual.

ALT-M - Convierte el contenido de la ventana de texto en texto 3D (max. 100 caracteres).

ALT-R – Rehacer



Figura 1. Barra de Herramientas de Texto.

El portapapeles de Blender que corta/copia/pega está *separado* del portapapeles de Windows así que *no podrá* copiar desde o hacia Blender. Para acceder al portapapeles de Windows utilice SHIFT-CTRL-C SHIFT-CTRL-V. Para GNU/Linux esta opción no está disponible. Para eliminar un buffer de texto tan sólo presione el botón 'X' próximo al nombre del buffer. La más importante combinación de teclas es ALT-P la cual hace que el contenido del buffer sea pasado al intérprete de Python de Blender.

Otros usos para la ventana de texto

La ventana de texto se utiliza también a la hora de compartir los ficheros .blend con la comunidad. Puede ser usada para escribir un texto aclaratorio de los contenidos del fichero. Esto es mucho más interesante que tener dicha aclaración en un archivo separado. Para ellos hay que mantener visible la ventana mientras se salva el fichero. Por ejemplo, si se desea compartir el fichero bajo alguna licencia se puede especificar en la ventana de texto.

Plugins

También se puede escribir un plugin. El lenguaje que se utiliza para esto es C, aunque se debe señalar que resulta más factible escribir un script que un plugin estos últimos no dejan de resultar útiles en determinadas tareas. Blender soporta dos tipos de plugins que permiten un control adicional sobre el proceso de render. Son los plugins de textura y de secuencia. Los plugins de textura pueden extender el editor de texturas para dar control sobre la intensidad y el color mientras que los de secuencia brindan facilidades de post-producción y transición en el editor de secuencia.

El sistema de Plugins

El soporte de plugins en Blender está dado por la familia de llamadas `dlopen()`. El sistema `dlopen` permite a un programa, Blender en este caso, utilizar objetos compilados como si fueran parte del programa. La ventaja del uso del sistema `dlopen` para los plugins es que es muy rápido para acceder a una función, y no hay sobrecarga en la interfaz del plugin, lo que resulta crítico cuando (como los plugins de textura) el plugin puede ser llamado varios millones de veces en un único render. La desventaja es que como el código del plugin funciona como parte del propio Blender si el plugin falla Blender también.

Interacción entre texturas y materiales

Blender difiere de la mayoría de paquetes 3D en lo que se refiere a la separación lógica entre materiales y textura. En Blender las texturas son objetos que devuelven ciertos valores. Los materiales controlan el mapeado de texturas en los objetos, lo que está afectado, en que cantidad, en que manera, etc.

Los plugins diseñados adecuadamente deberían incluir únicamente variables que afecten a la señal devuelta y no al mapeado de ella. Es mejor incluir botones para controlar el escalado, el rango, los ejes únicamente cuando hagan a la textura más fácil de usar o aceleran el calculo. De otra manera los botones del material hacen a esos botones redundantes, y la interfaz incrementa innecesariamente su complejidad.

Interacción con el usuario

Blender no tiene ninguna manera de saber cuántas entradas espera el plugin, por lo que le es posible a un usuario adjuntar sólo una entrada a un plugin que espera dos. Por esta razón es importante comprobar siempre los buffers que usa el plugin para asegurarnos de que son todos válidos. Los plugins de secuencia deberían incluir también una etiqueta de texto describiendo el número de entradas requeridas en la interfaz de botones. [10]

2.5 Extendiendo Gimp

Una de las principales ventajas del GIMP es la facilidad con que se puede extender su funcionalidad por medio de plugins y scripts.

Scripts

Los scripts en Python para Gimp son una herramienta muy potente y, lo más importante es que los usuarios pueden crear los propios teniendo pocos conocimientos de Python y de la API de Gimp.

Alternativas a Python-fu

Scheme: Gimp lo trae por defecto para crear scripts. Es otra manera de crear scripts para Gimp, aunque puede resultar complicada de leer. Funciona mediante paréntesis y se puede crear variables, funciones, listas. La primera desventaja frente a Python es que puede resultar complicado de leer con respecto a este y el segundo es que Python además proporciona una gran variedad de módulos para realizar otros trabajos. Esta investigación no aborda los script con Scheme.

Perl-fu: Perl brinda todas las posibilidades que ofrece Python, solo es otro lenguaje de programación. Aunque es un lenguaje de características similares a Python a simple vista y sin demasiada experiencia, Perl resulta más complejo. Perl se caracteriza por su capacidad de hacer lo mismo de varias formas diferentes. Esto suele confundir a usuarios noveles, e incluso usuarios experimentados tienen problemas para entender código de otras personas.

Python-fu

Para poder trabajar con los scripts de python en GIMP primeramente se debe tener ambos instalados (python y GIMP). En algunas ocasiones, dependiendo de la distribución de GNU/Linux del puesto de trabajo será necesario instalar un paquete adicional llamado *gimp-python*. Después de esto solo es necesario copiar o crear los nuevos scripts y colocarlos en el directorio para los plugins de Gimp que en GNU/Linux suele ser algo como: `/home/<user>/.gimp-2.X/plugin-ins`.

Aunque esta investigación se enfoca y recomienda el uso de Python-fu se debe señalar que esta no es la opción definitiva para el tema de los plugins. Las desventajas más importantes que se pueden señalar son, en primer lugar, la pobre interfaz de usuario, lo que provoca un difícil entendimiento de los scripts cuando son un poco complejos. La otra es que GIMP no ofrece buenas herramientas para realizar un buen debug.

Ventajas

- Uso de la librería estándar de Python
- Cambios sobre la marcha
- GUIs automáticas
- Reutilización de plugins

Desventajas

- Difícil debugging
- GUIs Simples [11]

Estructura de un script en Gimp

La estructura de un script en Gimp, es muy parecida a la de un script en Python. Solo que ahora no hay un programa principal como tal, sino una llamada a una función de nombre "register", a la que se le pasan una serie de parámetros con la información del script. El código para nuestro programa se encuentra en una función. La función "register" lo que contiene es el nombre del script, una descripción del programa, los componentes que aparecerán en la ventana, la función que va a ejecutar el script. Así que los scripts en Gimp tendrán básicamente la misma estructura que uno en Python.

Función Principal

Es aquí donde el script cumple con su objetivo, es decir, donde realiza la acción o acciones para las que fue creado.

```
def funcion_principal (img, drawable, parametro1, parametro2, ... , parametroN):
```

Los dos primeros parámetros son obligatorios. El primero hace referencia a la imagen y el segundo a la capa activa. Luego se colocan en orden todos los demás que necesite el script. A estos se suma la variable `pdb`, que es una variable global y que no es necesario pasar en la función principal.

Función "register"

Es en esta parte donde Gimp recibe la información para ejecutar el script. La función `register` espera la menos once parámetros en el siguiente orden:

- Nombre del script. Tiene que ser único, y no debe tener acentos, espacios o caracteres especiales.
- Los próximos cuatro parámetros son nombre de autor, texto de ayuda y textos descriptivos.
- Fecha del script.

- Ruta donde se va a encontrar el script en el menú de Gimp. Esta puede empezar de dos formas: <Image>, que indica que el script será accesible desde el menú de la imagen abierta y la segunda forma es <Toolbox>, que permite ejecutar el scrip desde el menú de la caja de herramientas de Gimp.
- Tipo de Imagen.
- Parámetros del script.
- Resultados. Se dejan siempre como un par de corchetes [].
- Función: El nombre de la función de nuestro script.

Parámetros de los script. Se colocan entre paréntesis y son cuatro elementos separados por comas, aunque en algunos casos existe un quinto elemento especial. Estos son:

- Tipo de elemento.
- Nombre de la variable (sin espacios).
- Título que aparecerá a la izquierda del elemento en el script
- Valor por defecto del elemento.
- Parámetro especial (caso que sea necesario).

La API de GIMP

Para encontrar nuevas funciones o conocer las características de alguna función específica Gimp tiene una herramienta denominada navegador de bases de datos procesales (PDB). Es posible acceder a este desde el menú principal Exts/Python-Fu/PDB Browser (Ver Figura 4 Anexos). La función del PDB es listar las funciones disponibles para los programadores.

Este es la mayor fuente de información sobre las funciones disponibles en la versión en uso de GIMP. El navegador permite realizar búsquedas restringidas por el nombre de la función o por su descripción. Para conocer más sobre una determinada función al ser seleccionada en la parte derecha del navegador se muestra una descripción de la misma, los parámetros que necesita y si retorna alguna variable.

Errores

GIMP tiene dos maneras de presentar los errores, la primera y más efectiva, pues mostrará cualquier mensaje de error es la consola de comandos si el programa ha sido inicializado desde esta. La otra es a través de la Consola de errores, a la cual se puede acceder desde Archivo/Diálogos/Consola de errores, (Ver Figura 5 Anexos) aunque esta variante no muestra todos los errores que pueden ocurrir.

Recomendaciones sobre los scripts para Gimp

- Tener en cuenta que los acentos, en los comentarios pueden traer problemas de portabilidad. No poner acentos en los parámetros del script.
- Comprobar que todos los parámetros que se pasan en la función register se pasan correctamente en la función principal.
- Colocar siempre la cabecera de python: `#!/usr/bin/env python` .
- Comprobar que el nombre del script (primer parámetro de la función register) es único y que no tiene espacios ni caracteres especiales.
- Recordar que en Linux se deben dar permisos de ejecución al script.
- De ser posible comprobar que el script funciona tanto para Linux como para Windows.
- Recordar que una vez cargado el script solo se podrán realizar cambios al mismo dentro de la función principal. Si se añade un nuevo parámetro a la función register habrá que reiniciar el Gimp.

Plugins

Con el uso de plugins se pueden modificar las imágenes casi de la misma manera en que lo harían los usuarios. Si se necesita una nueva capacidad en el GIMP, puede ser mucho más fácil agregarla escribiendo un pequeño plugin, que modificando el código complejo que hace a la propia aplicación. Estos añadidos pueden tener unas cien o doscientas líneas de código fuente en C.

La distribución principal del GIMP incluye varias decenas de plugin, que se instalan automáticamente junto con la aplicación, se pueden encontrar estos en el menú Exts/Explorador de complementos (Ver Figura 6 Anexos). La mayoría pueden ser accedidos por el menú Filtros. Cada entrada a este menú se corresponde con un plugin aunque no es la única forma de utilizarlos. Los plugins también pueden ser utilizados por el usuario sin que este conozca que la funcionalidad o herramienta que esta ejecutando es realmente un plugin. [12]

Los plugins pueden ser creados por cualquier usuario, y también se pueden descargar libremente desde la Web. Ciertamente necesitan para su desarrollo un dominio mayor de la aplicación que los script, al tiempo que requieren más conocimientos de programación, principalmente del lenguaje C. Es necesario además tener cuenta que los plugins escritos para una versión de Gimp pueden no funcionar (de hecho es muy probable que no) para otras versiones. Siempre es importante cerciorarse a la hora de utilizar un plugin que está disponible para la versión de Gimp utilizada.

2.6 Ventajas del uso de scripts y plugins

Estos añadidos ofrecen la posibilidad de extender las funcionalidades de las herramientas de forma natural y robusta, proporcionando una vía de realizar tareas repetitivas o sencillas sin necesidad de modificar el código de los programas. Esto no significa que sean solo una opción más en el uso de las aplicaciones. Constituyen una parte importante en el trabajo con la herramienta y en general la calidad y potencialidades de estos añadidos en si mismos se presentan como una de las principales ventajas y fortalezas de las aplicaciones estudiadas. Una ventaja adicional es la posibilidad de encontrar gratuitamente en Internet tanto plugins y scripts como ayuda referente a la instalación, uso y desarrollo de los mismos.

2.7 Equipos de Trabajo

Desarrollar nuevos plugins y scripts para Gimp y Blender usando python y C como lenguajes de programación resulta más sencillo de lo que podría esperarse. Aunque los añadidos pueden ser fácilmente creados por usuarios no programadores aquellos que tengan conocimientos más sólidos de lenguajes y técnicas de programación tendrán un mejor aprovechamiento y explotarán mejor las funcionalidades que aportan a las aplicaciones abordadas. Por esta razón se recomienda que a la hora de formar equipos de trabajo que vayan a utilizar estas herramientas se incluya personal con conocimientos de programación, preferentemente en lenguajes scripts y en C. Es importante tener en cuenta que estos añadidos tienen un desarrollo muy dinámico, se actualizan y se crean nuevos a un ritmo realmente rápido, además, puede suceder que alguna versión de los mismos no sea compatible con la versión en uso de la aplicación. De esta manera la creación de equipos de desarrolladores que dieran soporte a las versiones de plugins utilizados en los trabajos de diseño y que además crearan nuevos en respuesta a las necesidades de los usuarios constituye una necesidad para el desarrollo de una industria de diseño basada en el uso de herramientas libres, al tiempo que contribuirá al desarrollo de las herramientas y del propio movimiento del Software Libre.

CONCLUSIONES

Durante la investigación se llegó a las siguientes conclusiones:

1. Blender y Gimp pueden sustituir a sus contrapartes de software propietario.

Estos dos programas tienen la calidad necesaria para realizar todos los trabajos de diseño en sus respectivas áreas. En el caso de Gimp todavía tiene que mejorar para lograr sustituir completamente al Photoshop de Adobe, sobre todo para trabajos de impresión, aunque se ya se trabaja en este tema. Sin embargo en el contexto nacional se presenta como una alternativa a considerar, dado su carácter gratuito y sus bajos requerimientos de hardware.

2. Ventajas del uso de la programación.

El uso de plugins y script puede facilitar el trabajo a los diseñadores y usuarios en general. Su fácil desarrollo posibilita que puedan ser creados y modificados por las propias personas que utilizan el programa sin que sean necesarios conocimientos profundos de programación. También están disponibles en la Web desde donde pueden ser descargados, y encontrar documentación y soporte sobre los mismos.

En general se considera que los objetivos propuestos para la investigación se cumplieron en su totalidad. Se han expuesto las características generales de cada herramienta en particular, así como su historia y un acercamiento a su futuro desarrollo. También se exponen las ventajas que brinda el uso de la programación aplicada a estas herramientas con el objetivo de extender sus funcionalidades, y se demuestra la factibilidad de la aplicar las mismas. Con la realización de este trabajo se han adquirido conocimientos sobre las herramientas de diseño de código abierto y sobre la creación y aplicación de plugins y scripts a estas aplicaciones. En sentido general se logró un acercamiento a las todavía poco conocidas herramientas de diseño del software libre. Se espera la investigación contribuya a la migración, al constituir una guía para los usuarios acerca de las alternativas que brinda el software libre en el área del diseño gráfico.

RECOMENDACIONES

Se recomienda continuar con la investigación para incluir otras herramientas, se propone abordar al menos Cinelerra, Inkscape y Sodipodi, pues constituyen parte del grupo de aplicaciones del Software Libre que están ganando terreno dentro del diseño y la edición. Por otro lado se recomienda crear un equipo de trabajo que agrupe a diseñadores y programadores para seguir el desarrollo de estos programas. Es necesario recordar que estas herramientas tienen ciclos de desarrollo rápidos, y que la comunidad de usuarios aporta muchas nuevas funcionalidades y manuales sobre el trabajo con estas aplicaciones. Una vez identificadas las herramientas a utilizar en la Universidad el equipo de trabajo podría dedicarse además al desarrollo de estas, proporcionando de esta manera facilidades para el trabajo interno del centro y contribuyendo al desarrollo del Software Libre a nivel internacional.

REFERENCIAS BIBLIOGRÁFICAS

Introducción

- [1] ZUÑIGA, L. Título: Voces libres de los campos digitales: una investigación social sobre el Software Libre en América Latina y el Caribe. San José, 2006. 126 p.

Capítulo 1

- [2] DEL VALLE, AMAURY E. Estados Unidos bloquea Internet en Cuba (I).
URL: <http://www.cubavsbloqueo.cu/Default.aspx?tabid=1497>
Fecha Consulta 20/12/06
- [3] MORENO, LUCIANO Título: *Diseño Gráfico*
URL: <http://www.desarrolloweb.com/articulos/1277.php>
Fecha Consulta: 20/12/06
- [4] a.d. Título: *History*
URL: <http://www.blender.org/blenderorg/blender-foundation/history/>
Fecha Consulta: 12/01/07
- [5] a.d. Título: *System Requirements*
URL: <http://www.blender.org/features-gallery/requirements/>
Fecha Consulta: 12/01/07
- [6] a.d. Título: *System Requirements*
URL: <http://www.autodesk.co.uk/adsk/servlet/index?siteID=452932&id=7745635>
Fecha Consulta: 15/02/07

- [7] DEL CASTILLO, ÁLVARO Título: *Historia de GIMP*
URL: <http://acsblog.es/articulos/trunk/TodoLinux/Gimp/gimp1/html/x29.html>
Fecha Consulta: 07/03/07
- [8] a.d. Título: *Adobe Photoshop CS3*
URL: https://store1.adobe.com/cfusion/store/index.cfm?store=OLS-US&view=ols_prod&category=/Applications/Photoshop&distributionMethod=FULL&nr=0
Fecha Consulta: 28/03/07
- [9] a.d. Título: *Requisitos del sistema e idiomas*
URL: <http://www.adobe.com/es/products/photoshop/photoshop/systemreqs/>
Fecha Consulta: 28/03/07

Capítulo 2

- [10] a.d. Título: *Plugins*
URL: <http://www-users.cs.umn.edu/%7Emein/blender/plugins/plugindoc.html>
Fecha Consulta: 24/04/07
- [11] CONTI, JUAN JOSÉ Título: *Cinturón blanco en Python-Fu*
URL: <http://docs.gimp.org/es/gimp-scripting.html>
Fecha Consulta: 24/04/07
- [12] a.d. Título: *Programar en el GIMP*
URL: <http://docs.gimp.org/es/gimp-scripting.html>
Fecha Consulta: 24/04/07

BIBLIOGRAFÍA

- a.d. Título: *Blender*
URL: <http://www.blender.org/>
Fecha Consulta: 10/01/07
- a.d. Título: *Features*
URL: <http://www.blender.org/features-gallery/features/>
Fecha Consulta: 10/01/07
- a.d. Título: *3D Software Comparison Chart*
URL: http://wiki.cgsociety.org/index.php/Comparison_of_3d_tools
Fecha Consulta: 16/02/07
- a.d. Título: *Gimp*
URL: <http://gimp.org/>
Fecha Consulta: 16/02/07
- a.d. Título: *About Gimp*
URL: <http://gimp.org/about/introduction.html>
Fecha Consulta: 07/03/07
- a.d. Título: *History*
URL: <http://www.blender.org/blenderorg/blender-foundation/history/>
Fecha Consulta: 12/01/07
- a.d. Título: *System Requirements*
URL: <http://www.blender.org/features-gallery/requirements/>
Fecha Consulta: 12/01/07

- a.d. Título: *Definiciones de Plugin en la web*
URL: http://www.google.com.cu/search?hl=es&client=firefox-a&rls=org.mozilla:en-GB:official&defl=es&q=define:Plugin&sa=X&oi=glossary_definition&ct=title
Fecha Consulta: 11/04/07
- a.d. Título: *Definiciones de Script en la web*
URL: http://www.google.com.cu/search?hl=es&client=firefox-a&rls=org.mozilla:en-GB:official&defl=es&q=define:Script&sa=X&oi=glossary_definition&ct=title
Fecha Consulta: 11/04/07
- a.d. Título: *System Requirements*
URL: <http://www.autodesk.co.uk/adsk/servlet/index?siteID=452932&id=7745635>
Fecha Consulta: 15/02/07
- a.d. Título: *Adobe Photoshop CS3*
URL: https://store1.adobe.com/cfusion/store/index.cfm?store=OLS-US&view=ols_prod&category=/Applications/Photoshop&distributionMethod=FULL&nr=0
Fecha Consulta: 28/03/07
- a.d. Título: *Requisitos del sistema e idiomas*
URL: <http://www.adobe.com/es/products/photoshop/photoshop/systemreqs/>
Fecha Consulta: 28/03/07
- a.d. Título: *Plugins*
URL: <http://www-users.cs.umn.edu/%7Emein/blender/plugins/plugindoc.html>
Fecha Consulta: 24/04/07

- a.d. Título: *Programar en el GIMP*
URL: <http://docs.gimp.org/es/gimp-scripting.html>
Fecha Consulta: 24/04/07
- AMIDALA. Título: *¿Qué es el Diseño?*
URL: <http://www.chanta.cl/foros/showthread.php?t=13184>
Fecha Consulta 20/12/06
- CONTI, JUAN JOSÉ Título: *Cinturón blanco en Python-Fu*
URL: <http://docs.gimp.org/es/gimp-scripting.html>
Fecha Consulta: 24/04/07
- DEL CASTILLO, ÁLVARO Título: *Historia de GIMP*
URL: <http://acsblog.es/articulos/trunk/TodoLinux/Gimp/gimp1/html/x29.html>
Fecha Consulta: 07/03/07
- DEL VALLE, AMAURY E. Estados Unidos bloquea Internet en Cuba (I).
URL: <http://www.cubavsbloqueo.cu/Default.aspx?tabid=1497>
Fecha Consulta 20/12/06
- FROMMEL, O Título: *Gimp script*
URL: <http://www.linux-magazine.es/issue/05/GimpScripts.pdf>
Fecha Consulta: 10/05/07
- LÓPEZ, CARLOS Título: *Pythonlópez*
URL: <http://klopes.iespana.es/tutoriales/pythonlopes/pythonlopes1.htm>
Fecha Consulta: 27/04/07

- MCCLURE, PHILIP Título: *Gimp vs. Photoshop*
URL: http://grimthing.com/archives/2007/01/11/Gimp_vs_Photoshop/
Fecha Consulta: 07/03/07
- MORENO, LUCIANO Título: *Introducción al Diseño Gráfico*
URL: <http://www.desarrolloweb.com/articulos/1276.php>
Fecha Consulta: 20/12/06
- MORENO, LUCIANO Título: *Diseño Gráfico*
URL: <http://www.desarrolloweb.com/articulos/1277.php>
Fecha Consulta: 20/12/06
- PÉREZ, JAVIER PACHECO Título: *Python-fu para no programadores*
URL: <http://www.javielinux.com/articulos/pygimp/book1.htm>
Fecha Consulta: 24/04/07
- ZUÑIGA, L. Título: *Voces libres de los campos digitales: una investigación social sobre el Software Libre en América Latina y el Caribe*. San José, 2006. 126 p.

ANEXOS

Anexo 1. Imágenes.

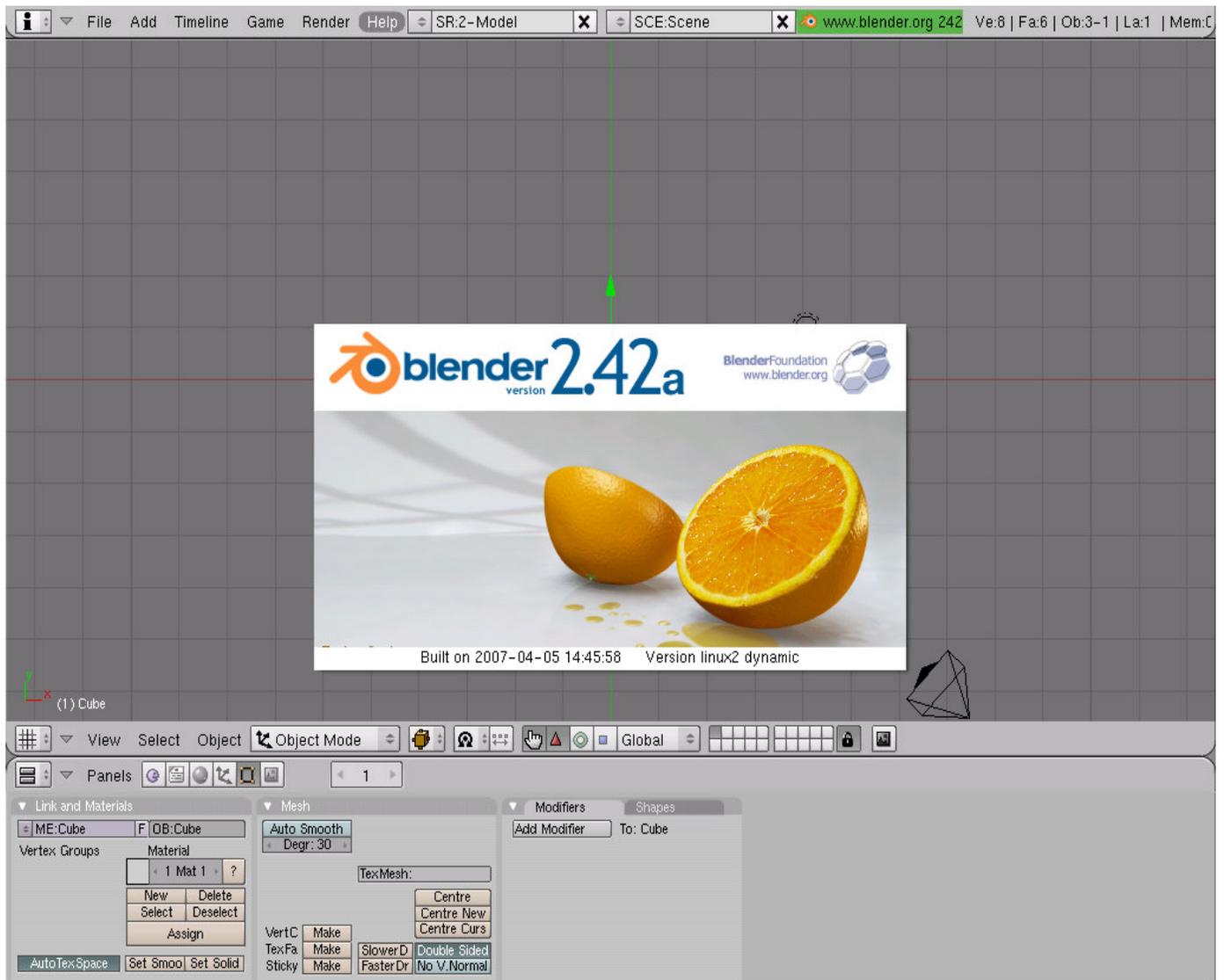


Figura 1. Pantalla de bienvenida del Blender.

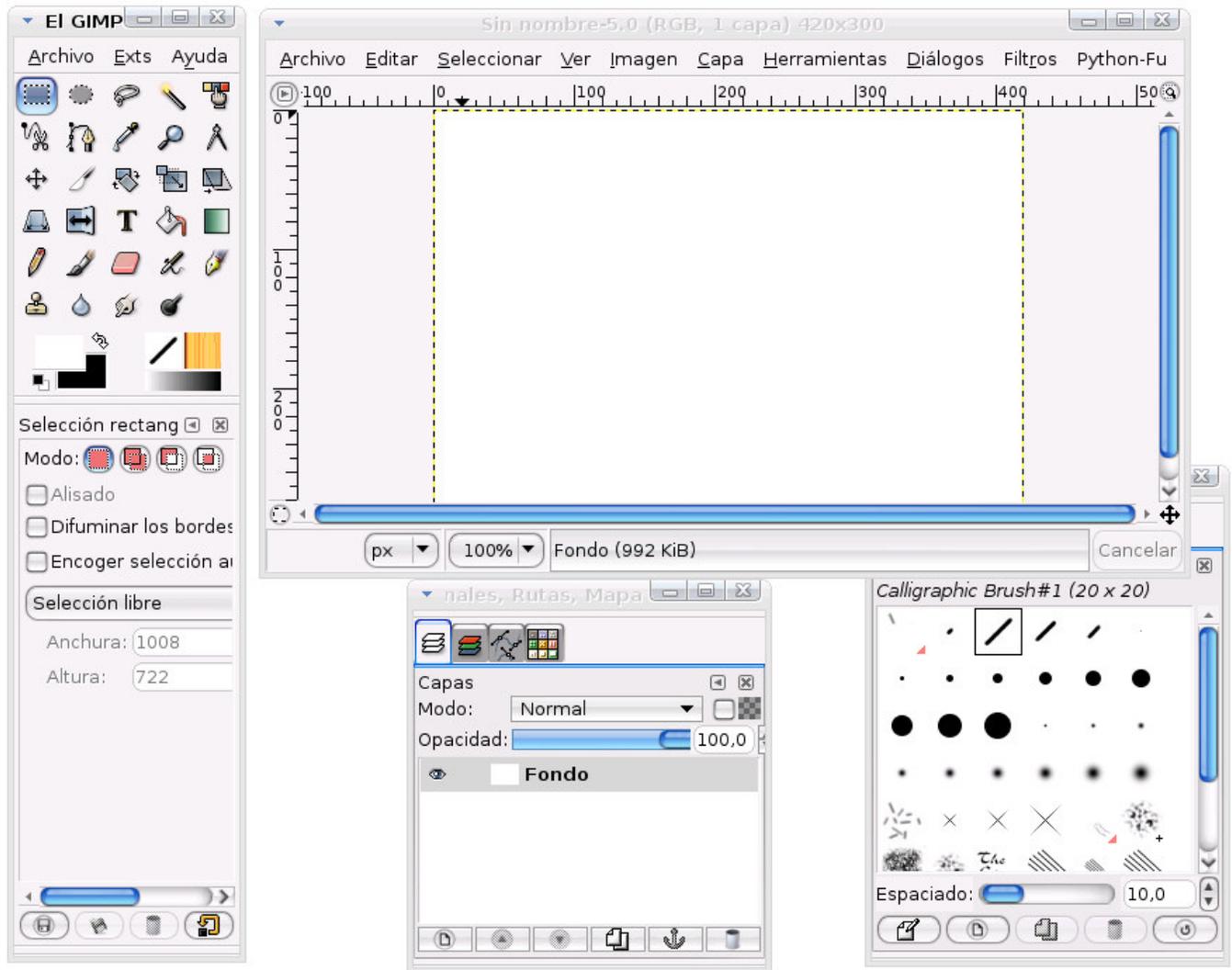


Figura 2. Ventanas del Gimp.

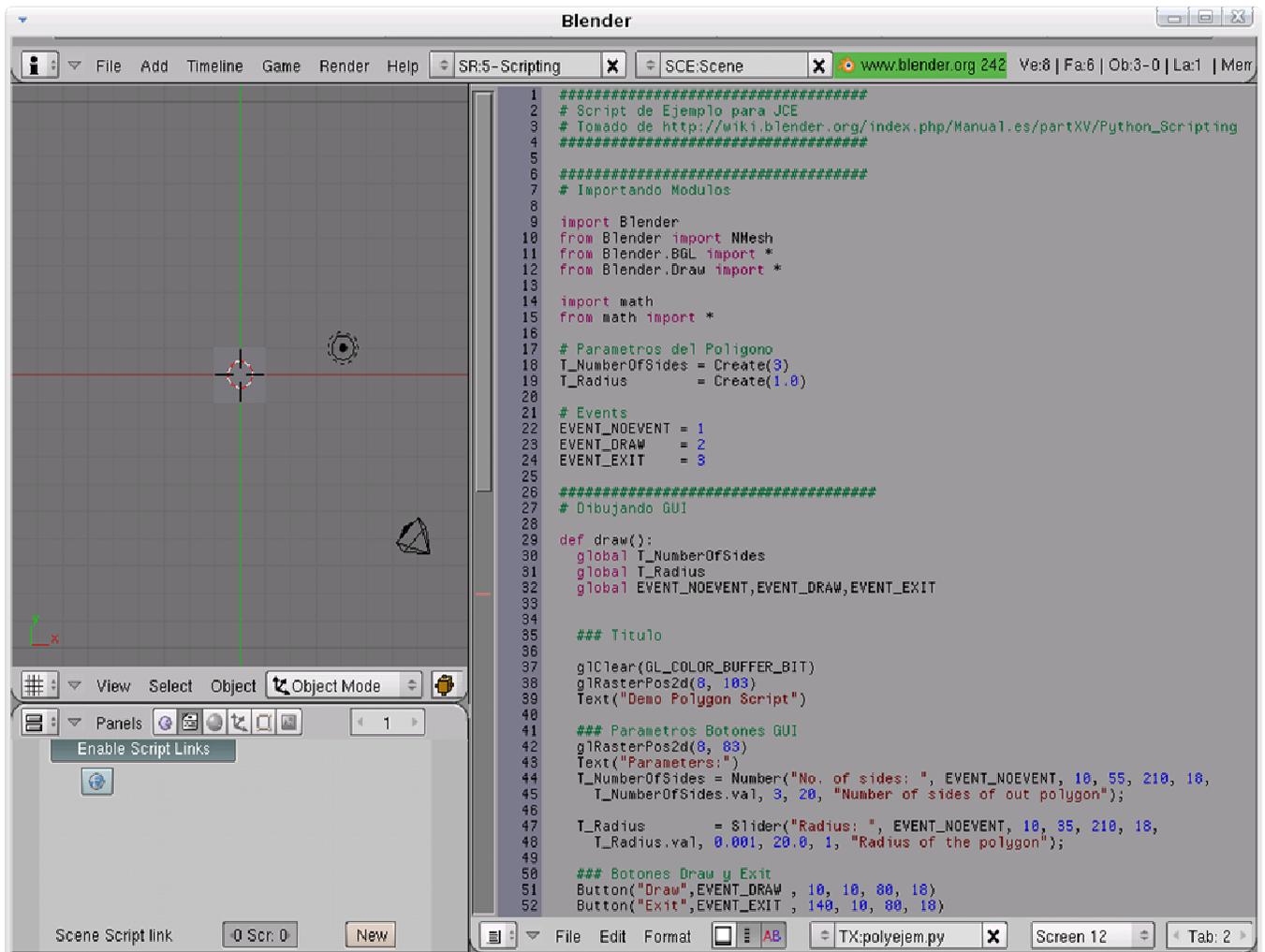


Figura 3. Ventana de Texto del Blender.

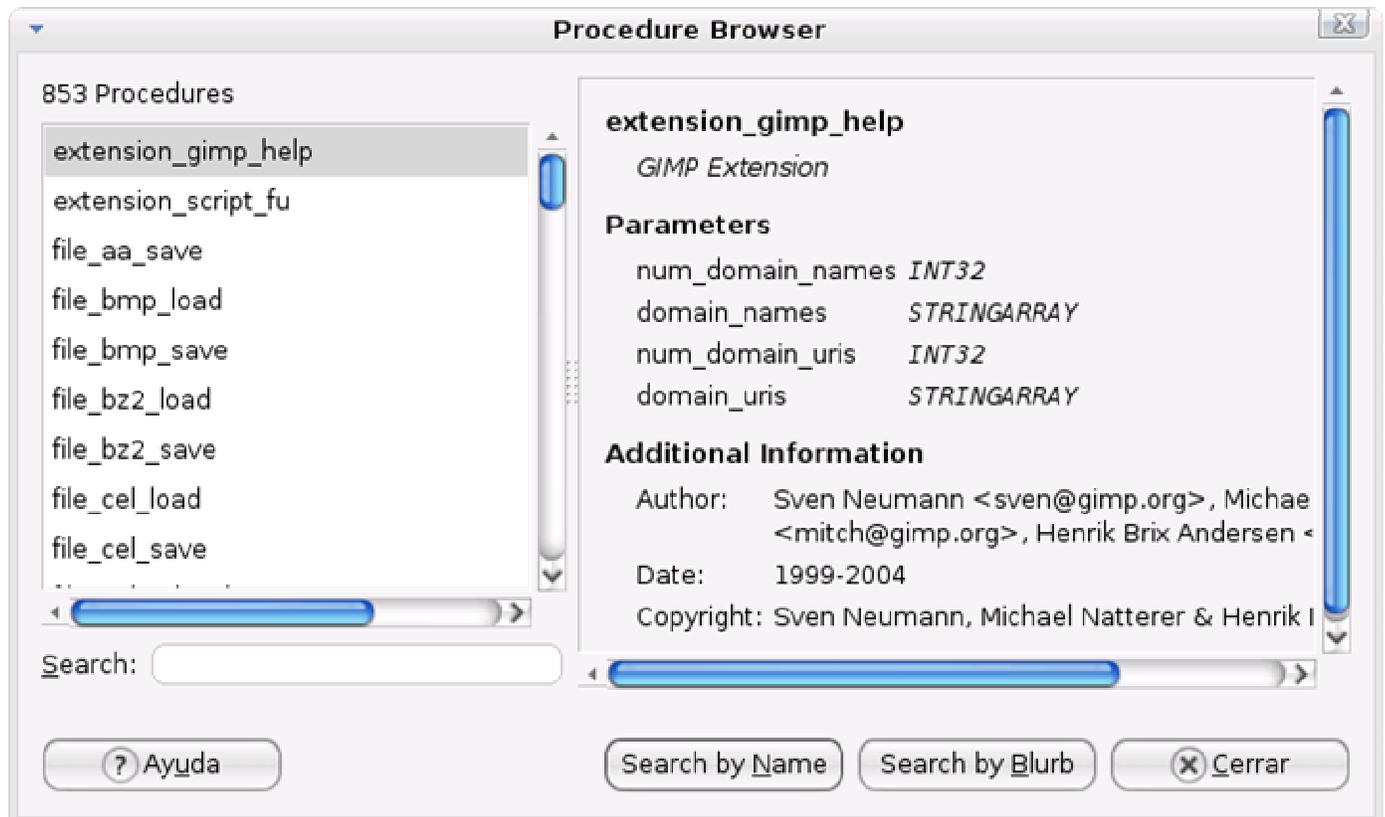


Figura 4. PDB Browser.

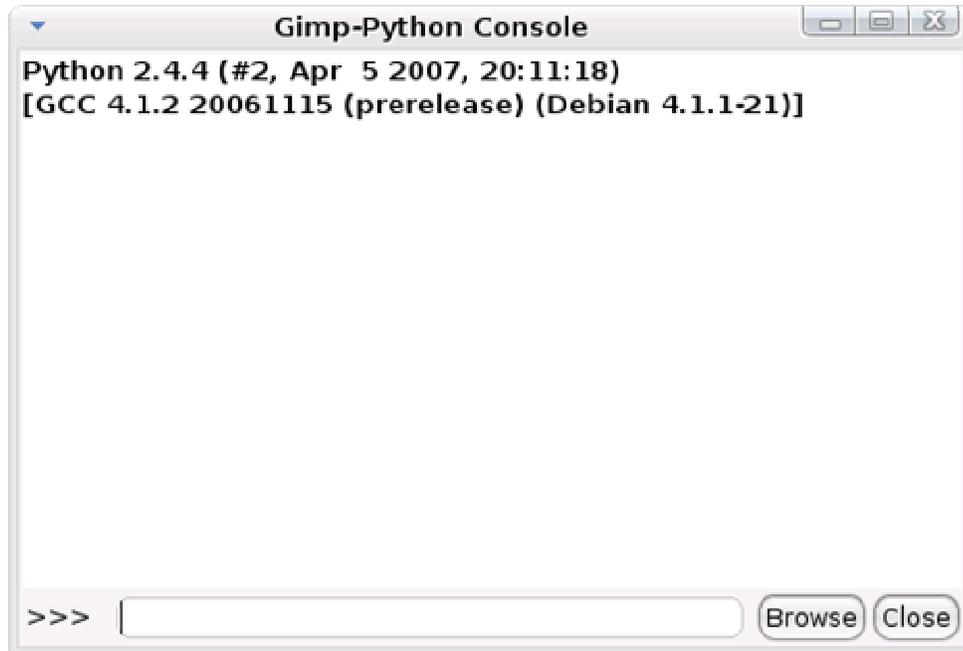


Figura 5. Consola de Errores del Gimp.

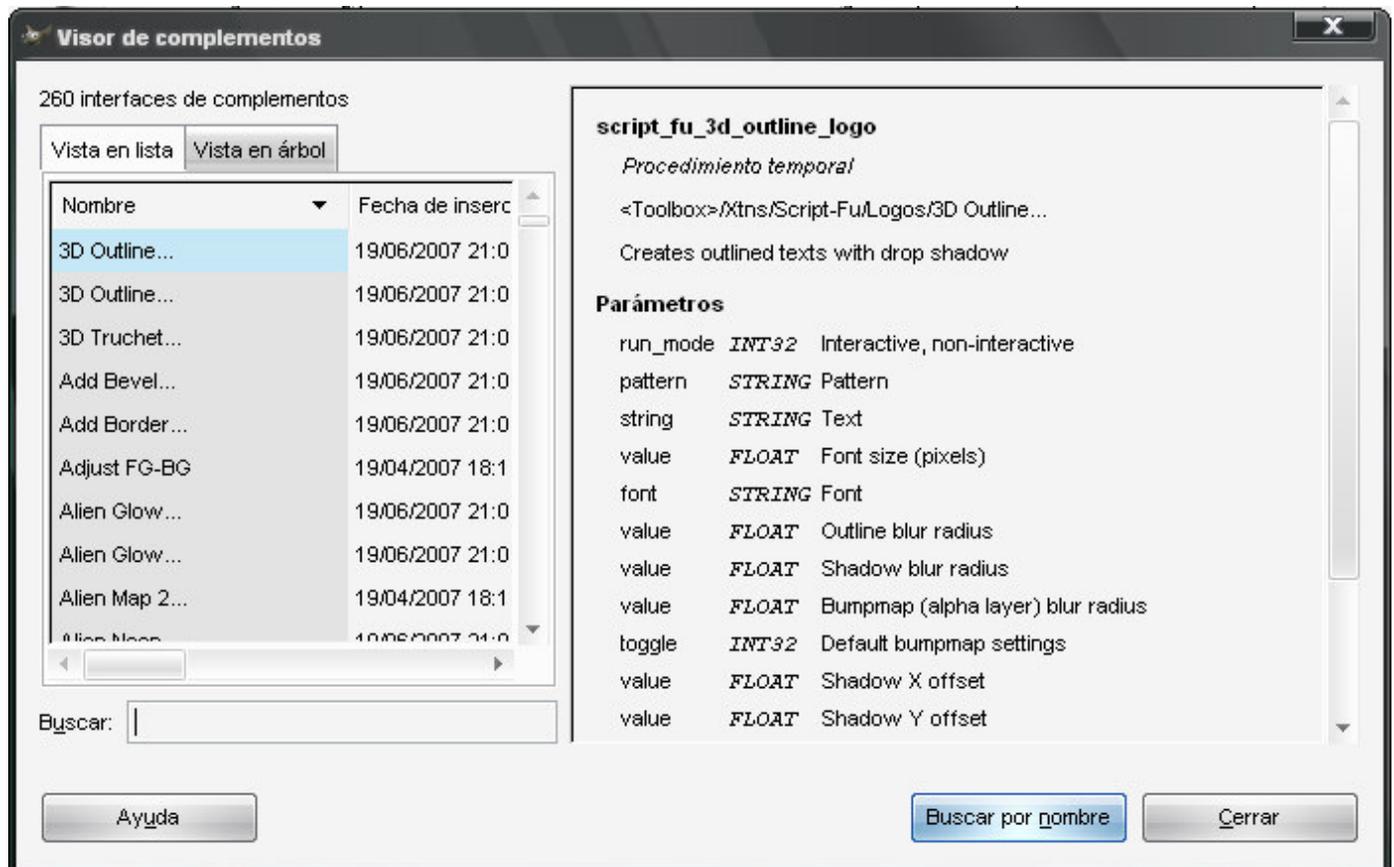


Figura 6. Explorador de Componentes.